Titre: Title:	Engineering Fairness in Machine Learning Systems
Auteur: Author:	Moses Openja
Date:	2026
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Openja, M. (2026). Engineering Fairness in Machine Learning Systems [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/76114/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/76114/
Directeurs de recherche: Advisors:	Foutse Khomh
Programme: Program:	Génie logiciel

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Engineering Fairness in Machine Learning Systems

MOSES OPENJA
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Mars 2026

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Engineering Fairness in Machine Learning Systems

présentée par **Moses OPENJA**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Michel DESMARAIS, président
Foutse KHOMH, membre et directeur de recherche
Heng LI, membre
Federica SARRO, membre externe

DEDICATION

Dedication to

My dearest beloved wife, Tusiime Patricia, for her unwavering support, prayers, and companionship throughout this important journey. To My beloved mother, Rose Biywaga (decease) whose love and sacrifices continue to guide and inspire me. I further dedicate this work to My father, John Yuka; My big brother, Godfred and My dear siblings; Adubango Emmanuel and his family; as well as the family of Tumusiime John.

I can't thank you all enough for the endless love,

prayers, sacrifices, and support...

ACKNOWLEDGEMENTS

My sincere gratitude goes to my supervisor, prof. Foutse Khomh, for his tremendous support, guidance, motivation, and patience throughout my PhD journey, and for the excellent research environment he provided that enabled me to carry out this work. His vast knowledge and logical way of thinking has been very helpful to me throughout my doctoral studies. Furthermore, his knowledge and warmth made the collaboration highly beneficial for me. It has been a great privilege to work under his supervision; the lessons he taught me extends far beyond what is written in this thesis and will continue to guide me in both my academic career and personal life.

I would also like to sincerely thank the members of my PhD thesis jury for their interest in my work and for dedicating their time and effort to carefully review my thesis and provide insightful and constructive feedback. In particular, I am grateful to the jury president, Prof. Michel Desmarais, and the jury members Prof. Federica Sarro, Prof. Heng Li, and Prof. Soumaya Yacout for their valuable comments and contributions, which helped strengthen this dissertation.

RÉSUMÉ

Les systèmes d'apprentissage automatique (AA), y compris les modèles classiques, les réseaux de neurones profonds (DNN) et les grands modèles de langage (LLMs), sont de plus en plus déployés dans des domaines à enjeux élevés tels que la santé, le recrutement, la finance et les politiques publiques, où des décisions injustes ou biaisées peuvent avoir des conséquences sociétales majeures. Bien que des progrès substantiels aient été réalisés en matière d'apprentissage équitable, la plupart des approches existantes considèrent l'équité comme une contrainte statique lors de l'entraînement et supposent des distributions de données fixes, des attributs sensibles prédéfinis et une possibilité de réentraîner sans restriction. En pratique, toutefois, les systèmes d'apprentissage automatique évoluent en permanence : le biais peut être introduit lors du prétraitement des données et de la sélection des caractéristiques, émerger au cours de l'entraînement à travers l'apprentissage des représentations, persister dans des modèles préentraînés, ou être amplifié après le déploiement en raison de changements de distribution. Il est donc essentiel de garantir que l'équité n'est pas compromise tout au long du développement et de l'évolution des systèmes d'AA—non seulement pour produire des systèmes logiciels de qualité, mais aussi pour la conformité réglementaire et l'audit. Cette thèse envisage l'équité comme une propriété de qualité au niveau logiciel des systèmes d'AA, qui doit être continuellement testée, diagnostiquée et corrigée tout au long du cycle de vie du système. S'inspirant de l'ingénierie logicielle et de l'assurance qualité, la thèse adopte une démarche proactive en matière d'équité, mettant l'accent sur la détection précoce et la correction à faible coût des composants générant des biais, tout en développant également des mécanismes de correction post-entraînement et post-déploiement spécifiques aux contextes où le réentraînement n'est pas faisable. Aux stades des données et de l'entraînement, la thèse propose une méthode systématique pour identifier les caractéristiques du modèle susceptibles de produire des biais, même sans connaître à l'avance les attributs sensibles. Grâce à une analyse cause-à-effet, cette approche permet aux experts du domaine de raisonner sur les sources de biais et démontre empiriquement que corriger ou supprimer les caractéristiques moins importantes mais génératrices de biais peut améliorer significativement l'équité sans détériorer la performance prédictive.

Pour les DNN entraînés, la thèse introduit `FairFLRep`, un cadre de localisation et de réparation des défauts sensible à l'équité qui considère les comportements injustes comme des défectuosités localisées au sein des représentations apprises. `FairFLRep` identifie les poids de neurones contribuant de manière disproportionnée aux disparités entre sous-groupes et applique des mises à jour ciblées et minimales des paramètres, inspirées par les techniques

de correction logicielle issues de l'ingénierie logicielle traditionnelle. En outre, la thèse étudie l'équité dans les modèles transformateurs préentraînés au travers de deux études complémentaires. La première examine la manière dont les biais sociaux sont représentés en interne dans les transformateurs, à partir d'un ensemble de données de relations biaisées — des triplets codant des stéréotypes selon divers types de biais — et de techniques d'attribution neuronale. L'étude montre que la connaissance biaisée est souvent concentrée dans de petits sous-ensembles de neurones tout en étant aussi redondamment encodée à travers le réseau. Ces résultats établissent le biais comme un phénomène à la fois architectural et représentatif, plutôt qu'un simple artéfact lié aux données, et en évaluent l'impact sur des tâches d'ingénierie logicielle en aval. Sur la base de ces observations, la seconde étude propose un cadre de réparation rigoureux pour les transformateurs préentraînés. Elle introduit une localisation neuronale contrastive et optimale selon Pareto afin d'identifier les neurones de biais de manière concentrée et fiable, puis applique une optimisation axée sur la justesse à l'aide de la méthode d'optimisation par essaim particulaire (PSO) pour les corriger. Les résultats démontrent que la réparation ciblée au niveau des neurones est efficace pour atténuer les biais et constitue une alternative légère et pratique au réentraînement global. Enfin, la thèse aborde la réparation de l'équité sous changement de distribution, un phénomène fréquent dans les déploiements réels, même lorsque les facteurs sources du décalage de données ou de biais sont inconnus ou impossibles à mesurer. Pour dépasser les limites des approches de réparation statiques comme `FairFLRep`, la thèse propose `DShift-FRep`, un cadre adaptatif intégrant un alignement des données, un finetuning léger à point de départ chaud, et une réparation ciblée de neurones via PSO. En adoptant le paradigme Adapter puis Réparer, `DShift-FRep` traite conjointement le décalage de domaine et le biais résiduel, améliorant à la fois la performance prédictive et l'équité entre la source et la cible, sur des données d'images et tabulaires. Les résultats expérimentaux montrent que `DShift-FRep` surpasse l'état de l'art en adaptation de domaine, auto-apprentissage et techniques de réparation statique, tout en évitant une dérive inutile des paramètres.

Dans l'ensemble, cette thèse apporte des bases rigoureuses et pratiques pour développer, tester, maintenir et déployer des systèmes d'AA équitables, fiables et responsables dans des contextes réels.

ABSTRACT

Machine learning (ML) systems—including classical models, deep neural networks (DNNs), and large language models (LLMs)—are increasingly deployed in high-stakes domains such as healthcare, hiring and finance, where unfair or biased decisions can have significant societal consequences. Although substantial progress has been made in fairness-aware learning, most existing approaches treat fairness as a static, training-time constraint and assume fixed data distributions, predefined sensitive attributes, and unrestricted retraining. In practice, however, ML systems evolve continuously: bias may be introduced during data preprocessing and feature selection, implicitly encoded during representation learning as models internalize spurious correlations and demographic proxies, persist in pretrained models as inherited biases embedded in pretrained parameters, or be amplified after deployment due to distribution shifts. Ensuring that fairness is not violated throughout the development and evolution of ML systems is therefore critical—not only for producing high-quality software systems, but also for regulatory compliance and auditing.

This thesis frames fairness as a software-level quality property of ML systems–one that must be continuously tested, diagnosed, and repaired throughout the system lifecycle. Inspired by software engineering and quality assurance, the thesis adopts a shift-left perspective on fairness, emphasizing early detection and low-cost repair of bias-inducing components, while also developing post-training and post-deployment repair mechanisms for settings in which retraining is infeasible. At the data and training stages, the thesis proposes a systematic method for identifying potential bias-inducing features of the model without knowing the sensitive attributes upfront. Using cause-to-effect analysis, the approach enables domain experts to reason about bias sources and shows empirically that correcting or removing less important yet bias-inducing features can significantly improve fairness without degrading predictive performance.

For trained DNNs, the thesis introduces `FairFLRep`, a fairness-aware fault localization and repair framework that treats unfair behavior as a localized defect in learned representations. `FairFLRep` identifies neuron weights that disproportionately contribute to subgroup disparities and applies targeted, minimal parameter updates inspired by program repair techniques from traditional software engineering. Moreover, the thesis investigates fairness in pretrained transformer models through two complementary studies. First, it examines how social bias is internally represented in transformers, using datasets of biased relations—triplets encoding stereotypes across multiple bias types—and neuron attribution techniques. The study shows

that biased knowledge is often concentrated in small subsets of neurons while also being redundantly encoded across the network. These findings establish bias as an architectural and representational phenomenon rather than a purely data-driven artifact and evaluate its impact on downstream software engineering tasks. Building on these insights, the second study proposes a principled repair framework for pretrained transformers. It introduces Pareto-optimal, contrastive neuron localization to identify compact and high-confidence bias-inducing neurons and applies correctness-aware optimization using Particle Swarm Optimization (PSO) to repair them. Results show that targeted neuron-level repair is effective for bias mitigation and provides a lightweight, practical alternative to global retraining. Finally, the thesis addresses fairness repair under distribution shifts, which commonly arise in real-world deployment—even when the underlying nuisance factors driving data and bias shifts are unknown or unmeasurable. To overcome the limitations of static repair approaches such as `FairFLRep`, the thesis proposes `DShift-FRep`, an adaptive framework that integrates data-level alignment, lightweight warm-start fine-tuning, and targeted neuron repair via PSO. By following an Adapt-then-Repair paradigm, `DShift-FRep` jointly addresses domain shift and residual bias, improving both predictive performance and fairness across source and target distributions on image and tabular data. Experimental results show that `DShift-FRep` outperforms state-of-the-art domain adaptation, self-training, and static repair techniques, while avoiding unnecessary parameter drift.

Overall, this thesis provides principled and practical foundations for developing, testing, maintaining, and deploying fair, reliable, and responsible ML systems in real-world settings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

ML/DL	Machine Learning/ Deep Learning
DNN	Deep Neuron Networks
PSO	Particle Swarm Optimization
`FairFLRep`	Fairness aware fault localization and repair of Deep Neural Networks
`DShift-FRep`	Adaptive Fairness-Aware Repair of DNN under Distribution Shifts
LLMs	Large Language Models
MLMs	Masked Language Models
FFNs	Feed Forward Networks
IG	Integrated Gradients
CBA	Contrastive Bias Score
NLP	Natural Language Processing
SPD	Statistical Parity Difference
DI	Disparate Impact
EOD	Equalized Odds
EOD-max	Maximum Equalized Odds
TPR	True Positive Rate
FPR	False Positive Rate
Kn	Knowledge Neuron
PPL	Perplexity
CNN	Convolution Neural Network
AI	Artificial intelligence
MMD	Maximum Mean Discrepanc
OT	Optimal Transport
MLP	Multilayer Perceptron
RQ	Research Question
SE	Software Engineering
DANN	Domain-adversarial neural networks

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Machine learning (ML) systems, including classical models, deep neural networks (DNNs), and transformer-based large language models (LLMs), are increasingly deployed in high-stakes domains such as healthcare [1–3], finance [4], hiring [5], and public policy [6, 7]. While these systems often achieve impressive predictive performance, their deployment has revealed persistent concerns around reliability, robustness, and fairness [8, 9]. In particular, models can exhibit systematic performance disparities across demographic groups, amplify historical biases embedded in data, or degrade unpredictably when exposed to new environments. These challenges undermine trust in ML systems and pose tangible societal risks [10–14].

A key limitation of existing fairness-aware learning approaches (e.g., [15–19]) is their narrow scope. Most techniques assume static data distributions [15–19], fully observed sensitive attributes [20–26], and a single training phase in which fairness is enforced through modified objectives or constraints. However, In practice, ML systems evolve continuously. Bias can be introduced early through feature selection and data preprocessing, emerge during representation learning, or resurface after deployment due to distribution shifts. Addressing fairness therefore requires moving beyond isolated mitigation techniques toward a lifecycle-oriented perspective [27–30].

Even when adopting a lifecycle view, fairness interventions must respect the original objectives of ML systems. Effective mitigation should preserve predictive performance, avoid removing task-relevant information, minimize unintended regressions, and remain cost-effective. At the data level, this requires systematically identifying all potential bias-inducing features and understanding whether these features are also necessary for task performance. Such identification enables domain experts to assess whether the bias introduced by specific features is acceptable, given the contextual and ethical implications of the decision task. Supporting such context-specific judgments requires transparency into what induces bias and how. For example, in cases of statistical discrimination [31,32] in economics, auto insurers may consider gender differences in accident rates to maximize profit by charging higher premiums to male drivers. Similarly, in legal contexts, dominant doctrines of discrimination—such as equal protection under the Fourteenth Amendment of the U.S. Constitution—prohibit discriminatory intent by government actors, while still allowing certain race-conscious policies (e.g., affirmative action [33]) to further compelling governmental interests. These examples illustrate that fairness is inherently context-dependent, reinforcing the need to expose bias-inducing factors rather than obscuring them through blunt mitigation.

Beyond the data level, fairness can also be addressed at the model and post-processing levels by localizing internal components responsible for biased behavior and repairing them directly. Such targeted interventions preserve a model's original functionality and avoid discontinuous changes that ignore interaction effects among parameters. Crucially, fairness mitigation should optimize explicit fairness objectives, rather than achieving apparent gains through information loss or degraded confidence. Fairness should therefore be formulated as a continuous, correctness-aware optimization problem over localized parameters, enabling minimal and controlled updates that reduce bias while avoiding catastrophic interference.

Motivated by this view, this thesis frames fairness as a software-level property of ML systems. From a software engineering perspective, bias is treated as a defect that must be tested, diagnosed, and repaired throughout the system lifecycle [27, 28]. Drawing inspiration from software quality assurance, the thesis adopts a shift-left perspective on fairness: the earlier bias is identified and corrected, the lower the cost and the greater the long-term benefit [29, 30]. Just as software defects are cheaper to fix when detected early, bias-inducing components in ML systems should be addressed as close as possible to their source [34]. Following this principle, the thesis first investigates fairness repair during the training process. We study bias detection and evaluation in classical ML models, showing that unfair behavior can often be traced to a small subset of bias-inducing features. By identifying and removing low-importance yet harmful features, we demonstrate that fairness can be improved without degrading predictive performance. More importantly, this process supports informed decision-making by domain experts through systematic identification and prioritization of bias sources. This work illustrates how repairing the training pipeline can prevent bias from being encoded into learned representations.

Despite early interventions, many real-world scenarios involve pretrained or already-deployed models that cannot be retrained from scratch due to cost, data availability, or operational constraints. While data- and training-time mitigation strategies are desirable, they are not always sufficient to eliminate unfair behavior. In many cases, biased decisions are not merely the result of surface-level correlations in the data, but are encoded within a model's internal representations. From this perspective, bias should not be viewed solely as an undesirable output pattern, but as an internal fault: a localized representational mechanism that systematically skews model behavior under specific input–output conditions. Such faults may remain latent during training and only manifest under particular demographic configurations or deployment environments. This viewpoint motivates a fairness-aware repair paradigm analogous to post-deployment debugging in traditional software systems [28, 35]. If harmful or biased associations are stored in identifiable internal representations and are selectively activated by certain inputs, then fairness violations can be addressed through targeted lo-

calization and intervention, rather than global retraining or indiscriminate suppression. In particular, attribution-based analysis [36–38] enables the identification of a small subset of fairness-critical neurons whose activations disproportionately contribute to group-level disparities. By directly modifying the outgoing weights of these neurons, it becomes possible to generate targeted repair "patches" that mitigate unfair behavior while preserving task-relevant knowledge [15, 39, 40]. Crucially, such repairs must be optimized rather than heuristically applied: parameter updates should explicitly balance fairness improvement against correctness preservation, ensuring that bias reduction does not arise from information loss or degraded predictive confidence. To address this setting, the thesis introduces `FairFLRep`, a neuron-level fault localization and repair framework for DNNs. `FairFLRep` treats bias as a localized defect within the model, identifies neurons that disproportionately contribute to unfair outcomes, and repairs them through targeted parameter updates. This minimally invasive approach improves fairness while preserving accuracy, demonstrating that trained models can be effectively repaired without full retraining.

The thesis further extends fairness analysis to pretrained transformer-based models, which present unique challenges due to their scale, contextualized representations, and emergent behaviors. In such models, bias is not only triggered by lexical cues [41, 42] but is also embedded in internal representations that are difficult to interpret or control [43]. Specifically, we conduct two complementary study: (i) First, we perform a systematic analysis of the localization of social bias in pretrained transformer models. Using a dedicated dataset capturing multiple types of social bias and neuron attribution techniques, we show that biased knowledge is often concentrated in relatively small subsets of neurons, while also being redundantly encoded across the network. This study is primarily diagnostic: it establishes that bias in pretrained transformers is not purely a data-level phenomenon, but an architectural and representational property of the model itself. At the same time, the observed redundancy highlights inherent limitations of naive localization and motivates the need for more principled intervention strategies. (ii) Second, we move beyond analysis to intervention by proposing a repair-oriented framework for bias mitigation in pretrained transformers. We introduce Pareto-optimal, contrastive localization, which identifies compact and high-confidence sets of bias-inducing neurons by explicitly balancing fairness impact against correctness preservation. Building on this localization, we propose a targeted neuron-level repair strategy based on correctness-aware optimization using Particle Swarm Optimization (PSO). Experimental results show that global retraining is not always necessary for effective bias mitigation: carefully repairing a small, high-quality subset of parameters can significantly reduce biased behavior while preserving task performance. Together, these two studies establish a progression from bias localization and understanding to principled, optimization-driven repair in

pretrained transformer models, reinforcing the thesis's central claim that fairness should be treated as a diagnosable and repairable software property of ML systems.

Finally, the thesis addresses fairness repair under distribution shifts, a setting that closely reflects real-world deployment. Distribution shifts—covariate, label, or concept—can simultaneously degrade accuracy and exacerbate subgroup disparities. To address this challenge, we propose `DShift-FRep`, an adaptive and modular framework for fairness-aware testing, localization, and repair under hybrid shifts. `DShift-FRep` integrates fairness-aware label propagation, shift-vector-based data alignment, warm-start fine-tuning, and targeted neuron repair using PSO. Across both classical DNNs and transformer-based LLMs, `DShift-FRep` improves fairness and accuracy while avoiding unnecessary parameter drift.

Overall, this thesis presents a unified, lifecycle-oriented framework for fairness in ML systems. By framing bias as a diagnosable and repairable defect—and by developing methods that operate across training, post-training, pretrained models, and shifting environments—this work provides principled and practical foundations for building fair, reliable, and robust ML systems.

1.1 Thesis Overview

This thesis investigates how fairness can be systematically engineered, tested, localized, and repaired in ML systems across their lifecycle. Rather than treating fairness as a one-time training objective, fairness is viewed as a software quality property that must be continuously assessed and maintained as models evolve, are reused, or encounter new data distributions.

Adopting a shift-left perspective, the thesis progressively studies fairness interventions at increasing levels of abstraction and system maturity. It begins with bias detection in training data and classical ML models, advances to post-training repair of DNNs, extends to fairness localization in pretrained LLMs, and culminates in fairness repair under non-stationary and shifting data distributions.

Across these settings, the thesis develops localization-driven and optimization-based repair techniques that preserve predictive performance while mitigating unfair behavior. Together, these methods form a coherent framework for engineering fairness in modern ML systems operating under real-world constraints.

1.2 Thesis Contribution

This thesis makes the following key contributions toward a unified framework for fairness-aware testing and repair of ML systems. The contributions are organized according to the stage at which fairness is diagnosed and repaired, progressing from early, training-time interventions to post-deployment repair under distribution shift.

- *Bias detection and repair at the training stage (shift-left fairness)* — This first contribution focuses on early-stage fairness diagnosis, before complex models are trained. We propose a systematic method for identifying all bias-inducing features in classical ML models without assuming prior knowledge of sensitive attributes.

  Using a cause-to-effect analysis based on single- and double-feature swapping, the method isolates features whose presence disproportionately influences model predictions for specific subgroups. This enables domain experts to reason about bias sources even when sensitive attributes are proxy-based, partially observed, or unavailable.

  Empirical results show that removing or correcting features that induce bias but contribute little to predictive performance can significantly improve fairness without degrading accuracy. This contribution demonstrates the feasibility and value of shift-left fairness repair, where bias is mitigated at the lowest cost point in the ML pipeline.

- *`FairFLRep`: Fairness-aware fault localization and repair in trained DNNs* — The second contribution addresses fairness failures in trained deep neural networks, where retraining from scratch may be impractical. We introduce `FairFLRep`, a neuron weights fairness-aware fault localization and repair framework. `FairFLRep` treats unfair behavior as a localized defect within learned representations. It identifies neuron weights that disproportionately contribute to group-level disparities by analyzing the weights gradient loss and forward impacts and applies targeted, minimally invasive weight updates to repair them.

  Experimental results show that `FairFLRep` effectively reduces fairness violations while preserving—or in some cases improving—task performance. This work demonstrates that fairness violations can be localized, diagnosed, and repaired in DNNs at the representation level, bridging concepts from software fault localization and ML fairness.

- *Bias localization in pretrained transformer models (Diagnostic study)* — This contribution presents a systematic empirical study of how social bias is represented in pretrained transformer models. We construct a dataset capturing multiple types of social bias and adapt neuron attribution techniques to trace stereotype-related behavior to

internal components of pretrained transformers. We show that biased knowledge is often concentrated in small subsets of neurons within the transformer feed forward networks (FFN), while also being redundantly encoded across the model. This study provides interpretable evidence that bias in transformers is an architectural and representational phenomenon, not solely a data artifact. The observed redundancy reveals important limitations of naive localization and motivates the need for interaction-aware and optimization-driven mitigation strategies.

Suppression experiments demonstrate that modifying these neurons by zeroing can significantly reduce bias while maintaining downstream task performance. However, we also observe that bias is partially redundantly encoded, revealing important limitations of naive neuron suppression and motivating more interaction-aware repair strategies.

- *Fairness-aware repair in pretrained transformer models via Pareto-optimal localization (Intervention study)* — Building on the diagnostic findings, this contribution proposes a principled repair framework for mitigating bias in pretrained transformers. We introduce Pareto-optimal, contrastive neuron localization, which identifies compact and high-confidence bias-inducing neurons by jointly considering fairness impact and correctness preservation. Using correctness-aware optimization based on PSO, we repair these neurons through minimal parameter updates.

  Results show that global retraining is not always necessary for effective bias mitigation: targeted neuron-level repair provides a lightweight and practical alternative. We further demonstrate that suppression-based interventions alone are insufficient for production systems, as hard suppression introduces information loss and uncertainty without explicitly optimizing fairness. In contrast, repair-based interventions improve interpretability, auditability, and stability—properties critical in high-stakes domains such as healthcare, hiring, and education.

- *Fairness repair of DNN under distribution shift (`DShift-FRep`)* — This final contribution addresses fairness repair in DNNs, under non-stationary data distributions, where fairness guarantees established during training or post-training repair may no longer hold. We propose `DShift-FRep`, an adaptive and modular framework for fairness-aware repair under hybrid distribution shifts. `DShift-FRep` integrates fairness-aware label propagation, shift-vector-based data alignment, warm-start fine-tuning, and targeted neuron repair via PSO.

  `DShift-FRep` achieves superior cross-domain stability, improving both fairness and accuracy compared to static repair and domain adaptation baselines. This contribution

demonstrates how fairness repair can be made robust to evolving domains, unifying adaptation and repair within a single framework.

Collectively, these contributions provide a unified perspective on fairness in ML systems. Fairness is framed not as a static training constraint, but as a software engineering problem requiring continuous testing, localization, and repair across the ML lifecycle.

1.3 Thesis organization

Chapter 2 provides all background of the concept used in this thesis, including the different fairness definitions and engineering fairness practice across ML workflow, fault-localization and repair of fairness in DNNs and transformer models, distribution shift problem.

Chapter 3 details the related works on fairness testing and mitigation in classical ML, DNNs and LLM systems, such as the based on model editing of neurons activations. Moreover, we discuss the related works on addressing fairness in DNNs under distribution shifts problem by situating our approach within the broader literature on domain adaptation, self-training, and fairness-aware learning. We note that while fairness in static, in-distribution settings has been extensively studied, fairness under distribution shift remains comparatively under-explored.

Chapter 4 details our first study that propose an approach for systematically identifying all bias- inducing features of a model to help support the decision-making of domain experts.

Chapter 5 details `FairFLRep`, an automated fairness-aware fault localization and repair technique that identifies and corrects potentially bias-inducing neurons in DNN classifiers.

Chapter 6 report our diagnostic study of bias behavior in pretrained transformer models, where we build a dataset of *biased relations* and adapt neuron attribution strategies to trace and suppress biased neurons in *BERT* models.

Chapter 7 details a principled fairness-aware fault-localization and repair framework for mitigating bias in pretrained transformers, built on findings from Chapter 6. Here, instead of using standard neuron attribution, we integrate fairness-aware on both stage from fault-localization and repair. We introduce Pareto-optimal, contrastive neuron localization, which identifies small and high-quality bias-inducing neurons by jointly considering fairness impact and correctness objectives.

Chapter 8 details the final study unified framework, `DShift-FRep`, for fairness-aware repair of DNNs under distribution shifts. The approach integrates fairness-aware label propagation, shift-vector-based data alignment, warm-start fine-tuning, and targeted neuron repair via

PSO to achieve a systematic fairness repair when deploying DNNs across evolving domains.

In Chapter 9, we summarized all the studies presented in this thesis and discus the implications. We also highlights our future research directions and conclude the thesis.

CHAPTER 2 BACKGROUND

2.1 Engineering Fairness

Traditional fairness-aware learning approaches predominantly treat fairness as a model-level constraint, enforced during the training phase through regularization or constrained optimization. However, modern machine learning systems are complex software artifacts, composed of data pipelines, feature engineering logic, trained models, evaluation scripts, and deployment environments. Fairness violations may therefore originate from multiple sources and emerge at different stages of the system lifecycle.

Engineering fairness in machine learning systems requires a socio-technical perspective [34, 44] that goes beyond algorithmic constraints. It involves the systematic and continuous identification, measurement, and mitigation of bias throughout the entire AI workflow [27]– from data collection and preprocessing to model training, evaluation, deployment, and post-deployment operation. This perspective aligns fairness with other software quality attributes such as reliability, robustness, and safety.

In practice, engineering fairness entails: defining fairness objectives in a context-dependent manner, auditing system behavior using appropriate metrics, diagnosing the sources of unfairness, and applying targeted technical interventions at the data, model, or representation levels. This thesis adopts this engineering perspective and studies fairness as a continuous repair problem, rather than a one-time training objective.

2.2 Formal Fairness Definitions

A wide range of formal fairness definitions have been proposed in the literature, each reflecting different normative assumptions and application goals. No single definition is universally appropriate, and many definitions are mutually incompatible. In this section, we briefly review the most commonly used notions of fairness that underpin the methods studied in this thesis.

2.2.1 Group fairness (Statistical/ demographic parity)

Group fairness aims to ensure equal statistical measures for the individual across the group to be equal by imposing the condition on the classifier such that the prediction outcomes for individuals across the group are almost with equal probability. I.e., the probability for

positive or negative prediction is equal.

A binary classifier $h(X)$ satisfies demographic parity if

$$P[h(X) = 1|S = s] \;=\; P[h(X) = 1] \;\forall s$$

where, S denotes the sensitive attribute.

In other words, this definition requires the selection rate (i.e., the probability of predicting the positive outcome) to be equal across all groups. By implication, the rate of negative predictions is also equal. Demographic parity is widely used due to its simplicity, but it does not account for differences in underlying label distributions and may conflict with accuracy or utility in certain settings. Variants of group fairness include Equal Opportunity and Equalized Odds, which condition fairness on the true label rather than enforcing unconditional parity [24]

2.2.2 Individual fairness

This fairness captures the principle that the model return a similar outcome for a similar individual, i.e., a similar individual should be treated similarly. A predictor achieves individual fairness iff

$$H(x_i) \approx H(x_j)|d(x_i, x_i) \approx 0$$

where $d : X \times X \to R$ is a distance metric for individuals. The (D, d)-Lipschitz property can be used to capture individual fairness which states that $D(H(x_i)Y, H(x_j)Y) \leq d(x_i, x_j)$ where D is a distance measure for distributions. However, it is more challenging to compute a metric for individual fairness than group fairness, as 'similarity' needs to be defined in appropriate distance measures in the input feature and the model outcome.

2.2.3 Counterfactual Fairness

Counterfactual fairness requires that a model's prediction for an individual remains unchanged under counterfactual changes to sensitive attributes, holding all other causal factors constant [45]. Formally, a decision is counterfactually fair if:

$$H_{S\leftarrow s}(x) = H_{S\leftarrow s'}(x), \;\; \forall s, s'$$

where $H_{S\leftarrow s}$ denotes the prediction under a counterfactual intervention on the sensitive attribute.

Counterfactual fairness provides strong guarantees but relies on explicit causal models, which are often unavailable or difficult to specify in real-world ML systems.

Taken together, these fairness definitions are often mutually incompatible, and their suitability depends on the application context. Moreover, satisfying a fairness metric during training does not guarantee fairness under distribution shift or deployment changes–an issue central to this thesis.

2.3 Engineering Practices Across the ML Workflow

Fairness must be proactively considered at every stage of the ML lifecycle to address bias at its source and reduce downstream repair costs. We now review how fairness considerations arise across the ML workflow.

2.3.1 Data Engineering (Pre-processing)

This technique modifies the training data to keep the algorithm from discrimination, following the idea that the training data is the root cause of discrimination that an algorithm learns from. The training data might have captured historical discrimination, or there are more subtle patterns in the data, such as under-representation of the minority group that causes bias in that group both most likely and less costly under specific accuracy measures.

Common pre-processing techniques include reweighting or resampling data to balance group distributions, learning fair representations that remove sensitive information, modifying labels or features to reduce bias. While effective in controlled settings, pre-processing methods assume that bias sources are known a priori (identifiable before training) and that data distributions remain stable. In practice, latent biases, proxy effects, and distribution shifts may only emerge after training or deployment, motivating post-training analysis and repair. Nevertheless, this early detection and mitigation can help reduce downstream repair costs.

2.3.2 Model Engineering (In-processing)

This technique usually modifies specific learning algorithms, e.g., by imposing new constraints with respect to some protected attribute such as gender, race, have been by far the most common approach. For instance, by applying the regularization term to a logistic regression classification [46], optimization using convex relaxation proposed by Calders and Verwer [47], modified representation of the data that is most effective at classification while still being free of signals pertaining to the sensitive attribute [48].

Although powerful, these methods typically require access to sensitive attributes, retraining the model from scratch, and assumptions of stationary data distributions. These requirements limit their applicability in real-world systems where models are reused, fine-tuned, or adapted over time.

2.3.3 Model Evaluation and Testing

Fairness evaluation aims to detect disparities in model behavior across groups or individuals. Common practices include, computing fairness metrics on held-out test sets, subgroup analysis, stress testing under controlled perturbations. However, most fairness testing approaches are non-localizing: they reveal that unfairness exists but not why it exists. This gap motivates fairness-aware fault localization, which seeks to identify internal components responsible for observed disparities.

Common Fairness Evaluation Metrics

Fairness in ML/DNNs classification models are quantified/ evaluated using fairness metrics, particularly in terms of how they treat different sensitive groups (e.g., based on race, gender, age) [49]. Fairness metrics are important because they help ensure that the outcomes of machine learning models do not disproportionately favor or harm specific groups, addressing concerns about bias and discrimination. This study will focus on *group fairness*, which seeks for equal statistical measures for the individual across the group according to a particular protected attribute, and statistics about model predictions is calculated for each group and compared across groups. Several group fairness metrics are commonly used:

Statistical Parity Difference (*SPD*): This metric [50] compares the probability of individuals receiving the desired algorithmic decisions (positive outcomes) between an unprivileged class (i.e., $\mathcal{A} = a_0$) and a privileged class ($\mathcal{A} = a_1$), such as race or gender. It defines fairness as different subgroups having an equal probability of being classified with the positive label. Formally, this is expressed as: $SPD := P(\hat{Y} = 1 \mid \mathcal{A} = a_0) - P(\hat{Y} = 1 \mid \mathcal{A} = a_1)$. Ideally, a value of $SPD = 0$ would indicate that the model's prediction decision $\hat{Y}$ is statistically independent of the sensitive attribute $\mathcal{A}$, suggesting a fair model. However, achieving an exact $SPD = 0$ is nearly impossible, so the value is typically compared against a threshold $SPD \leq \epsilon$. The threshold ϵ is determined based on specific real-world requirements. Sensitive parity is independent of the ground truth labels, making it particularly useful in situations where reliable ground truth information is not available, such as in employment and justice contexts. When a ratio is used instead of a difference to represent the equality of predictions across

different groups, the fairness metric is referred to as *Disparate Impact* (*DI*). Formally, it is defined as: $DI := \frac{P(\hat{Y}=1|\mathcal{A}=a_0)}{P(\hat{Y}=1|\mathcal{A}=a_1)}$. In this case, a value $DI = 1$ would indicate a fair model. In real-world situations, the value of *DI* is compared with a fairness threshold, such as $DI \geq \tau$. The choice for τ is usually guided by the social implication, for example, a threshold of 0.8 represents the 80% rule [51]. In this context, the 80% rule states that the selection rate for a disadvantaged (unprivileged) group should be at least 80% of the selection rate for the advantaged (privileged) group. If the ratio falls below 80%, it may indicate potential discrimination or an unfair impact on the unprivileged group. Moreover, to remain consistent with other fairness metrics, *DI* can be converted to $abs(1 - DI)$, indicating that a lower value is better (0 means no bias).

Equality of Opportunity/Equalized Odds (*EOD*): A limitation of *SPD* and *DI* is that they do not consider potential differences in the compared subgroups. *EOD* overcomes this by taking into consideration the distributions of different groups in terms of the true positive rate of the label. This is expressed as $P(\hat{Y} = 1 \mid \mathcal{A} = s, Y = 1) = P(\hat{Y} = 1 \mid Y = 1), \forall s$. Essentially, this compares the true positive rate (*TPR*) across different groups. In other words, the model's predictions should be equally accurate across different sensitive groups when conditioned on the true outcome. A similar measurement can be calculated for the false positive rate (*FPR*) (i.e., the probability of incorrectly predicting a positive outcome): $P(\hat{Y} = 1 \mid \mathcal{A} = s, Y = 0) = P(\hat{Y} = 1 \mid Y = 0)\ \forall s$. By enforcing both true positive rate parity and false positive rate parity simultaneously (i.e., considering both $Y = 1$ and $Y = 0$), we achieve *Equalized Odds* as described by [24]: $P(\hat{Y} = 1 \mid \mathcal{A} = s) = P(\hat{Y} = 1) \forall s$.

Predictive Quality Parity: This metric measures prediction quality difference between different subgroups. The quality denotes quantitative model performance in terms of model predictions and ground truth. In this paper we focus on accuracy for binary classification.

While these metrics provide important tools for assessing fairness in DNNs, they do not always capture all dimensions of bias. Some metrics, like counterfactual fairness [52] and fairness through unawareness [52], causal fairness [53] explore alternative dimensions, but the metrics discussed above form the foundation for measuring group fairness in DNN models. Understanding these metrics is crucial for designing and evaluating fairness-aware systems.

Fault Localization and Repair

Fault localization [54] is a debugging technique aimed at pinpointing the defective components of a program based on testing outcomes. In conventional software, a program component

(e.g., a statement) is deemed highly faulty or suspicious if it is covered by more test cases that reveal failures and fewer test cases that pass. This concept has been extended to other software systems, including DNNs. In [39], a fault localization method and a repair method for DNNs are proposed, which operate as follows:

1. it selects a subset of positive and negative inputs, where positive inputs are instances correctly classified by the model $\mathcal{M}$, and negative inputs are those misclassified by $\mathcal{M}$. Negative inputs reveal the model's incorrect behavior and highlight areas needing correction;
2. the forward impact and the gradient loss of each weight connected to the last layer are calculated. The gradient loss is determined using classic backpropagation. The rationale for focusing solely on the weights of the last layer is to minimize the search space;
3. weights that have both high forward impact and high gradient loss are extracted (i.e., those on the Pareto Front of these two metrics). The rationale is that these weights are likely to be the most responsible for the model's incorrect behavior (high gradient loss) while also significantly influencing its decisions (forward impact);
4. in the repair phase, the identified neuron weights are adjusted to reduce bias and improve fairness within the DNN. The process involves searching for alternative weight values that adjust the model's behavior to improve fairness without causing a notable drop in overall accuracy.

2.3.4 Deployment and Monitoring (Post-processing)

Fairness is an ongoing commitment that extends beyond model training and deployment. Post-processing techniques aim to enforce fairness constraints by adjusting model outputs without modifying the underlying model parameters. These approaches are particularly attractive in scenarios where retraining is infeasible due to computational, legal, or operational constraints.

A common class of post-processing methods takes an existing trained classifier and the sensitive attribute as input, and derives a transformation of the model's predictions to satisfy specified fairness constraints. For example, decision thresholds may be adjusted separately for different sensitive groups to equalize statistical measures such as selection rates or error rates. Another prominent family of methods is reduction-based post-processing, which treats fairness as a constrained optimization problem. These approaches generate a sequence of reweighted datasets and corresponding retrained models, producing a set of candidate classifiers that span the fairness–accuracy trade-off frontier. Practitioners can then select a

model based on application-specific utility functions, business rules, or cost considerations.

Despite their practical appeal, post-processing approaches have several limitations. They often degrade predictive accuracy, require access to sensitive attributes at inference time, and provide limited robustness under distribution shift. Moreover, because these methods operate only on model outputs, they do not address the underlying sources of bias embedded in learned representations or internal model parameters. More broadly, once models are deployed, they are exposed to evolving data distributions, changes in feature semantics, and shifts in the relationship between sensitive attributes and labels. Under such conditions, fairness guarantees established during development may erode over time. These challenges highlight the need for continuous fairness monitoring and targeted repair mechanisms in production environments, rather than one-off post hoc interventions. Fairness is an ongoing commitment even after a model is deployed. Post-processing techniques adjust model outputs to satisfy fairness constraints without modifying the model itself. Notably, post-proposing algorithms takes an existing classifier and the sensitive feature as input and derive a transformation of the classifier's prediction to enforce the specified fairness constraints. Reduction-based post-processing unfairness mitigation algorithms, take a standard black-box machine learning estimator and generate a set of retrained models using a sequence of re-weighted training datasets. Users can then pick a model that provides the best trade-off between accuracy (or other performance metric) and disparity, which generally would need to be based on business rules and cost calculations.

2.4 Fairness Repair in Deep Neural Networks

Bias/unfairness in ML/DNN models can arise when sensitive attributes, such as race or gender, inadvertently influence decision-making, even if these attributes are not provided directly as inputs. For instance, an AI system designed for hiring might inadvertently favor certain genders over others, basing decisions on attributes indirectly correlated with gender instead of qualifications. This type of bias can subtly manifest in the model's internal representations or activations, emphasizing certain features that correlate with protected attributes. Although it is difficult to directly link high-level features to specific model weights, these biased representations can still impact decision-making, potentially skewing outcomes. Bias in ML/DNNs can be broadly categorized into either *prediction outcome discrimination* and *prediction quality disparity* [55].

Prediction Outcome Discrimination

It occurs when DNN models deliver unequal treatment to individuals based on membership in sensitive groups. For example, a recruiting AI could exhibit bias by favoring male candidates over female candidates. Training data often includes biases, whether from inherent noise or annotators' subjective judgments [56]. As DNNs are designed to fit these data, they may reinforce these biases or even develop further implicit associations, amplifying societal stereotypes [57]. This can result in algorithmic discrimination, which has critical implications for access to resources in fields like hiring, lending, and facial recognition. Outcome discrimination can be further classified into:

Discrimination via Input: Discrimination in outcomes can stem from input data even when sensitive attributes, like race or gender, are not explicitly included. DNN models may inadvertently link these attributes through correlated features, leading to biased predictions [58].

Discrimination via Representation: This refers to situations where predictive outcome discrimination is analyzed from a representational perspective, particularly when attributing bias to the input is nearly impossible. For instance, a CNN might identify gender from an image of the retina, even though this information is undetectable by humans. Such cases result in varying intermediate representations for different sensitive groups. Decisions are thus made based on this indirectly encoded group information, which can lead to biased classification outcomes detectable within the model's deep representations.

Prediction Quality Disparity

DNN models sometimes deliver lower-quality predictions for certain sensitive groups, often due to underrepresentation in the training data. When specific populations are either inadequately represented or have less reliable data, the model may attempt to minimize the overall error, often leading to a performance gap between majority and minority groups [59, 60]. While optimizing accuracy for the overall population, this approach can result in subpar performance for underrepresented groups, creating a disparity in prediction quality for these populations.

2.5 Transformer-Based Language Models

Transformer-based language models [61] form the foundation of modern natural language processing systems, including BERT and its variants. A Transformer encoder consists of a

stack of identical layers, each composed of a multi-head self-attention module followed by a position-wise feed-forward network (FFN).

Let $X \in R^{n \times d}$ denote the input token representations for a sequence of length n with hidden dimension d. For each attention head h, self-attention is computed as:

$$Q_h = XW_h^Q, K_h = XW_h^K, V_h = XW_h^V$$

$$\text{Self-att}_h(X) = \text{softmax}(Q_h K_h^T) V_h,$$

Where W_h^Q, W_h^K, W_h^V are learned projection matrices. The outputs of all heads are concatenated and linearly projected to form the hidden representation H.

Following self-attention, each layer applies a position-wise FFN:

$$FFN(H) = GELU(HW_1) \cdot W_2$$

where $W_1 \in R^{d \times d_{ff}}$ and $W_1 \in R^{d_{ff} \times d}$ are learned weight matrices, and d_{ff} denotes the FFN intermediate dimensionality. For clarity, we omit bias terms and normalization layers.

Although self-attention and FFNs differ in activation functions (softmax versus GELU), their structures are closely related. In particular, prior work [37, 62] observes that FFN layers behave analogously to key–value memories: individual FFN neurons selectively activate for specific semantic or factual patterns and contribute additively to the model's output. This interpretation is central to our approach.

2.5.1 Knowledge Representation and Bias in FFN

Recent studies [36, 37] have shown that Transformer FFNs encode linguistic, factual, and social knowledge in a sparse and localized manner. Rather than being uniformly distributed across parameters, many concepts are associated with a small subset of highly influential FFN neurons. These neurons act as reusable feature detectors whose activations are shared across tokens and layers.

Crucially, this neuron-level representation also applies to social biases and stereotypes. Empirical evidence [36] suggests that biased associations—such as those linking demographic attributes to stereotypical behaviors or traits—can be traced to specific internal units. When triggered by particular contextual cues, these neurons disproportionately influence the model's predictions.

This observation motivates viewing bias not merely as an undesirable output pattern, but as

an internal fault: a localized representation that systematically skews model behavior under certain conditions. From this perspective, bias can be localized, analyzed, and repaired using principles similar to those applied in software fault localization and repair [15, 39, 54].

2.5.2 Fairness as Behavioral Property

Fairness in machine learning is commonly defined in terms of disparities across sensitive attributes, such as gender, age, or race. In classification settings, this often involves constraints on group-level metrics (e.g., equalized odds or demographic parity). However, for masked language models (MLMs), fairness manifests differently. In MLMs, the task is not to classify inputs into fixed labels, but to generate token predictions conditioned on context. Bias arises when the model systematically prefers stereotype-aligned completions over neutral or fair alternatives, even when both are linguistically plausible.

Importantly, fairness must be considered under correctness constraints. A model that avoids biased outputs by becoming incorrect or uninformative does not constitute a fair solution. Instead, fairness should be achieved by removing reliance on sensitive or stereotypical evidence while preserving correct task behavior. This motivates a correctness-aware view of bias:

1. *Correct predictions* reveal which internal representations support legitimate task performance.

2. *Incorrect predictions*, especially on neutral (non-stereotyped) inputs, may indicate spurious or biased internal dependencies.

Hence, by conditioning bias analysis on prediction correctness, we can distinguish neurons that: 1. spuriously amplify biased evidence when the model errors, from 2. those that support neutral evidence when the model behaves correctly. This distinction underpins our later definitions of suppression-oriented and amplification-oriented neuron scores propose in Section 7.3.2.

2.6 Distribution shift problem

The fundamental challenge in domain adaptation lies in addressing the differences between the training (source) and test (target) data, which originate from similar but distinct distributions. Traditionally, domain adaptation has been approached in two scenarios: unsupervised and semi-supervised settings, distinguished by whether the target data is completely unlabeled or partially labeled. The unsupervised setting, being more general and challenging, has

garnered significant attention. Most existing methods fall into two categories: 1. instance-based learning [63, 64], which adapts classifiers to account for inter-domain differences, and 2. representation learning [65–68], which seeks domain-invariant representations compatible with deep neural networks.

Representation learning approaches, in particular, have shown promising performance by minimizing divergence metrics like maximum mean discrepancy (MMD) [65] or employing adversarial techniques [66,67] to align source and target domain distributions in latent space. For example, Wasserstein distance has been used in adversarial frameworks [69] to achieve this alignment. However, these techniques focus on training processes and often overlook fairness considerations, particularly under distribution shifts.

Our study diverges from these paradigms by addressing the distribution shift and fairness problem through model repair and minimal retraining rather than the full retraining process, benefiting of both repair and retraining. We propose a fairness-aware model repair technique that directly targets and modifies the problematic neurons in a trained deep neural network. These neurons, identified as responsible for biased predictions across the source and target domains, are repaired using search-based modifications with fairness and cross-domains constraints. This approach explicitly addresses the fairness gap introduced by distribution shifts by leveraging the flexibility of model repair to adapt the trained model to new domain conditions without requiring extensive retraining.

A few recent studies [70–73] have explored fair transfer learning, such as instance-based adaptation under limited sensitive attribute availability or fairness preservation across tasks. Our method uniquely focuses on repairing the internal components of a trained model. By directly tackling the bias-inducing neurons, we ensure fairness-aware adaptation across domains, providing a novel and efficient solution to fairness challenges under distribution shifts.

2.6.1 Distribution shifts via Transportation maps

Transportation maps, provide a powerful framework for explaining and addressing distribution shifts. A distribution shift represents a transformation or relocation of data points (samples) from one distribution (source) to another (target). This conceptualization relies on the assumption that there is a meaningful relationship between the source distribution ($P_{\text{src}}(\mathcal{X})$) and the target distribution ($P_{\text{tgt}}(\mathcal{X})$). Without such a relationship, modeling or explaining the shift becomes more challenging.

The shift can be modeled as a transportation process, represented by a transport map $\mathcal{T}$, that moves samples from the source to the target domain. The concept of transport maps

originates from Monge's formulation of Optimal Transport [74, 75], which seeks a mapping $\mathcal{T}$ that aligns two distributions with minimal cost under a given cost function $c(x, \mathcal{T}(x))$. If x is a sample drawn from the source distribution ($x \sim P_{\text{src}}$), applying the transport map $\mathcal{T}$ results in a new sample that lies in the target distribution ($\mathcal{T}(x) \sim P_{\text{tgt}}$).

Counterfactual Representations

When interpretable features are unavailable, the transport map can be illustrated by presenting pairs of source inputs x and their corresponding counterfactual outputs $\mathcal{T}(x)$. For example, if x is an image from the source distribution, $T(x)$ represents how the image would appear in the target distribution. These counterfactual examples serve as practical explanations of distributional change and provide insight into how feature-level patterns differ across domains.

Fommerly, a transport map $\mathcal{T}$ is designed to map the source distribution ($\mathcal{X}_{\text{src},}$) to the target distribution ($\mathcal{X}_{\text{tgt},}$) such that the push-forward distribution satisfies:

$$T_\psi P_{\text{src}} \approx P_{\text{tgt}}$$

Where, $T_\psi P_{\text{src}}$ denotes the distribution of transformed sample $\mathcal{T}(x)$ when $x \approx P_{\text{src}}$. This formulation, central to optimal transport theory, ensures minimal-cost alignment between distributions.

2.6.2 Controlling for Domain Shift

Domain shifts refer to variations in the underlying characteristics or environmental context of data across domains, a topic widely examined in domain adaptation research. These variations are generally classified into visual and contextual categories, as discussed in prior studies on adversarial domain adaptation and style transfer (e.g., [66, 67, 76]). Common examples include: 1. *Visual changes*, involves alterations in image properties such as illumination, surface texture, or image resolution. 2. *Contextual shifts*, arising from differences in background scenes, object placement, or the presence of previously unseen categories.

Feature-level transformation $\mathcal{T}(x)$ highlight the attributes–such as skin tone, facial contour, lighting conditions, or higher-level semantic patterns–that differ most significantly between the source and target domains. In face recognition or demographic classification tasks (e.g., gender or race prediction), such feature-level interpretations shed light on how the alignment process captures both visual and contextual domain shifts. They further help identify which transformations enhance cross-domain consistency and which may unintentionally reinforce

disparities between demographic groups.

2.6.3 Controlling for bias shift

This thesis addresses fairness under distribution shift by combining adaptation mechanisms with targeted repair, ensuring stability across evolving domains. Bias arises when the relationship between features and outcomes differs across the source and target distributions with respect to sensitive groups. This phenomenon has been widely studied in the context of fairness under distribution shifts (e.g., [70, 71, 73]), emphasizing the need for approaches that explicitly account for these disparities. Such discrepancies can lead to unfair or undesirable outcomes, particularly when specific demographic groups are disproportionately affected.

During cross-domain alignment, it is essential to ensure that data transformations or feature adjustments:

Feature-Specific Shifts: Encourage T to generate counterfactual examples $T(x)$ that: 1. Avoid systematic shifts or disproportionate effects on sensitive groups [77]. 2. Mitigate bias by ensuring that transformed representations are consistent in quality across different demographic groups. Example: If images of individuals from a specific demographic are consistently transformed into representations of comparable quality, it helps identify and reduce biases. Purpose: Refine the transport map to prevent undesirable shifts, fostering more robust and fair machine learning models.

2.7 Summary

This Chapter presented the conceptual and technical foundations that underpin the thesis. We began by framing fairness as an engineering concern in ML systems, emphasizing that unfairness can emerge at multiple stages of the ML lifecycle and should be addressed as a continuous analysis and repair problem rather than a one-time training objective. We reviewed widely adopted formal fairness definitions, including group fairness, individual fairness, and counterfactual fairness, highlighting their normative assumptions, practical trade-offs, and mutual incompatibilities. This discussion motivated the need for context-aware fairness evaluation and for methods that remain effective beyond the training phase, particularly under distribution shift.

The Chapter then surveyed fairness-aware practices across the ML workflow, covering preprocessing, in-processing, evaluation and testing, and post-processing techniques. While existing methods provide valuable tools for mitigating bias, we highlighted their limitations in terms of localization, robustness, retraining requirements, and long-term deployment sta-

bility. These limitations motivate fairness-aware fault localization and repair as a complementary paradigm.

We further discussed fairness repair in DNNs, distinguishing between prediction outcome discrimination and prediction quality disparity, and argued that many forms of bias are embedded in internal representations rather than solely in inputs or outputs. This perspective naturally aligns with repair-based approaches that target internal model components. Building on this, we introduced Transformer-based LLMs and reviewed evidence that knowledge and social biases are sparsely encoded in FFNs. Treating biased neurons as internal faults enables principled analysis and targeted intervention. We also emphasized fairness as a behavioral property that must be evaluated under correctness constraints, particularly for MLMs. Finally, the Chapter examined fairness under distribution shift, introducing transportation maps as a conceptual tool for understanding domain shifts and bias shifts. We argued that fairness guarantees established during training may erode across domains and that targeted model repair provides a practical mechanism for maintaining fairness under evolving data distributions. Together, this background establishes the theoretical, methodological, and engineering foundations for the fairness-aware localization, repair, and adaptation techniques developed in the subsequent Chapters.

CHAPTER 3 LITERATURE REVIEW

3.1 Bias and Fairness

The analysis of bias has been approached from different directions, including legal reasoning, sociological theories, statistical methods, and economic models [78–81]. Some previous researchers such as [82–84] have studied how to prevent bias in the data mining process. In this study, instead, we focus on the detection of bias in a dataset involving historical decisions, and the literature presented the most relevant literature.

3.1.1 Testing and Mitigating Direct Bias/ Unfairness

Perera et al. [22] proposed a novel search-based fairness testing (SBFT) approach based on the concept of fairness degree to test for fairness and evaluate the fairness of regression-based ML systems. Their proposed SBFT approach works by computing the maximum difference in the predicted values by the machine learning system for all pairs of identical instances apart from the sensitive features to describe the worst-case behavior of the system.

Chakraborty et al [21] proposed a Fair-SMOTE algorithm to test for the root causes of bias in the prior decisions about the data selection and the labeling assigned to the data. Then, a mitigation technique is proposed to solve the data imbalance. The steps include dividing the data based on subgroups based on class and protected features defined by the sensitive features and then generating synthetic data points for all the subgroups except the subset with the maximum number of data points. The fair-SMOTE algorithm uses the situation testing technique to test how the labeling can induce bias in the model by flipping the value of sensitive features for every data point and computing the propensity score to estimate the impact on the model prediction.

Aggarwal et al. [85] uses the combination of symbolic execution and local explainability for automatic generation test case auto-generation detecting individual bias in machine learning models. Given the machine learning model, domain constraints, and a protected attribute set, the aim is to generate test cases to maximize the successful test cases defined by the protected attribute leading to a bias behaviour for different combinations of protected attribute values.

Tramèr et al. [86] proposed FairTest, a fairness testing approach that uses manually written tests to measure four types of discrimination scores. The authors leverage the idea of identifying the association between protected features and model prediction (e.g., salary is related to age) to generate test cases. To enable the interpretability of black-box predictive models,

they follow an iterative procedure based on an orthogonal projection of input features which allows quantifying the relative dependence of a black-box model on its input attributes. The relative significance of the inputs is then used to assess the fairness of the predictive model under test.

Clearly, the above approaches assume the prior knowledge of the sensitive features in the dataset under analysis. The sole idea of the above approach is to analyze and mitigate the bias defined by the protected subgroup of these sensitive features to estimate the direct discrimination. Similarly other previous studies have mainly relied upon simple statistical analysis involving association or correlation measures [87–89]. However, such analyses can lead to incorrect conclusions because they largely ignore the effect of confounding variables – variables that can be used to determine both the outcome and the feature pairs. In other words, quantifying bias through such analyses can distort the causal effect of belonging to the protected or unprotected features.

3.1.2 Testing and Mitigating Indirect Bias/ Unfairness

Next, we present the relevant literature for indirect distribution analysis.

Mancuhan and Chris [90] propose Bayesian networks as a technique for modeling the probability distribution of a class to identify discrimination. The method includes discovering all the dependencies between attributes and using these dependencies to estimate the joint probability distribution.

Ruggieri et al. [91] introduce a reference model, a form of rule inference for classification then analyze those rules to help discover the indirect discrimination of Automatic Decision Support Systems (DSS). The notion of itemsets, association rules, and classification rules are used to encode the background knowledge about the feature correlation. The itemset was used to represent the sensitive features e.g., $sex = female$, $age = older$, or $race = black$. The associated rule will combine the classification rule of these itemset $sex = female$, $age = older$, $car = own \rightarrow credit = no$, and potentially biased subgroup (i.e., the intersection of itemset consisting of only the protected feature value).

Our work deviates from the above literature in that:

- The thesis proposed technique is based on the concept of probabilistic causal instead of defining the correlation.
- Our technique can be used to detect both direct and indirect bias as compared to the above methods that focus on the specific type of bias.

- The approach applies to both classification and regression models.
- Our work does not require prior knowledge of the sensitive features; instead, we identify all the features that are potentially inducing bias to the model, and the domain expert makes the decision.

3.2 Fairness-Aware Repair of DNN systems

Li et al. [16] introduce Faire, a method that aims to repair fairness in neural networks by focusing on neuron conditions and modifying the model's internal structure to mitigate bias. While this approach shares some similarities with our proposed `FairFLRep` technique, there are key differences and limitations of Faire in comparison to our method: (i) The Faire method primarily targets individual fairness and is designed for tabular data, which presents significant challenges when trying to apply it to image classification tasks like the ones we are addressing in this study. In the context of group fairness for DNNs on image datasets (e.g., `UTKFace`, `FairFace`), Faire's focus on individual fairness rather than group-level disparities (e.g., race, gender) makes it less applicable. (ii) Faire uses DeepLIFT, an external framework for analyzing neuron contributions, to identify biased neurons. While DeepLIFT is effective for neuron attribution, its integration into the Faire workflow introduces additional complexity when adapting it for different model architectures or fairness scenarios. `FairFLRep`, in contrast, performs neuron analysis directly based on gradient-based fault localization, which is more integrated with the fairness repair process and does not require external frameworks.

`Arachne` [39] is a more recent method designed to repair DNNs by addressing misclassification errors through fault localization and model repair techniques. Although not originally intended for fairness, `Arachne` can be adapted by prioritizing input data from deprived subgroups to identify neurons linked to biased predictions. However, `Arachne` focuses primarily on general model repair and not fairness-specific issues. When adapted for fairness, it may overlook more complex bias patterns, especially if not properly tuned for fairness-specific tasks. Maheshwari and Perrot [17] present FairGrad, a method that integrates fairness considerations directly into the training process by modifying the gradient descent optimization. Its main limitations are: (i) FairGrad modifies the optimization algorithm, which may limit its application to models where gradient-based optimization is appropriate. In contrast, `FairFLRep` focuses on neuron analysis and repair, targeting specific areas of the model that contribute to unfair behavior, providing more targeted intervention at the neuron level. (ii) FairGrad applies fairness at the gradient level, affecting all weights in the model dur-

ing training. FairFLRep takes a more localized approach, identifying and modifying specific neurons responsible for bias, allowing for finer-grained control over fairness while preserving model performance. (iii) While FairGrad integrates fairness into the overall model optimization, it does not explicitly distinguish between the fault localization and repair stages as FairFLRep does. FairFLRep provides a two-stage process: first identifying the unfair behavior (fault localization), and then applying targeted repairs, which may result in a more precise fairness intervention.

FairGRAPE proposed by Lin et al. [18] is a technique designed to mitigate bias in DNNs, particularly in the domain of face attribute classification. The method introduces fairness into the model by pruning certain neurons, based on gradient information, to reduce unfair behavior while maintaining the model's overall performance. Model pruning—the process of removing unnecessary or redundant neurons to make the model more efficient—is a well-known technique in deep learning for reducing model size and complexity. FairGRAPE extends this idea by incorporating fairness into the pruning process, aiming to reduce bias along with the usual benefits of pruning, such as lower computational costs and memory usage. The pruning process in FairGRAPE is driven by fairness considerations rather than just model efficiency, ensuring that fairness is improved as neurons are pruned. Its main limitations are: (i) FairGRAPE removes neurons that are associated with biased behavior. In contrast, FairFLRep focuses on repairing the faulty neurons responsible for unfair behavior, which may result in finer control over the fairness intervention. FairFLRep does not necessarily reduce the model size but adjusts neuron weights to mitigate bias. (ii) FairGRAPE is constrained by its pruning approach, meaning that it reduces the model's capacity by removing neurons. In contrast, FairFLRep takes a broader approach by identifying and repairing unfair neurons, allowing the model to retain its full capacity while still improving fairness. (iii) FairGRAPE needs to be applied during training, making it less flexible for situations where a pretrained model must be repaired for fairness. On the other hand, FairFLRep can be applied post-training, offering greater flexibility for real-world use cases where retraining a model may not be feasible.

FairNeuron [19], introduced by Gao et al., improves DNN fairness through adversarial training, focusing on selective neurons that contribute to biased predictions. FairNeuron provides a lightweight approach that is able to scale to large models, and showed its effectiveness on three datasets. Despite its advantages, there are still limitations: (i) FairNeuron uses adversarial training, which requires an additional adversarial network. This adds complexity, while FairFLRep focuses on post-training repair, which is more straightforward and applicable to pretrained models. (ii) FairNeuron is primarily designed for MLP architectures with tabular inputs. Its direct applicability to CNNs or hybrid models is limited due to assump-

tions about fixed neuron layer sizes. Additionally, FairNeuron assumes the availability of two types of labels—binary ground truth labels and regression-style labels (e.g., 1 to 10) used for path-based contribution analysis. Such label configurations may not be available in many real-world settings, particularly in binary classification tasks. In contrast, `FairFLRep` generalizes across diverse architectures, including convolutional models and domain-shifted scenarios (e.g., image datasets), making it suitable for a broader range of real-world applications. (iii) The simultaneous training of both the main network and the adversary in FairNeuron increases computational overhead. `FairFLRep` avoids this by focusing on fault localization and repair without the need for an adversarial network.

MOON (Multi-Objective White-Box Test Input Selection) [92] is a spectrum-based fault localization technique originally developed for test input selection. It identifies suspicious neurons contributing to erroneous decisions by analyzing neuron activation spectra and employs multi-objective optimization to select discriminative inputs. These inputs are then used to retrain the model in an attempt to mitigate bias. While MOON is effective for identifying problematic inputs and exposing model weaknesses, its fairness repair mechanism—retraining based on test input activation patterns—is inherently indirect. It does not directly target or adjust the weights of the neurons responsible for biased outcomes. Moreover, its reliance on statistical correlations from neuron activation frequencies limits its ability to capture causal dependencies driving group-level disparities. In contrast, `FairFLRep` performs gradient-based fault localization and explicitly identifies and modifies biased neuron weights, enabling precise and effective fairness interventions without full retraining. As such, while MOON contributes to fairness-aware testing, its repair mechanism is less fine-grained and insufficient for deeply embedded biases in complex DNNs.

`NeuronFair` [93] identifies biased neurons and employs white-box fairness testing techniques for better interpretability in model decision-making. Unlike `NeuronFair`, which remains primarily diagnostic, `FairFLRep` offers an actionable repair phase, directly mitigating the biases it identifies rather than merely reporting them.

`Fairify` [94] introduces a framework for verifying fairness properties in neural networks using formal methods. `Fairify` aims to provide formal guarantees by encoding fairness constraints (i.e., individual fairness) into SMT problems and systematically verifying whether the model satisfies these constraints across all possible inputs. However, `Fairify` does not allow to improve the model (as done by `FairFLRep`) once unfairness is detected. Moreover, `Fairify` primarily targets smaller, fully connected networks due to the computational complexity of formal verification, limiting its scalability for deep architectures like CNNs. In contrast, `FairFLRep` is designed for efficient post-training fairness repair in large-scale deep models.

Latent Imitator (LIMI) [95] is a method for generating natural discriminatory instances for black-box fairness testing. LIMI approximates a surrogate decision boundary in the latent space of a generative model and probes samples near this boundary to generate discriminatory test inputs that retain naturalness and semantic realism. To further demonstrate the importance of the naturalness of the generated instances, the authors of LIMI conducted retraining experiments, combining the generated discriminatory instances with the original training data. They then evaluated fairness improvements using metrics such as IFR (Individual Fairness Rate), IFO (Individual Fairness Outliers), *SPD*, and *EOD*, reporting noticeable improvements in model fairness. However, it is important to note that while LIMI contributes meaningfully to fairness testing and demonstrates that retraining on targeted samples can enhance fairness, its focus remains primarily on generating test inputs and probing latent spaces, rather than directly identifying or repairing specific fairness faults in the model's internal decision-making processes, as FairFLRep does. Moreover, LIMI's reliance on surrogate boundary approximation may introduce variability, and its performance may fluctuate depending on data complexity and the quality of the underlying generative model, potentially limiting its robustness across different real-world scenarios. MetaRepair [96] proposes a meta-learning framework that learns to repair DNNs by leveraging prior repair experiences. It trains a meta-repair model that generalizes across repair tasks and can quickly adapt to new models or faults using few-shot learning. Unlike FairFLRep, which directly targets fairness-related faults, MetaRepair focuses on general model errors and the repair strategy does not explicitly consider fairness. Moreover, MetaRepair requires a meta-training phase with access to multiple faulty models, where as FairFLRep operates directly on a given pretrained model without needing prior repair knowledge. Jain et al. [97] propose a performance-aware fairness repair framework that uses Auto-Sklearn AutoML toolkit [98] to automatically search for fairness-preserving model modifications while maintaining high predictive performance. However, this approach is limited to classical machine learning models (and requires retraining or architectural changes) and is not applicable to DNNs.

Das et al. [99] introduce a multi-task CNN framework that jointly learns to classify gender, age, and ethnicity while mitigating inter-group bias. By exploiting correlations among these tasks and balancing feature learning across groups, the model improves fairness in visual attribute classification during model training.

Hendricks et al. [100] address bias in image captioning models, specifically gender bias arising from co-occurrence patterns in training data. It uses training and counterfactual data augmentation strategy to reduce bias in gender-specific captions without degrading overall caption quality. This work targets semantic bias in generative models, while FairFLRep repairs fairness in classification tasks.

Overall, `FairFLRep` stands out by offering a more integrated, flexible, and precise approach to fairness repair. It provides neuron-level analysis and repair without the need for retraining or external frameworks, making it more applicable in real-world scenarios where group fairness and model flexibility are key concerns.

3.3 Model Editing and Fairness of LLM

Recent advances in interpretability and model editing [37, 101, 102] have shown that the behavior of LLMs can be traced to localized internal components, particularly neurons in FNNs [37, 62]. This section reviews prior work on neuron-level knowledge representation, bias manifestation in LLMs, and neuron editing techniques for fairness. Together, these studies motivate our approach of treating biased associations as localized internal faults that can be analyzed and selectively intervened upon without retraining the entire model.

3.3.1 Neuron Activations and Knowledge Neurons

Building on the intuition that FNNs in Transformer networks are interpreted as key–value memories, where intermediate neurons act as keys whose activations determine how stored values are retrieved. Dai et al. [37] introduced the concept of *knowledge neurons*. Their approach leverages the cloze task: given a factual triplet <*head, relation, tail*> (e.g., <*Ireland, capital, Dublin*>) [103], the model is prompted with a masked sentence such as *"The capital of Ireland is [MASK]"*, and attribution is computed for the prediction of the masked token. To quantify the contribution of each neuron, they proposed a knowledge attribution method based on integrated gradients [104], which measures how changes in neurons' activation influence the probability of a correct answer.

Since individual prompts may activate neurons spuriously due to lexical overlap, Dai et al. introduced a *refining strategy*. For each fact, multiple paraphrased prompts are used; only neurons consistently ranked as salient across diverse prompts are retained. This procedure isolates neurons that robustly encode the underlying relation, rather than superficial cues.

Crucially, the authors also demonstrated that manipulating these neurons—by suppressing their activations (setting them to zero) or amplifying them—causally alters the model's predictions. Suppression significantly decreases the probability of recalling the fact, while amplification increases it, often without substantially affecting unrelated knowledge. These results suggest that *a small number of neurons act as causal carriers of specific knowledge within pretrained Transformers*. Starting from the hypothesis that stereotypes are encoded as knowledge within such models, **our work builds on their methodology and trans-**

fers it to biased relations, testing whether harmful stereotypes are similarly localized and whether their suppression can reduce bias expression while preserving task performance.

3.3.2 Bias/ Unfairness of LLM

With the rise of LLMs, whose scale and general-purpose nature exacerbate the risks of bias [105], ensuring fairness in practice remains a significant challenge. Recent investigations show that LLMs reinforce stereotypes across multiple domains, exposing tangible risks in real-world decision-making [106–110]. For instance, Khan et al. [107] found systematic gender stereotyping in occupational associations (e.g., linking *nurse* to women, *engineer* to men). Other studies highlighted further disparities, such as socioeconomic biases in text generation [109] and discriminatory treatment of African American English speakers [110]. Within SE, fairness concerns are no less pressing. LLMs are increasingly applied to developer-facing tasks such as recruitment, role assignment, or requirements analysis, where biased predictions may shape team composition and project outcomes [111, 112].

3.3.3 Neuron Editing for Fairness

The ability to trace and manipulate neurons responsible for specific knowledge raises the question of whether similar techniques can be leveraged to address harmful stereotypes. Early work on *knowledge neurons* [37] showed that factual associations could be localized within small subsets of feed-forward units, and that suppressing or amplifying these neurons causally influences model predictions. More recently, attempts such as BiasWipe [113] and interpretable neuron editing by Yu et al. [114] extended these ideas to fairness, demonstrating that pruning or editing biased weights can reduce social bias while preserving overall model accuracy. These methods share the principle that internal mechanisms—not only inputs or outputs—can be targeted to mitigate unwanted behaviors. More closely, Xu et al. [115] proposed BiasEdit, which learns lightweight editors to adjust parameters for debiasing pre-trained transformers globally. Our approach, however, traces the prevalence of biased associations to specific neurons and selectively suppresses their activations. Moreover, while BiasEdit is evaluated only on general NLP bias benchmarks, we extend the analysis to fairness-sensitive SE tasks, introducing a novel dataset of biased relations spanning nine categories.

In general, related work remains limited in two important ways. First, prior neuron editing methods concentrate on gender bias or toxic language, while little is known about whether *biased relations across multiple social dimensions* (e.g., age, disability, socioeconomic status) are similarly encoded in neuron-level structures. This is critical, as recent evidence shows

that even small-scale editing of general neurons can disrupt core capabilities of LLMs [114], raising concerns about the trade-off between fairness and task performance. Second, existing work has primarily focused on broad NLP benchmarks such as toxic content detection or synthetic bias datasets [115], without evaluating fairness in real, practical contexts.

3.4 Distribution Shift and Fairness Transfer

This section reviews existing work related to fairness under distribution shift, situating our approach within the broader literature on domain adaptation, self-training, and fairness-aware learning. Consistent with Section 2.6, we focus on unsupervised and semi-supervised settings, where the target distribution differs from the source and fairness guarantees learned during training may no longer hold.

3.4.1 Self-training and fairness transfer

Recent work by Bang et al [71] explores fairness transfer under distribution shift through a self-training framework motivated by intra-group expansion assumptions. Their approach introduces fairness-aware consistency regularization and evaluates theoretical claims using synthetic datasets with simulated shifts. Similar to other self-training methods, this approach depends on carefully designed transformation sets, which may be difficult to define in realistic domains characterized by complex transport maps. Related studies investigate fairness in self-supervised and self-training settings but primarily focus on improving fairness within the same distribution rather than across domains [116–118]. As such, they do not directly address bias shifts induced by feature-level transformations between source and target distributions.

3.4.2 Fairness transfer under distribution shift

While fairness in static, in-distribution settings has been extensively studied, fairness under distribution shift remains comparatively under-explored. Prior work can be broadly categorized into five lines of research. 1. *Group-wise distribution matching.* Several works attempt to preserve fairness under distribution shift by aligning group-conditional distributions across domains. For instance, the line of work that derive theoretical upper bounds on target-domain fairness violations, motivating joint optimization of source-domain fairness and group-level feature distribution alignment between source and target domains [119].Other approaches adopt optimal transport–based distances, such as Wasserstein distance, to match group distributions [120]. While conceptually aligned with the transport-based view, these methods typically require access to labeled target-domain data, making them impractical in

unsupervised or settings with scarce labeled target data. Moreover, they inherit the general limitations of distribution matching approaches, particularly under severe domain shifts. 2. *Causal inference–based methods.* Causal approaches leverage structural assumptions about the data-generating process to achieve robustness under anticipated distribution shifts [70]. By exploiting invariances in causal mechanisms, these methods can support domain adaptation and robustness guarantees. However, they rely heavily on access to an accurate causal graph, which is often unavailable or overly simplistic in real-world applications. Empirical evidence shows that practical settings—such as medical or biometric decision-making—exhibit complex causal dependencies that violate standard assumptions [121]. 3. *Reweighting-based methods.* When group proportions change across domains, instance reweighting has been proposed to approximate the target distribution using source data. Such methods have been applied to fairness under covariate shift [122] and demographic shift [123]. However, reweighting assumes sufficient support overlap between source and target distributions—an assumption frequently violated under domain shifts modeled by transportation maps, where samples may be relocated to previously unseen regions of the feature space. 4. *Distributionally robust optimization (DRO).* Another line of work aim to learn models that are robust to worst-case distributional changes by optimizing fairness objectives over uncertainty sets defined around the source distribution [124, 125]. These approaches primarily target subpopulation shifts rather than full domain shifts and do not explicitly model feature-level transformations induced by transport maps, limiting their applicability in settings with substantial visual or contextual changes.

3.4.3 Distribution Shift and Domain Adaptation

Motivated by early theoretical foundations on domain adaptation by Ben-David et al. [126], numerous feature-level distribution alignment methods have been proposed. Domain-adversarial neural networks (DANN) [66] and related variants [67,127,128] seek to learn domain-invariant representations via adversarial training and have demonstrated success in certain applications. However, subsequent analyses reveal that such methods often optimize only loose theoretical bounds and can fail under realistic shifts [129–131]. Our experiments similarly show that representative distribution-matching methods, including DANN and its variances, are limited in their ability to transfer both accuracy and fairness. Moreover, these approaches do not consider the risk of unfair outcomes after adaptation.

3.5 Summary

This Chapter reviewed related work on fairness in machine learning systems, with a particular focus on fairness-aware repair, interpretability, and model editing techniques for DNNs and LLMs. We surveyed existing approaches for testing and mitigating direct and indirect bias in classical ML systems, highlighting key limitations. In particular, many prior methods assume prior knowledge of sensitive features, rely heavily on simple statistical association or correlation analyses, and often overlook the influence of confounding variables, which can lead to misleading or incomplete conclusions. In contrast, the technique proposed in this thesis is capable of detecting both direct and indirect bias, applies to both classification and regression models, and does not require prior knowledge of sensitive features. Instead, it identifies features that are potentially inducing bias, leaving the final determination to domain experts. This design choice increases flexibility and practical applicability in real-world settings.

The Chapter further discussed fairness-aware repair methods for DNNs that conceptualize unfair behavior as a correctable fault. We showed that many existing approaches operate at coarse granularity, depend on retraining, or require external frameworks. By comparison, the approach proposed in this thesis offers a more integrated, flexible, and precise fairness repair strategy, enabling neuron-level analysis and repair without retraining, while preserving model performance. We then examined interpretability and neuron-level analysis techniques, demonstrating that both factual knowledge and social biases in LLMs are often sparsely encoded in identifiable neurons, particularly within FNNs. Finally, we reviewed model editing and neuron intervention methods that establish the feasibility of selectively modifying internal model components to alter behavior.

Collectively, the reviewed literature and the contributions of this thesis provide a unified perspective on fairness in ML systems, framing fairness as a software engineering problem that requires ongoing testing, fault localization, and repair under evolving conditions.

CHAPTER 4 DETECTION AND EVALUATION OF BIAS-INDUCING FEATURES IN MACHINE LEARNING

This chapter presents the first study of the thesis, which proposes a systematic approach identifying all bias-inducing features in ML model to help support the decision-making of domain experts. The approach is based on the idea of swapping the values of the features and computing the divergences in the distribution of the model prediction using different distance functions. We evaluated our technique using four well-known datasets to showcase how our contribution can help spearhead the standard procedure when developing, testing, maintaining, and deploying fair/equitable machine learning systems.

4.1 Context of the study

The cause-to-effect analysis can help us decompose all the potential causes of a problem. This implies that one can investigate all the possible causes until the root cause. The advantage of this is to propose a fix based on the cause ranking, decompose or simply a complex problem, visualize the different general causes, and where to put more focus. In the context of machine learning, cause-to-effect analysis has been used [132–136] to understand the reasons for the biased behavior of machine learning systems.

In this study, we propose an approach for systematically identifying all bias-inducing features of a model to help support the decision-making of domain experts. Specifically, we propose a novel single feature swapping and double feature swapping functions to help estimate the direct and indirect impact of each feature on the model prediction. The single feature swapping function modifies the values of the single feature keeping other features unchanged, and the function's output is used to estimate the direct impact of the feature on the model prediction. The double features swapping function will alter the values of pairs consisting of the feature and all the mediating variables (following the temporal priority ordering [137]) used to estimate the total natural impact of the feature on the model prediction. The impact of swapping the feature's values on the model predictions is studied by computing the statistical difference in the distribution of the model prediction before and after swapping data, using four different distance functions. Finally, we validate our technique by performing multiple empirical experiments using four well-known datasets to demonstrate the usefulness of our contributions, based on the following two main research objectives:

- **Identify features that potentially introduce bias to the model:** The first objec-

tive was to demonstrate how our proposed swapping functions can be used to identify the features that directly and indirectly introduce bias to the model.

- **The important features to the model:** Given these identified bias-inducing features, we wanted to assert whether or not they are important to the model. By feature importance, we determine the relative importance of each feature in the dataset on the model performance;— by assessing the impact of each feature on the model prediction without necessarily comparing the impact across the sub-group (coarse-grained), as for the case of bias assessment (fine-grained) [138]. We also refer the reader to [139], for further reading about the difference. To this end, we contrast our proposed techniques with the SHAP Values (an acronym from Shapley Additive exPlanations), a model explainable method to explain the individual prediction based on game theoretically optimal Shapley values. Notably, we want to demonstrate that treating the concepts of feature importance and bias inducing features as separate is essential in making informed decisions about what features can be most relevant or least relevant, when building a fairer ML model. Providing insights into what features are most relevant or least relevant to the model prediction and are least bias-inducing as interpreted by the domain experts will help the domain expert choose the features that improve both the predictive performance and fairer model.

Through the above two research objectives, we demonstrate empirically that the features potentially bias-inducing to the model and are less important to the model can be removed to improve the fairness of the machine learning model. We believe that is the first step to making an informed decision by the domain experts through the systematic identification of all bias-inducing features. Moreover, This will help visualize the cause of bias in the model, systematic features selection, prioritizing the fixes, and contributing to the standard procedure when developing, testing, deploying, and maintaining fairer machine learning systems.

Chapter organization. Section 4.2 provides details about the contribution of our study. In Section 4.3 we introduced the main notations used in this paper followed by the the problem definition and the proposed solutions addressing the problems. Next, we detailed the concept of a counterfactual approach to causal inference on which our method is based in Section 4.5. In Section 4.6, we detailed the proposed approach for systematically identifying bias-inducing features of the model. Section 4.7 provides the empirical study to validate our proposed techniques following two main research questions. Section 4.8 presents the results of our experiments on identifying the bias-inducing features and the important features of the model. Section 4.8 evaluate our framework— demonstrating that, treating bias-inducing

and feature importance as separate is essential when building a fairer ML model. Section 4.9 contains the discussion on how our framework can be used to diagnose and fix an unfair model in a given context. as well as the possible threat to the validity of this work. Finally, Section 4.10 concludes the paper.

4.2 Contribution of this Study

In summary, the following are the main contributions of our study:

- First, we investigate ML model bias following the counterfactual approach to causal inference. Some of the concepts of the counterfactual approach to causal inference are discussed in this paper.
- Next, a bias detection technique is proposed based on a novel single feature and double features swapping function to detect bias features in the dataset and ML model.
- Thirdly, an evaluation method is proposed based on divergence measurement to evaluate the impact of biased features on the ML model.
- We validate our proposed technique by performing multiple experiments using four different well-known datasets.
- We showed with the help of the state-of-the-art model explainability tool that the potential bias-inducing features that are less important to the ML model can be removed from the dataset to improve the fairness. We have shown empirically that treating the two entities (bias-inducing and feature importance) as separate is essential in making an informed decision.
- Our study is the first step in helping domain experts make an informed decision by following a systematic identification of bias-inducing features. The domain experts can use our approach to visualize the cause of bias in the ML model, systematic features selection, prioritizing the fixes, and therefore helping contribute to the standard procedure when developing, maintaining, and deploying fairer ML systems.

4.3 Notation and Definition

We employ a dataset D_{test} consisting of independent and identically distributed (i.i.d) samples of random variable $\{(X_i, Y_i)\}_{i=1}^{n}$ from a joint distribution $\rho(X, Y)$ with domain $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^m$, i.e. our input space is a vector with m features. We let X_{ij} be the jth feature

of the ith instance. For example, X_i can be the record of ith loan applicant and X_{ij} can either be the age, gender, race, or credit history of this applicant. We also let $\hat{Y}$ be a model prediction i.e. $f(X) \rightarrow \hat{Y}$, where f can either be a classification or regression model. We can store all measurements of the input in the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ by using the convention that X_{ij} is the ith row and jth column of $\boldsymbol{X}$.

We will sometimes represent the ith row of $\boldsymbol{X}$ as $[X_{i1}, X_{i2}, ..., X_{ij}, ..., X_{im}]$. We will also think of each feature j as being binarized meaning X_{ij} can only take one of two values ($X_{ij} \in (C_{1j}, C_{2j})\ \forall i$), where one category is considered more protected than the other. Furthermore, we assume that there exists a subset of features $\mathcal{S} \subseteq \{1, 2, \ldots, m\}$ that are considered sensitive meaning that the jth column of $\boldsymbol{X}$ with $j \in \mathcal{S}$ potentially introduce bias in the model. For the target variable, we will be using $\boldsymbol{Y}$ to denote vector containing the true label (i.e., class label), i.e., $\boldsymbol{Y} = [Y_1, Y_2, ..., Y_i, ..., Y_n]^T$.

4.4 Problem Statement and Research questions

The goal of this study is to detect the potentially biased features and empirically evaluate the bias in machine learning using the proposed framework described in Section **??**. In the sections, we used the proposed framework to detect the biased features and evaluate the machine learning using the real-world dataset. Specifically, we want to evaluate our technique based on the following two research questions:

RQ1: **Can we identify which features potentially introduce bias to the model?**

This research question aim to examines the features that directly and indirect introduces bias to the model using our proposed bias detection techniques.

RQ2: **Given these identified bias inducing features, can we assert whether or not they are important to the model?**

This research question aims to examine if the bias-inducing or least bias-inducing features are indeed important features to the model. Using the SHAP value, we want to demonstrate how our proposed swapping functions can be integrated with the state-of-the-art model interpretability tool to help explain the most relevant and the least bias-inducing features. The insights that will be derived from answering this question can help the domain experts choose the features that improve predictive performance while making the models the least biased.

By answering the above research question, we demonstrate the potential application of our techniques to detect bias in various datasets and machine learning models that are considered

biased. We hope that our results will help design explainable and more reliable machine learning models.

4.5 Counterfactual Approach to Causal Inference

This section will briefly review the approach of using counterfactual to causal inference and probabilistic causation, on which our method is based. For the detailed discussion on this topic we refer the readers to the literature [132, 134–137, 140–144].

Researchers [132, 134–136] have used the counterfactual approach to causal inference when trying to identify the presence or absence of bias between an observing feature (e.g., a male gender) and model outcome (e.g., loan approval). To this end, the researchers try to understand the outcome of the given predictor when the alternative value of the observing feature is used (e.g., female, for the gender feature). In other words, they answer the counterfactual question: What would have happened to the outcome of this system if the applicant was a female instead of a male? Answering this question is fundamental to be able to conclude that there is a causal relationship between some specific feature and discrimination, which, in turn, is necessary to conclude that discriminatory behaviors or processes contributed to an observed differential outcome. While this question might not be directly answered, observing the causal relationship between the feature under investigation and discrimination can be ascertained. When understanding causal relationships, one can alter the value or turn on/off the value of the feature under investigation leading to multiple outcomes observed for a single profile, hence answering the above counterfactual question with certainty. Indeed, altering the value of the given feature and monitoring the system outcome is a potential interpretation of causality. However, it's nearly impossible to measure causality in the real world systematically; instead, one can draw causal inferences.

The counterfactual approach to causal inference has seen a number of research studies [132–136] formalizing the assumptions and the deductive process needed to draw cause-and-effect inferences from statistical data.

In the following, we will use the Directed Acyclic Graph (DAG) to describe the cause-and-effect concept given the observing features and the statistical models. In DAG, each node represents a different feature, and the graph needs to include unobserved variables that influence observable features. Directed edges between nodes illustrate cause-and-effect relationships between variables. By definition, paths following the directed edges in an acyclic graph cannot lead from a node back to itself (i.e., there is no loop), as it is assumed that a variable cannot be a cause at the same time effect of another variable. In the formal statis-

tical theory of DAGs presented by Pearl, Judea [135], the absence of an edge in the graph corresponds to conditional independence of the variables corresponding to the nodes, given all the other variables represented in the graph.

We can decompose the total cause-to-effect in the DAG by analyzing the path-specific components, which is also referred to as the mediation analysis [145]. We demonstrate the above concept using the four variables shown by the DAG in Figure 4.1, although the idea of counterfactual and casual inference extends to more complex structures.

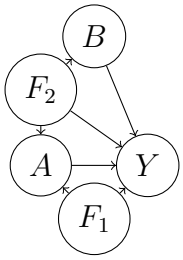


Figure 4.1 The Directed Acyclic Graph (DAG) illustrating the concept of cause-to-effect between the basic four variables case.

Consider the case where we are interested in understanding the impact of the two features F_1 and F_2. In Figure 4.1, we can reach Y from F_1 by following two different paths. A direct path (i.e., no intermediate nodes) and indirectly reaching Y through variable A, also called the 'mediator'. In this case, the conditional expectation $E[Y|F_1 = f_1]$ is the addition of both paths. We can estimate the total effect of the action $F_1 := f_1$ on Y by eliminating the confounding path following the do-operator $E[Y|\text{do}(F_1 := f_1)]$. However, the total effect still lumps the direct and the indirect impact following the two pathways.

Given that the direct effect does not require any counterfactuals, we can easily estimate the direct impact of F_1 feature above by keeping the mediator variable fixed at state $A := a$ and studying the divergence of the distribution in the outcome when $F_1 := f_1$ compared to when an alternative treatment $F_1 := \neg f_1$, using the do-operator. If the conditional distributions $\rho(Y|\text{do}(F_1 = f_1, A = a))$ and $\rho(Y|\text{do}(F_1 = \neg f_1, A = a))$ are known, then the direct impact will be:

$$\Im_{\mathbb{E}}\Big[\ \rho(Y|\text{do}(F_1 = f_1, A = a))\ ,\ \rho(Y|\text{do}(F_1 = \neg f_1, A = a))\ \Big] \tag{4.1}$$

$\Im_{\mathbb{E}}$ in Equation (4.1) denote the expectation of some distance measure quantifying the conditional distributions of the outcome when do-operations is performed on the values of feature F_1, keeping the mediator variable fixed at $A := a$. Similarly we can compute the direct

impact of F_2 on Y following similar process, but in this case, we keep both two mediator variables (i.e., A and B) of F_2 at their respective fixed states, as follows:

$$\Im_{\mathbb{E}}\Big[\,\rho(Y|\mathrm{do}(F_2 = f_2, A = a, B = b))\,,\,\rho(Y|\mathrm{do}(F_2 = \neg f_2, A = a, B = b))\,\Big] \tag{4.2}$$

Throughout this paper, we will refer to the direct impact determined using the Equation (4.1) or Equation (4.2) as "Controlled Direct Impact", because they require setting the other variables in some state.

Definition 4.5.1. *(Controlled Direct Impact [146, 147]) Is the measure of contrast between counterfactual outcomes of the exposure values with alternative values of the exposure (e.g., $F_1 := f_1$ and $F_1 := \neg f_1$, where $f_1 \neq \neg f_1$), when the mediator (s) were kept to their respective fixed value. In this study, we will estimate the Controlled Direct Impact using Equation (4.1) or Equation (4.2).*

However, the major drawback of directly computing the impact of the features following the above formulation is that it generally does not produce valid counterfactual. Considering that A and B are relevant variables influencing the outcome, computing counterfactual with respect to F_1 or F_2 would require adjusting even the downstream variables A and B. We can estimate the impact of the Features F_1 and F_2 on the outcome Y, by adjusting variables A and B, as follows:

$$\Im_{\mathbb{E}}\Big(\rho(Y|\mathrm{do}(F_1 = f_1, A =_{(a,\neg f_1)})), \rho(Y|\mathrm{do}(F_1 = \neg f_1, A =_{(a,\neg f_1)}))\Big) \tag{4.3}$$

and:

$$\Im_{\mathbb{E}}\Big(\rho(Y|\mathrm{do}(F_1 = \neg f_1, A =_{(a,f_1)})), \rho(Y|\mathrm{do}(F_1 = \neg f_1, A =_{(a,\neg f_1)}))\Big) \tag{4.4}$$

where the $\rho(Y|\mathrm{do}(F_1 = f_1, A =_{(a,\neg f_1)}))$ in Equation (4.3) denote the probability distribution that the outcome Y would obtain had F_1 been set to f_1 and had A been set to the value A would've assumed had F_1 been set to the alternative value: $F_1 := \neg f_1$. On the other hand, for Equation (4.4), the property $\rho(Y|\mathrm{do}(F_1 = \neg f_1, A =_{(a,f_1)}))$ is the probability distribution of the outcome Y when the F_1 is kept at the alternative $F_1 := \neg f_1$ while changing the value of the mediator variable A to the value it would have attained had $F_1 := f_1$ was used. The above formulations of impact analysis is referred to as 'Natural Impact', where by Equation (4.4) and Equation (4.3) can be used to estimate the natural direct and indirect impact of feature F_1 on the outcome Y.

Definition 4.5.2. *(Natural Impact [146, 147]) is the causal effect of the exposure defining the hypothetical contrast between the outcomes that would be observed simultaneously in the*

same individual in the presence and the absence of the exposure. When we add up the two Natural Impacts, i.e., the natural direct effects in Equation (4.3) and natural indirect effects in Equation (4.4), the resulting value is the total effect [148, 149].

So far, we have discussed how the cause-to-effect can be estimated using the counterfactual approach to causal inference involving the exposure to the mediator variables and how it compares to the traditional method. It is worth noting that the technical possibilities of the counterfactuals go beyond the scope of this study discussed above. Intuitively, through counterfactuals we can compute all sorts of effects specific to a pathway. The challenge with this approach is that we may be unsure if there is exposure-mediator interaction and the direction of the interaction. In general, what constitutes relevant features is very crucial when selecting and analyzing casual to effect. As Pearl and Mackenzie noted in their book [150], the biggest mistake one can make is mistaking a mediator for a confounder in causal inference and can result in most outrageous error as the latter invites adjustment; the former forbids it. We can exploit the criteria defined in the theories of probabilistic causation [137], to address the challenges of causal inference. Notably, Suppes [141] explanations provides a better understanding of probabilistic causation by introducing the notion called 'prima facie cause'. The definition is based on two principles: (i) *temporal priority* stating that any effect must happen after the cause (temporal priority), and (ii) *probability raising*, which states that the cause should raise the probability of observing the effect.

Definition 4.5.3. *(Probabilistic causation [137]) Whenever two events involving the cause e_1 and effect e_2, occurring at times t_{e1} and t_{e2}, respectively, under the mild knock down assumptions that their probability ranges: $0 < \rho(e_1), \rho(e_2) < 1$, the event e_1 is a prima facie cause of the event e_2 if it occurs before the effect and the cause raises the probability of the effect, such that:*

$$t_{e1} < t_{e2} \text{ and } \rho(e_1|e_2) > \rho(e_1|\neg e_2) \tag{4.5}$$

In the Equation (4.5), the *temporal priority* corresponds to the first condition $t_{e1} < t_{e2}$, where as the second condition: $\rho(e_1|e_2) > \rho(e_1|\neg e_2)$ is referred to as *probability raising* [141]. As shown in Equation (4.5), to sufficiently claim that event e_1 is a cause of event e_2 while also satisfying the prima facie cause, the above conditions will require us to know the time information of each event. However, because the information about the time is not always available and it is almost impossible to determine this automatically, but the user may have the information about part of the feature. We build on the idea that users may have partial or full knowledge about the timing order of the features in the data. For instance, the

features like race and gender may be considered antecedents to most of the features like job, and income, given the time of data collection. Therefore the user may specify that the feature race should be considered in first in the temporal ordering compared to the rest of the features whose timing may be unknown. To this end, we introduce a notion of full or partial temporal ordering that combines the full or partial temporal ordering manually specified by the user together with the idea of positive statistical dependence, introduced in [151]. The positive statistical ordering translates the timing of the two events as the rate of occurrence (frequency). i.e., for e1 to occur before e2, then:

$$p(e_1) > p(e_2) \iff \frac{\rho(e_2|e_1)}{\rho(e_2|\neg e_1)} > \frac{\rho(e_1|e_2)}{\rho(e_1|\neg e_2)} \tag{4.6}$$

I.e., the positive statistical dependence states that for the Eqn (4.5) to hold between the two events e_1 and e_2 for which e_1 raises the probability of e_2 more than e_2 raises the probability of e_1 (i.e., $\rho(e_1|e_2) > \rho(e_1|\neg e_2)$), if and only if e_1 is observed more frequently than e_2.

Therefore we require that the ordering specified by the user must satisfy the positive statistical dependence, but in case the ordering is not manually specified as input by the user (or only part of the ordering is used), then only the positive statistical is taken into account. The idea of using manual input for the temporal ordering caters to the case where we have the domain knowledge to capture the timing semantic between the features that may not be reflected as the frequency of occurrence. On one hand, we believe that integrating the domain knowledge is essential, while on the other hand, we do not require the complete knowledge about the time information, for our framework to work. The idea of using partial knowledge about the domain was used before in [152].

Divergence measurement Metrics

As evidenced by Eqns 4.1-4.4, the causal effects require a notion of divergence (distance) $\Im(\rho_1, \rho_2) \geq 0$ between two probability distributions ρ_1 and ρ_2 over the output space $\mathcal{Y}$. We now review some possible choices of divergence metrics that will be considered in this study.

- **Hellinger Distance:** is defined as

$$\Im(\rho_1, \rho_2) := D_{HL}(\rho_1||\rho_2) = \frac{1}{\sqrt{2}}\sqrt{\sum_{y\in\mathcal{Y}}\left(\sqrt{\rho_1(y)} - \sqrt{\rho_2(y)}\right)^2}$$

  It is noted that, the value of Hellinger distance is affected to the greater by the same absolute difference of $\rho_1(y)$ and $\rho_2(y)$ when the value of $f(x) \to \hat{y}$ is small. The Hellinger

distance is a measure of sine of the angle between the Hilbert vectors representing the two square-roots $\sqrt{\rho_1(y)}$ and $\sqrt{\rho_2(y)}$ in which each square-roots density is a point on the unit sphere in the Hilbert space.

- **Total variation distance:** is defined as

$$\Im(\rho_1, \rho_2) := D_{TV}(\rho_1||\rho_2) = \frac{1}{2}\sum_{y\in\mathcal{Y}} |\rho_1(y) - \rho_2(y)|$$

The total variation distance is closely related to the Hellinger distance as both distances represent the same topology of the space of probability measures. However, the use of Hilbert spaces (inner products) properties in computing the Hellinger distance gives the Hellinger distance some technical advantages compared to the total variation distance.

- **Wasserstein distance:** assuming $\mathcal{Y}$ is one-dimensional, the distance is defined as

$$\Im(\rho_1, \rho_2) := D_W(\rho_1||\rho_2) = \int_{\mathcal{Y}} |F_1(y) - F_2(y)|dy,$$

where F_1 and F_2 are the Cumulative Distribution Function (CDF) of the distributions ρ_1 and ρ_2 respectively. The Wasserstein distance is also known as the earth mover's distance, because it can be interpreted as the minimum amount of "work" required to transform one distribution into the other.

- **Kullback-Leibler (KL) divergence:** is defined as

$$\Im(\rho_1, \rho_2) := D_{KL}(\rho_1||\rho_2) = \sum_{y\in\mathcal{Y}} \rho_1(y) \log\left(\frac{\rho_1(y)}{\rho_2(y)}\right).$$

We note that this divergence measure is not necessarily symmetric *i.e.* $D_{KL}(\rho_1||\rho_2) \neq D_{KL}(\rho_2||\rho_1)$ in general. For this study, we will instead use the modified version called Jensen-Shannon (JS) divergence.

- **Jensen-Shannon (JS) divergence:** is an extension of the KL divergence which aims at being symmetric and having finite values

$$\Im(\rho_1, \rho_2) := D_{JS}(\rho_1, \rho_2) = \frac{1}{2}D_{KL}(\rho_1||\rho_m) + \frac{1}{2}D_{KL}(\rho_2||\rho_m) \tag{4.7}$$

Where $\rho_m = \frac{\rho_1+\rho_2}{2}$ is called the mixture distribution.

When the two distributions are identical in the Eqn (4.7), the divergence measure will equate to zero, indicating that the feature is not a potential bias feature. On the other hand, the

feature is a potential bias feature if the divergence is greater than zero. Next, we will define the bias machine learning model.

4.6 Proposed Approach

This section now details our approach for systematically identifying all bias-inducing features based on novel single feature swapping and double feature swapping functions to help estimate the direct and indirect impact of each feature on the model prediction.

4.6.1 Data Swapping

Having defined measures of divergence $\Im(\cdot,\cdot)$, we now want to estimate the hypothetical quantities such as $\rho(Y|\text{do}(F_1 = f_1, A = a)), \rho(Y|\text{do}(F_1 = \neg f_1, A = a))$ from the actual data and the machine learning model prediction. This estimation will allow us to quantify the impact of each feature on the model prediction, i.e., Controlled Direct Impact following Definition (4.5.1) and Natural impact following Definition (4.5.2) above, using the data swapping methods. Willenborg and De Waal [153] provide the formal definition of data swapping for $2k$ items in terms of k item swap. They defined data swapping of two data points i and j selected from the same dataset X and interchanging the values of the variable being swapped for these two data points. Visually, we can say that Data swapping involves "switching the values of columns for two pairs of rows". To be more precise, we want to condition the random input $\boldsymbol{X}$ and the model outcome Y before and after swapping the features in $\boldsymbol{X}$. In this case, we are interested in how much the distributions of Y change when we force X_{ij} to take a specific value. This method of forcing the random variable X_{ij} to take a certain value is what we call intervention. With this in mind, we will soon define a"Swapping function" to evaluate and quantify the bias in the machine learning model. But first of all, we must introduce some key concepts such as the swap ratio and the maximum distortion.

Definition 4.6.1. *(Swap Ratio) The swap ratio denoted as r, is the percentage number of the records randomly selected from $\boldsymbol{X}$ whose features will be swapped.*

The hyper parameter r is manually defined by the user such as 0.1, or 0.2 indicating that 10% or 20%, respectively, of the random sampled data points are selected for swapping. By default, 0.5 is used as the standard swap ratio indicating 50% of the data points are selected for swapping

Definition 4.6.2. *(Maximum Distortion) Denoted as d_{max} is the maximum allowed statistical distance to quantify how much the data point should change from the original. The idea is*

that the hypothetical input data should be considered valid only if it is within the accepted range compared to the actual input.

To compare the distortion with the maximum distortion, we must first quantify the distance between the original and the swapped data points. This quality described in terms of a distortion measure will allow us to know how much the changed data points deviate from the original. Because the input data is likely to contain both the categorical and continuous numerical data, we defined a distance measure for mixed variable data by combining the square Euclidean distance for numeric variables and a simple matching distance for categorical variables, as:

$$d(u, v) = d_C(u, v) + d_N(u, v) \tag{4.8}$$

In Eqn (4.8), $d_N(u, v)$ is the distortion between numeric variables, while the property $d_C(u, v)$ is the distortion measure between categorical variables. We however noted that the choice for the distance metric above and the maximum allowed distortion $d_{\max}$ should be guided by the social implication, with the help of domain experts and stockholder as well the legal consultation such as the 80% rule. In our case, we used the Hamming distance to compute the distance for categorical features and for numeric data, we defined a distance measure in the form of the ratio of the individual distance measures before u and after swapping v, as follows:

$$d_C(u, v) = \begin{cases} 1, & \text{if } u \neq v \\ 0 & \text{if } u = v \end{cases}$$

$$d_N(u, v) = \|u - v\|$$

It is clear from the Eqn (4.8) that the distortion value equal $d_x(u, v) = 1$ if only a single binary variable is swapped, and $d_x(u, v) = 2$ if the two binary variables are swapped. For this case, the swapped data points will not be subjected to the maximum distortion $d_{\max}$. The maximum allowed distortion is only used to check for the case where the variable(s) swapped is a continuous variable or a mixture of both continuous and discrete categorical variables.

Definition 4.6.3. *(Feature Values Partitioning) This is a function that splits the values of the given feature, following some binning technique so that the values will belong to only one of the two groups.*

Initially all features are assumed to follow a binary categorical format. If there exist contin-

uous or discrete features in X, a binning technique is applied to the feature values to ensure the values belong to one of the categories. Specifically, if we denote the different values of the features that we want to partition by u. We partition u by the medium point into two categories. Formerly, if we let $C_M = \{C_1, C_2\}$ to be a $M = 2$ partition of u, then:

$$C_m = \begin{cases} [u_{(1)}, u_{(1)+\delta} & C_1, \quad \text{if } m = 1 \\ [u_{(1)(M-1)\delta}, u_{(n)}] & C_2, \quad \text{if } m = M = 2 \end{cases} \tag{4.9}$$

Where δ is the bin width given by: $\delta = \frac{u_{(n)} - u_{(1)}}{M}$. In Eqn (4.9) $\{u_{(i)}\}_{i=1}^{n}$ denote the ordering of the statistics where the smallest value in the set is $u_{(1)}$ and $u_{(n)}$ is the largest value of the feature, and therefore the k^{th} smallest value in the set is $u_{(k)}$.

We must point out that, even if the bin is of equal width, the number of samples in each category is likely to be non-equal. Also, it is worth pointing out that the actual values of the x were not replaced; instead, this step was only used to help when choosing a random value from a different category during the swapping process. Hence the probability distribution of the data is reserved before the data swapping process. For example, for the dataset containing age feature values between 17 to 70 years, the first category may contain all the ages from 17 to 30, while C_2 are group of age values from 31, i.e., $C_m = \{\leq 30, > 30\}$. A data point with an age value, say 20, can be swapped with any value of age randomly chosen from the category > 30. Further details about the swapping process are provided with Definition (4.6.4).

Definition 4.6.4. *(Swapping function) is a function that switches between the values of the feature under investigation so that the values of that feature are swapped each time, under some set of constraints.*

The swapping function denoted as φ takes as input the dataset $\boldsymbol{X}$ (i.e., test dataset, for our case), the index of the feature j to swap, the set of indices of instances $I \subset \{1, 2, \ldots, n\}$ to swap, and the maximum allowed distorsion $d_{\max}$. This function returns an alternative dataset $\boldsymbol{X'}$ i.e.

$$\boldsymbol{X'} = \varphi(\boldsymbol{X}, j, I, d_{\max}), \tag{4.10}$$

such that

$$X'_{ik} = \begin{cases} \neg X'_{ij} & \text{if } i \in I' \text{ and } k = j \\ X_{ik} & \text{Otherwise} \end{cases}, \tag{4.11}$$

where $I' \subseteq I$ is a subset that is chosen in order to ensure that:

$$d(\boldsymbol{X}, \boldsymbol{X'}) \leq d_{\max}. \tag{4.12}$$

Simply put, the given feature under investigation X_{ij} is changed with the alternative value: $\neg X_{ij}$. The alternative value is determined by first identifying the category (i.e., C_1 or C_2) X_{ij} belongs following the data partitioning technique, see Definition (4.6.3, and Eqn (4.9)). If X_{ij} belongs to C_1 then the new value $\neg X_{ij}$ is randomly chosen from C_2, and vice versa. For the gender feature, for instance, this implies that male values become female and vice versa. In some situations, the pairs of data points for swapping need to meet some condition to be considered swapping candidate. For example, we may allow the switching of the value of the feature only if the distortion of distribution between the original and the post-swapped data point (as measured using some distance function) is not more than the maximum allowed threshold $d_{\max}$; see Definition (4.6.2). For instance, if switching a capital-gain feature of 2,000 with a value of 0 can result in a larger difference, as measured using Eqn (4.8), then the swap is not allowed.

• **Single Feature Swapping:** The single swapping function denoted as $\varphi(\boldsymbol{X}, j, I, d_{\max})$ aims at identifying the Controlled Direct Impact of the feature on the machine learning model following Definition (4.5.1).

We present the summary of Single feature swapping function in Algorithm (1). The function takes as input the dataset $\boldsymbol{X}$, the feature column to be swapped j, the random swap indices I, and the maximum distortion $d_{\max}$. For every candidate data point chosen for swapping (Line 4-5), the distortion is computed to ensure it is within the allowed range in Line 6. Without the loss of generality, X_{ij} and the entire dataset $\boldsymbol{X}$ is assumed to follow the binary categorical format. The initial step begins by determining the categories of the feature's values X_{ij} with Eqn (4.9). The output of this function will be used to estimate the Controlled Direct Impact

Algorithm 1 Single Feature Swapping function

INPUT: $\boldsymbol{X}, j, I, d_{\max}$
OUTPUT: $\boldsymbol{X}'$

1: $n, m = \boldsymbol{X}$.`shape`
2: $\boldsymbol{X}' =$ `zeros`(n, m)
3: **for** $i \leftarrow$ `range`$(0, n)$ **do**
4: $\quad \boldsymbol{X}'[i, :] = \boldsymbol{X}[i, :]$
5: $\quad$ **if** $i \in I$ **then**
6: $\quad\quad$ **if** $d(X_{ij}, \neg X_{ij}) \leq d_{\max}$ **then**
$\quad\quad\quad \boldsymbol{X}'[i, j] = \neg X_{ij}$
7: $\quad\quad$ **end if**
8: $\quad$ **end if**
9: **end for**
10: **Return** $\boldsymbol{X}'$

by computing the statistical distance of the distribution in the prediction of $\boldsymbol{X}'$ and $\boldsymbol{X}$.

• **Double Features Swapping:** The double features swapping function aims to quantify the Total Natural Impact of each feature in $\boldsymbol{X}$ on the machine learning model following Definition (4.5.2). The function takes as arguments (input) the dataset $\boldsymbol{X}$ (i.e., test dataset, for our case), the swap ratio r, temporal priority ordering $t_o : F \rightarrow N$(i.e., manually or automatically determined; please refer to discussion for Definition (4.5.3)), and the maximum allowed distortion $d_{\max}$ and returns the alternative dataset, which is the dataset with alternative values of a feature at column j and all values of the mediating values at column m switched, keeping the rest of variables unchanged and the swapped rows are subjected to the maximum allowed distortion and the swap ratio r. We must note that, before effecting the double swapping, we first check to ensure that the pairs consisting of the feature and a given mediating variable must satisfy the probabilistic causation; refer to Definition (4.5.3). To this end, we allow the user to provide the partial or full temporal ordering of the features as input and we constrain the rest of the feature using the Eqn (4.6). For the candidate pairs satisfying the probability causality defined by the temporal priority ordering $t_o : X_j \rightarrow X_m$ (indicating that X_j occurred before X_m) and Eqn (4.6), the following two scenarios are considered when swapping the values of the features and the mediating variable:

- The first scenario includes investigating the case $\boldsymbol{do}(X_j = x_j, X_m = (x_m, \neg x_j))$ vs $\boldsymbol{do}(X_j = \neg x_j, X_m = (x_m, \neg x_j))$ is about changing only the mediating variable vs changing the values for both the mediating variable and the feature to the alternative values as; $X_j := \neg x_j, X_m := \neg x_m$, for some selected data points I. When we use Algorithm (1), this implies calling the Algorithm twice, first with the mediator index m and the selected indices I, and get the set of swapped datapoints $\boldsymbol{X}'$. Next, we will use the swapping Algorithm (1) to swap the feature values at index j for the input $\boldsymbol{X}'$. Hence we refer to this swapping function as double features swapping function. The output of the double feature swapping function will be the tuple $(\boldsymbol{X}', \boldsymbol{X}'')$. Formally, we define the double features swapping of data $\boldsymbol{X}$ described above as:

$$\begin{aligned} \boldsymbol{X}' &= \varphi(\boldsymbol{X}, m, I, d_{\max}) \\ \boldsymbol{X}'' &= \varphi(\boldsymbol{X}', j, I, d_{\max}) \end{aligned} \tag{4.13}$$

  The difference between the distributions of model prediction of $\boldsymbol{X}'$ and $\boldsymbol{X}''$, in Eqn (4.13) will be used to determine the hypothetical Natural Direct Impact of the feature X_j.

- The second scenario includes studying the case $\boldsymbol{do}(X_j = \neg x_j, X_m = (x_m, x_j))$ vs.

$\boldsymbol{do}(X_j = \neg x_j, X_m = (x_m, \neg x_j))$. In this case the property $(X_j = x_j, X_m = (x_m, \neg x_j))$ includes changing only the value of feature without changing the mediating values for some selected data points I, while the property $\boldsymbol{do}(X_j = \neg x_j, X_m = (x_m, \neg x_j))$ corresponds to changing both the feature values and the mediating values in I to the alternative values. Formally, we can define the second scenario of double features swapping of data $\boldsymbol{X}$ above as:

$$\begin{aligned} \boldsymbol{X}' &= \varphi(\boldsymbol{X}, j, I, d_{\max}) \\ \boldsymbol{X}'' &= \varphi(\boldsymbol{X}', m, I, d_{\max}) \end{aligned} \tag{4.14}$$

The outcome of this function will be used as input to the model when estimating the Natural Indirect Impact of the above treatment on the model prediction

4.6.2 Evaluate the Potential Bias in Machine learning

As mentioned earlier, our study aims to identify the potentially bias-inducing features and evaluate the level of bias in a machine learning model following some evaluation metrics. We wish to identify all the columns j of $\boldsymbol{X}$ that may introduce bias to the machine learning model using some evaluation method.

Definition 4.6.5. *(Potential bias-inducing Feature (PBF) [20, 154]) Given a swap function φ, the feature $j \in \{1, 2, \ldots d\}$ is said to be a potentially bias-inducing feature when the random variables $\hat{Y} = f(X)$ and $\hat{Y}' = f(\,\varphi(X, j)\,)$ have a large divergence.*

In the Definition (4.6.5), $\varphi(X, j)$ denotes the swapping of input feature j in X under some set of constraints using the swapping function $\varphi(\cdot\cdot)$; refer to Definition (4.6.5) and Section 4.6.1, for the definition of the swapping function, and the different swapping functions proposed in this study. $\hat{Y}'$ is the model prediction on the swapped input $f(\,\varphi(X, j)\,)$.

Using the loan application as an example, X denotes the dataset of the n loan applicants, and $\hat{y}$ is the model's prediction of whether the applicant should be approved for a loan or denied a load. x_j can represent the gender feature, and $\hat{y}'$ represents the model prediction after swapping the gender value from male to female of the n application.

Next, we want to find the divergence in the original model outcome distribution and the model prediction distribution due to data swapping inputs. The idea here is that we want to quantify the potential impact of each feature as being bias-inducing if it satisfy the property $\Im(\,\rho(Y), \rho(\hat{Y}')\,) = \delta$, for some some divergence function $\Im(\cdot, \cdot)$, such as Jensen-Shannon divergence [155]. In this case, the larger value of δ estimates the larger impact of that

feature on the model prediction and is ranked as more bias-inducing. In this study, we aim to estimate the impact of the potential biased feature on the model's prediction using both the single feature swapping function and the double features swapping functions described above. In this study, we consider four different divergence measures (i.e., Hellinger distance, Total variation distance, Jensen-Shannon divergence, and Wasserstein distance, detailed in Subsection 4.5 above) to compute the statistical distance between the original data and the post-swapped data. Note that, for the double feature swapping, we will compare the statistical distance between the model prediction of the input data $\boldsymbol{X}'$ and $\boldsymbol{X}''$ for both scenarios (i.e., Eqn 4.13, and Eqn 4.14). On the other hand, for the single feature swapping function, we will be computing the statistical distance between predictions of $\boldsymbol{X}$ and $\boldsymbol{X}'$.

Definition 4.6.6. *(Bias Machine Learning Model) A machine learning is considered biased if it depends on one or more potential bias feature (PBF), either directly or indirectly through some mediating variable (s), given the actual model outcome [24, 156]*

We want to quantify the PBF that directly or indirectly impacts the outcome of the machine learning model and evaluate the level of bias induced in the model. To determine if a given feature is a PBF, we compute the statistical difference of the distributions of the model prediction before $\hat{Y}$ and after changing the value of the PBF $\hat{Y}'$, following the set of steps as described above. The larger divergence indicates the greater difference between the distributions of the model outcomes, which in turn suggests the high impact of the feature on the model prediction and likewise the presence of bias.

4.7 Experimental Evaluation

For empirical evaluation of bias in the features given the dataset and machine learning model. We first applied a cross-validation (CV) sampling technique to split the dataset into the training D_{train} and the test set D_{test} into k-folds, where each fold contains training and testing samples. This well-known technique will allow our method to generalize to different variability in the input data reflecting real-world situations. In this study, we used $k = 10$ folds where the training fold consists of 90% of the data, while the remaining 10% is used for testing. Specifically, the following are the steps we followed:

- Determine and remove all correlated features from the dataset.
- Split the dataset into training and test dataset using a 10-fold cross-validation sampling technique, where 90% is taken as the train set, and 10% is used for testing the model.
- Train the machine learning model using the dataset D_{train}.

- Use the trained model to predict all the data points in the testing dataset D_{test} as $\hat{Y}$.
- Apply data swapping on the test set using both single features swapping and double features swapping functions.
- Make the model prediction using the single feature swapped test dataset as $\hat{Y}'_S$, and similarly predict using the double features swapped dataset as $\hat{Y}'_D$.
- Using some distance functions, evaluate the divergence of the distributions $\hat{Y}$ and $\hat{Y}'_S$, and independently evaluate the divergence between the distributions of $\hat{Y}$ and $\hat{Y}'_S$, for each features. For the double swapping function, the direct natural impact and the natural indirect impact are computed separately from the input derived from the two scenarios presented in Equation (4.3)and Equation (4.4), respectively, then the total is reported.
- The features that return the higher divergences value indicate the larger the bias of that feature on the machine learning model.

Baseline Model

The proposed framework is flexible, with both classification and regression models as the backbone, including neural networks, logistic regression, and probabilistic classifiers. The work can be extended to other traditional classifiers such as Support Vector Machine (SVM) by simply changing the evaluation metrics e.g., using the propensity score instead of the distance measure.

SHAP Value

SHAP (SHapley Additive exPlanations) [157] is a state-of-the-art explanability method based on the famous Shapley values from game-theory. Such a technique provides a function $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^m$ that, for any input $x \in \mathcal{X}$, provides a m-vector $\boldsymbol{\phi}(x)$ whose components will sum up to

$$\sum_{j=1}^{m} \phi_j(x) = f(x) - \mathbb{E}[f(X)]. \tag{4.15}$$

That is the difference between the prediction at a specific x and the average prediction is shared among the different features. The vector $\boldsymbol{\phi}(x)$ is called the feature attribution and each score $\phi_j(x)$ is meant to convey how much feature j has contributed to the model output at x. Still, these score are local and must be aggregated in order to provide a global sensitivity

measure $\mathbf{\Phi} \in \mathbb{R}^m$. To do so, we average the magnitude of the feature attributions

$$\Phi_i = \mathbb{E}[\,|\phi_i(X)|\,], \tag{4.16}$$

which is the default approach in the SHAP Python library [157].

Dataset

- **Student:** This dataset contains the student performance of the two secondary schools in Portugal. The features include student demographic information, social and school related features. The target variable is final year grade. Specifically the datasets are about the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por). The feature G3 (final year grade) is known to have strong correlation with features G2 (second period grades) and G1 (first period grades).
- **Cleveland Heart Health [158]:** This dataset contains information about Patients such as the age, sex, ca (number of major vessels), thalach (maximum heart rate), among others from the Cleveland database. The target variable refers to the presence of heart disease in the patient ranging from 0, indicating no presence of the heart disease, to 4 (present).
- **COMPAS Recidivism:** This dataset contains over $10,000$ criminal defendants' information used by the COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) algorithm for scoring defendants in Broward County, Florida, for a period of two years. COMPAS is a popular recidivism algorithm used by judges, probation, and parole officers across Florida US State for scoring a criminal defendant's likelihood of recidivism (reoffending). This database is known to be biased in the sense of having different False Positive Rates between white and black sub-populations [159].
- **Bank dataset:** This dataset is about marketing campaigns based on phone calls of a Portuguese bank. The goal is to predict whether the client subscribe (yes/no) to the term deposit, used in the Decision Support Systems (DSS) [4].

4.8 Experimental Results

This Section details the experimental results evaluating our proposed approach on four different datasets to answer the two research questions.

4.8.1 RQ1: Can we identify which features potentially introduce bias to the model?

This section reports the results of the experiments using the proposed swapping functions to detect the features that potentially introduce bias to machine learning, answering our **RQ1**. As described earlier, our swapping functions consist of single feature changing to estimate the direct impact of each feature on the model prediction under controlled conditions (i.e., controlled direct impact), and the double features swapping functions to estimate the features that naturally impact the model prediction through mediating variables (i.e., total natural effects).

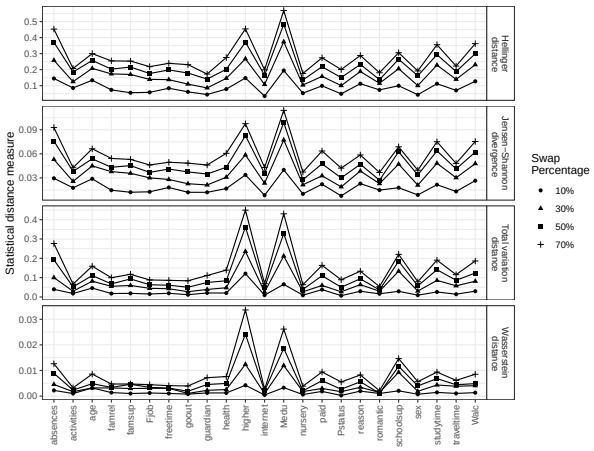
We studied the impact of swapping each feature using the swapping functions (i.e., single feature swapping functions and double features swapping function) for swapped percentages ($10\%, 30\%, 50\%,$ and 70%) of the test dataset, and constrained the swapped input not distorted more than 20% (i.e., $d_{\max} \leq 0.2$), motivated by '80% rule'. The results presented bellow are categories based on the dataset used for the experiments, i.e., in the order: *Student dataset*, *Cleveran Heart dataset*, *COMPAS Recidism dataset*, and *Bank dataset*:

Student Dataset

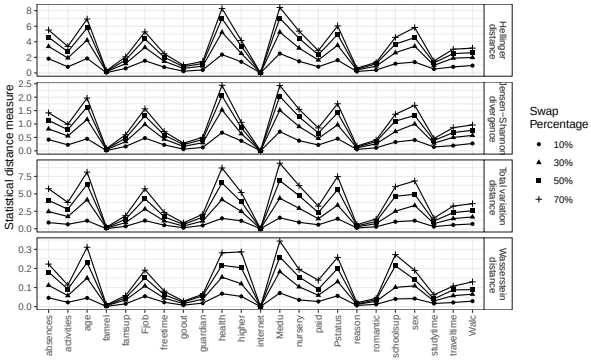
Figure 4.2 shows the experimental results of our swapping functions comparing the direct impact of each feature on the model prediction and the total effect of each feature on the model prediction for the Student performance dataset.

First, In Figure 4.2a we report the experimental results of the single feature swapping function showing how each feature directly impacts the model prediction when other variables/ features are kept unchanged (controlled direct impact), evaluated using the four different divergence measures: Hellinger, Jensen-Shanon divergence, Total variation distance, and Wasserstein distance.

As seen in Figure 4.2, for all four divergence measures, we can observe that the values increase with the increase in the swap percentage. Overall we can see that there is a consistency in the order of feature ranking when values of a single feature are swapped, keeping other features unchanged. As detailed earlier in Section 4.7, the student dataset contains information about student achievement in secondary education in Portuguese schools where the goal is to predict the student grades. on average, in Figures 4.2a, the minimum value of the statistical distance are observed when the feature 'nursary', 'Pstatus', 'romantic' and 'sex' is swapped and the statistical distance is maximum when features 'higher', 'Medu', or 'absences' is swapped. The similar trends are observed for the SHAP values. The feature 'higher' is the binary



(a) Results of single feature swapping function, showing the direct impact of each feature on the model prediction



(b) Results of the double features swapping function demonstrating the total effect of each feature on the model prediction.

Figure 4.2 Distance measure showing the experimental results of the single feature swapping function in Figure 4.2a, and double features swapping functions in Figure 4.2b for the Student's Performance dataset, using the swap percentage of $(10\%, 30\%, 50\%$, and $70\%)$

variable indicating whether a student wants to take higher education (1 or yes) or not (0 or no), 'Medu' is an acronym for mother's education, while 'absences' is the numeric feature indicating the number of school absences ranges from 0 to 93. These results indicate that the features 'higher', 'Medu', or 'absences' plays significant role to the model prediction hence highly impact the model compared to the most frequent minimizers features such as: attended nursery school ('nursary'), parent's cohabitation status ('Pstatus'), 'romantic' (in a romantic relationship) and 'sex'.

In Figure 4.2b when we set the temporal priority ordering: $sex \rightarrow age \rightarrow health \rightarrow Pstatus \rightarrow nursery \rightarrow Medu \rightarrow Fjob \rightarrow schoolsup \rightarrow absences \rightarrow activities \rightarrow higher \rightarrow traveltime \rightarrow paid \rightarrow guardian \rightarrow Walc \rightarrow freetime \rightarrow famsup \rightarrow romantic \rightarrow studytime \rightarrow goout \rightarrow reason \rightarrow famrel \rightarrow internet$, and perform a double features swapping to investigate the impact of each feature through the mediating variables. The results reported in Figure 4.2b is the total sum of the pairwise swapping of each feature and its respective mediating variables. For instance for the feature sex, will be $(sex, age) + (sex, health) +, .., +(sex, internet)$, refer to Table 4.1. According the result in Figure 4.2b, we can observe that the feature 'Medu' still indicates the maximum statistical distance, and 'age' is the close second maximal statistical distance. The student sex feature, which originally indicated a minimal impact when a single feature swapping function, now shows

a higher impact when the sex and all its mediating variables (from feature 'age' until 'internet') were swapped using the double features swapping function. The ordering in the statistical distance for 'sex' and 'schoolsup' are slightly different for the different swap percentages when *Wasserstein distance* or *Total variation distance* is used, but consistent when *Jensen-Shannon divergence* or *Hellinger distance* is used as distance measure. The similar difference in ordering is observed between the features 'higher' and 'health'.

Table 4.1 detailed breakdown of the results for double features swapping function, estimating the impact of each feature on predicting the student performance, through the mediating variables (i.e., column header). The results shown in Table 4.1 are only for the 50% swap percentage. Each features (i.e., the second column named 'Features') where checked against each mediating variable (column header starting from 'age' to 'studytime'), to understand how the total impact on the model prediction, using our proposed double features swapping functions. According to Table 4.1, the variables which dominantly shows the maximal causes of mediation in the model prediction and the pairwise features are: 'Medu', 'Absences', 'higher' and 'study time'.

> ◉ **Answer to RQ1 (Student Dataset).** *Concluding from our results of swapping values of each feature (keeping other features unchanged) of the student data, we can say that the features 'higher', 'Medu', or 'absences' plays a significant role in predicting the student's performance. Furthermore, similar features (i.e., 'higher', 'Medu', 'absences', and 'study time' inclusive) are observed to indicate the maximal causes of mediation in the model prediction and other features. Moreover, the features which are highly impacted by the mediating variables and the model prediction of student performance are frequently the features such as Age, sex, health, Medu, and Pstatus.*

Clevelan Heart Dataset

This Subsection detailed the empirical evaluation of our proposed swapping functions on the Clevelan Heart dataset. For the cleveran Heart dataset, the features for training and testing the machine learning model include: 'sex', 'age', 'thalach', 'ca', 'thal', 'exang', 'cp', 'trestbps', 'restecg', 'fbs', 'oldpeak', 'chol'. The descriptions of each features can be found in [158]. Similar to the Student dataset, the features that were found correlated were removed during the preprocessing step. The target variable is a binary class referring to the presence or absence of heart disease in the patient.

Figure 4.3 shows the experimental results on the Clevelan Heart dataset of our swapping functions comparing the direct impact of each feature on the model prediction and the total

Table 4.1 Summary of natural impact of pairwise feature (column 'Feature') and mediator (column from 'age' to 'studytime') on the model prediction for **Student performance dataset**, with temporal priority ordering: $sex \rightarrow age \rightarrow health \rightarrow Pstatus \rightarrow nursery \rightarrow Medu \rightarrow Fjob \rightarrow schoolsup \rightarrow absences \rightarrow activities \rightarrow higher \rightarrow traveltime \rightarrow paid \rightarrow guardian \rightarrow Walc \rightarrow freetime \rightarrow famsup \rightarrow romantic \rightarrow studytime \rightarrow goout \rightarrow reason \rightarrow famrel \rightarrow internet$. The maximum values along the rows are highlighted in ***bold face + italic***, and second highest are in **bold face**.

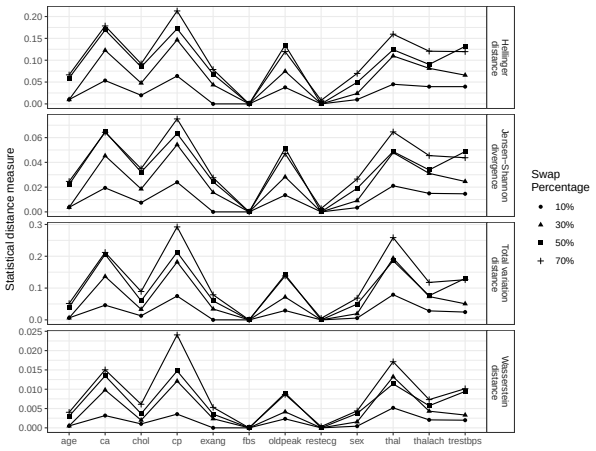
.

Meası	Featur	age	health	Medu	school sup	absences	activities	higher	Walc	free time	study time
Hellinger distance	sex	0.22	0.28	***0.34***	0.17	0.3	0.1	**0.31**	0.28	0.19	0.23
	age	-	0.35	***0.39***	0.28	0.37	0.24	**0.38**	0.31	0.25	0.3
	health	-	-	***0.48***	0.33	0.41	0.26	0.34	**0.46**	0.35	0.41
	Medu	-	-	-	0.38	***0.51***	0.34	0.48	0.47	0.41	**0.49**
	schoolsup	-	-	-	-	**0.34**	0.17	0.3	0.28	0.24	***0.36***
	absences	-	-	-	-	-	0.23	0.37	***0.41***	0.33	**0.39**
	higher	-	-	-	-	-	-	-	**0.28**	0.27	***0.32***

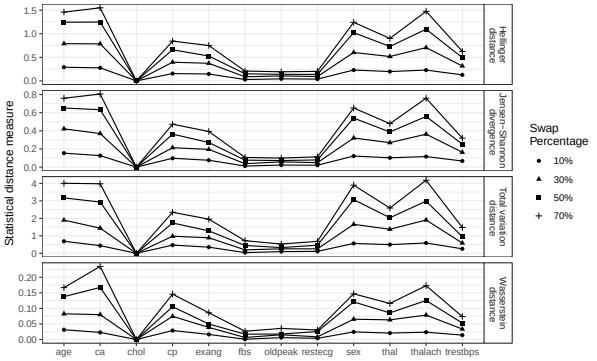
natural effect of each feature on the model prediction.

In Figure 4.3a, we show the results of the single feature swapping function on Clevelan Heart dataset to demonstrate the controlled direct impact of each of the features on predicting the presence or absence of heart diseases in the patient, using the swap percentage $(10\%, 30\%, 50\%, 70\%)$. Similar to the Student performance dataset reported earlier, as shown in Figure 4.3a, the statistical distance between the model prediction before and after swapping each features increases with the swap proportion. These remains consistent for all the four divergence measures. The ordering of the distance measures in Figure 4.3a are close similar, whereby the feature 'cp', 'ca', and 'thal' shows the maximal values of statistical distances, while the feature 'fbs' and 'restecg' demonstrate the minimal values. These results indicate that the features cp, ca, and thal have the highest impact on the model prediction of the presence of patient's heart diseases.

Figure 4.3b report the total natural impact of each features of the Cleveran heart dataset, estimated using our proposed double features swapping functions, with the temporal priority ordering: $sex \rightarrow age \rightarrow thalach \rightarrow ca \rightarrow thal \rightarrow Medu \rightarrow exang \rightarrow cp \rightarrow trestbps \rightarrow restecg \rightarrow fbs \rightarrow oldpeak \rightarrow chol$. The double features swapping estimate the total natural impact of each feature through the mediating variables. The reported results in Figure 4.3b indicates the total sum of the pairwise swapping of each feature and its respective mediating



(a) Single feature swapping, indicating the controlled direct impact of each feature in predicting the presence of heart disease.



(b) The results of the double features swapping functions to show the total natural effect of each feature on the model prediction.

Figure 4.3 The experimental results in terms of the statistical distance measure between the distribution of original model prediction and the single feature swapping function in Figure 4.3a, and double features swapping functions in Figure 4.3b for the Cleveran Heart dataset, with the swap percentage of $(10\%, 30\%, 50\%$, and $70\%)$

variables; refer to Table 4.2. According to Figure 4.3b, the features that consistently show the higher total natural impact on the model predictions across the four divergence measures are in the order of maximal: 'ca', 'age', 'thalach', and 'sex'. Some of these features, e.g., 'sex', and 'age', originally were not captured as highly impacting the model prediction when the single feature swapping function was used, implying that the natural relation between the 'sex' features and other features (also called mediating variable) that causes mediation in the model prediction was not captured by single feature swapping.

The detailed summary of the features and the respective mediating variables are shown in Table 4.2, for 50% swap percentage. The variables in the column header names starting from 'age' to 'chol' are the mediating variables for the features listed in column 'Feature'. The bold and itelic font face are the maximal value along the rows and the bold face (without italic) are the second highest values. According to Table 4.2, the features that frequently causes the higher values of mediation measures between the prediction of the presence of heart diseases and the given features are the variables: 'ca', 'cp', and 'oldpeak'. Notably, the mediating variables between 'sex' feature and model prediction are frequently the mediated by the variables 'ca' (number of major vessels), and 'trestbps' (i.e., peak exercise blood pressure).

Table 4.2 Summary of the natural effect estimated using the double features swapping on 50% of the **Cleveran Heart dataset**. The mediating variables are shown as column header in the column (from 'age' to chol). We used the temporal priority ordering: $sex \rightarrow age \rightarrow thalach \rightarrow ca \rightarrow thal \rightarrow Medu \rightarrow exang \rightarrow cp \rightarrow trestbps \rightarrow restecg \rightarrow fbs \rightarrow oldpeak \rightarrow chol$. The maximum row value is highlighted in highlighted in ***bold face+italic***, and second higher value along each row is shown in **bold face**.

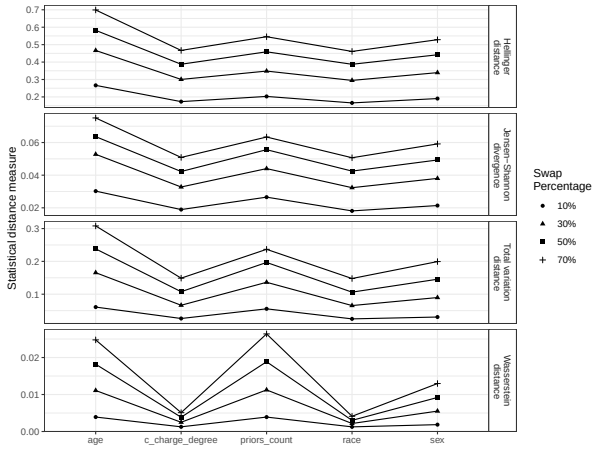
Meası	Feat	age	thalac	ca	thal	exang	cp	trestb	restec	fbs	oldpea	chol
Hellinger distance	sex	0.08	0.07	***0.17***	0.12	0.08	0.09	**0.15**	0.02	0.02	0.14	0.07
	age	-	0.12	***0.17***	**0.16**	0.11	0.14	0.13	0.06	0.06	***0.17***	0.13
	thalach	-	-	***0.17***	0.2	0.09	**0.15**	0.13	0.06	0.05	0.14	0.09
	ca	-	-	-	**0.19**	0.11	***0.22***	0.16	0.11	0.12	0.17	0.17
	thal	-	-	-	-	0.09	**0.14**	0.12	0.07	0.07	***0.15***	0.1
	exang	-	-	-	-	-	***0.13***	**0.12**	0.05	0.03	0.11	0.08
	cp	-	-	-	-	-	-	***0.18***	0.1	0.08	**0.17**	0.12
	trestbps	-	-	-	-	-	-	-	0.09	0.1	***0.16***	0.13

◉ **Answer to RQ1 (Cleveran Heart Dataset).** *As shown from our results, the features such as chest pain type (cp), number of major vessels (ca), and 'thal' have the most effect on the model prediction of the presence of heart disease when swapping values of each feature (keeping other features unchanged). On the other hand, the features such as resting electrocardiographic results (restecg), and fasting blood sugar (fbs) frequently show minimal impact on predicting the patients' heart disease. The features that are observed to cause the maximal mediation between the model prediction and other features are: 'ca', 'cp', and 'oldpeak' (i.e., depression due to exercise relative to rest). Moreover, the existence of mediating variables causes the features such as 'age', 'sex', 'ca', and 'thalach' to introduce more bias to the machine learning model trained on the cleveran heart dataset.*

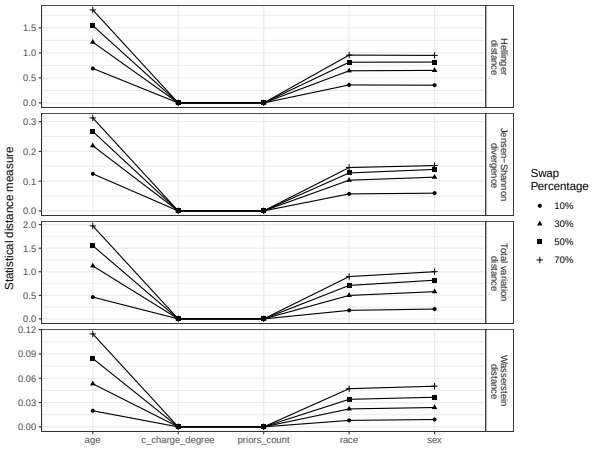
COMPAS Recidivism Dataset

This Subsection report the results of experimenting with our proposed swapping functions on the COMPAS Recidivism dataset. Like the other dataset used in this study, we fit a logistic regression model on the modified COMPAS Recidivism dataset. Prior to fitting the model, a number of preprocessing was done including removing the duplicated and correlated features (e.g., 'decile_score','priors_count'), dropping features such as 'id', 'name', 'first',

'last', 'compas_screening_date','dob'. Changing symbolics to numerics. The final features used to train and test the model include sex, race, priors_count, c_charge_degree, and age. The goal variable includes predicting whether the criminal is likely to offend in the future or not (recidivism/crime risk). Figure 4.4 report the experimental results of the proposed swapping function on the COMPAS Recidivism dataset comparing the controlled direct impact Figure 4.4a and the total natural effect (Figure 4.4b) of each feature on the model prediction.



(a) The results of single feature swapping, demonstrating the controlled direct impact of the feature on the model prediction



(b) The results of the double features swapping functions to show the total natural effect of each feature on the model prediction.

Figure 4.4 The experimental results in terms of the statistical distance measure between the distribution of original model prediction and the single feature swapping function in Figure 4.4a, and double features swapping functions in Figure 4.4b for the Cleveran Heart dataset, with the swap percentage of $(10\%, 30\%, 50\%$, and $70\%)$

Figure 4.4a shows the experimental results of the single feature swapping function on the COMPAS Recidivism dataset, demonstrating the direct impact of the features on the model prediction when the swap percentage are: $(10\%, 30\%, 50\%, 70\%)$. As seen in Figure 4.4a, the statistical distance increases with the swap percentage for all four divergence measures. Also, the ordering of the distance measures in Figure 4.4a is closely similar across all the four divergence measures, where the feature 'age' shows the maximal values of statistical distance measure, followed by the 'prior_count' feature. On the other hand, the feature c_charge_degree and 'race' shows the minimal values of statistical distance for a single feature swapping function.

Figure 4.4b shows the total natural effect of the features and mediating variables estimated using the double features swapping function on COMPAS dataset, with the temporal priority

Table 4.3 Summary of the statistical distance measure for the 50% swap percentage of the double features swapping on **COMPAS Recidivism dataset**, demonstrating the impact of each feature on the first columns on the model prediction through the mediating variables (i.e., first rows), with temporal priority ordering: *race* → *sex* → *c_charge_degree* → *priors_count*. The values are computed by keeping the value of the features (first column) fixed at some level, while altering the variables in the first rows. ***bold face+italic***, where as, the second higher values along the rows are shown in **bold face**

Measure	Feature	sex	age	c_charge_degree	priors_count
Hellinger distance	race	0.67	***0.98***	0.61	0.59
	sex	-	***0.91***	0.68	0.64
	age	-	-	***0.93***	1.01
	c_charge_degree	-	-	-	0.64

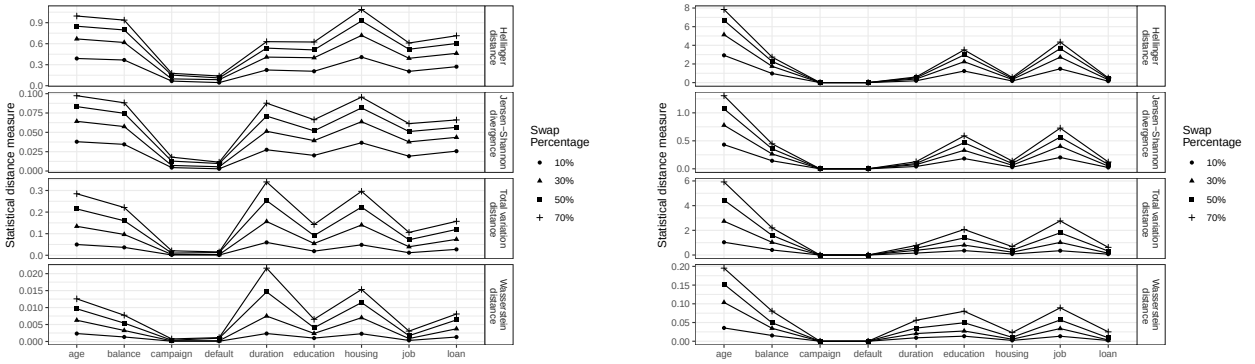
ordering of: *race* → *sex* → *age* → *c_charge_degree* → *priors_count*. As can be seen in Figure 4.4b, the feature 'age' remains maximal, followed by the sex and race, while the feature 'c_charge_degree' show the minimal statistical distance. The breakdown of the double features swapping functions is shown in Table 4.3, for the 50% swap percentage. According to Table 4.3, the age feature indicates the maximal causes of mediation between the model prediction and other features.

> ◉ **Answer to RQ1 (COMPAS Dataset).** *The age feature has a maximal statistical distance compared to other features, implying that it exhibits the highest direct impact and the total natural impact on the model prediction. Moreover, the age feature plays a significant role as the mediating variable in the COMPAS dataset, causing features such as race and sex to be more biased to the model prediction.*

Bank Dataset

In the final part of our experiment, we empirically evaluated our proposed swapping functions using the bank dataset and reported the results in this Section. This dataset contains information about the marketing campaigns of a banking institution in Portugal, in which the goal includes predicting if the client would subscribe to a term deposit. Figure 4.5 shows the experimental results of swapping values of the features in the bank dataset, to understand the impact of each features on the model prediction.

In the first part of Figure 4.5a, we show the results of the single feature swapping function on the Bank dataset, estimating the controlled direct impact of each feature on the model



(a) The evaluation results of swapping values of each feature on prediction

(b) The experimental results of double features swapping on the model prediction.

Figure 4.5 The experimental results in terms of the statistical distance measure between the distribution of original model prediction and the single feature swapping function in Figure 4.5a, and double features swapping functions in Figure 4.5b for the Bank dataset, with the swap percentage of (10%, 30%, 50%, and 70%)

prediction. According to Figure 4.5a, the ordering of the statistical distances is similar except for the small difference in the ordering for the feature duration with education when Hellinger distance is used, compared to the other three distance measures. Overall, the features 'age' and 'housing' show the higher statistical distances for single feature swapping, indicating their high impact on the model prediction. On the other hand, the features 'campaign' and 'default' have minimum values of statistical distance, showing how less they impact the model outcomes.

Next, in Figure 4.5b we visualize the total natural impact of each feature of the Bank dataset when the temporal priority ordering was set as $age \rightarrow education \rightarrow job \rightarrow loan \rightarrow balance \rightarrow housing \rightarrow duration \rightarrow campaign \rightarrow default$; refer to Table 4.4 for the pairwise summary of each feature and the mediating variables. According to Figure 4.5b, the features that consistently show the higher statistical distance for the double features swapping evaluated across the four divergence measures are in the order of maximal: 'age', 'sex', 'education', and 'balance'. Moreover, the 'age' features remained consistent as the ones that highly impacted the model prediction for both single and double feature swapping functions. From these results, we can say that the model prediction strongly depends on the 'age' feature for the model trained on the bank dataset.

The summary details of each feature and the respective mediating variables are shown in Table 4.4, for 50% swap percentage. The variable names in the column header, starting from 'education' to 'default' are the mediating variables for the features listed in column

Table 4.4 Impact of each feature on the first columns on the model prediction through the mediating variables (i.e., first rows), with temporal priority ordering: $age \rightarrow education \rightarrow job \rightarrow loan \rightarrow balance \rightarrow housing \rightarrow duration \rightarrow campaign \rightarrow default$. The values are computed by keeping the value of the features (first column) fixed at some level, while altering the variables in the first rows. The maximum row value is highlighted in highlighted in **bold face**.

Measu	Featur	educat	job	loan	balanc	housin	duratic	campai	default
Hellinger distance	age	1.23	1.34	1.38	**1.55**	1.53	***1.78***	1.47	0.9
	education	-	1.2	1.22	**1.47**	1.38	***1.55***	1.34	0.73
	job	-	-	1.04	1.32	1.23	**1.58**	***1.17***	**0.58**
	loan	-	-	-	**1.39**	1.35	***1.5***	1.08	0.67
	balance	-	-	-	-	**1.64**	***1.87***	1.49	0.99
	housing	-	-	-	-	-	***1.62***	1.53	0.96
	duration	-	-	-	-	-	-	1.65	0.61

'Feature'. The bold and italic font face are the maximal value along the rows, and the bold face (without italic) is the second highest value. According to Table 4.4, the 'duration' and 'balance' are the two features in the Bank dataset that frequently causes the higher values of mediation measures between the model prediction and other features. For example, the 'age' feature is highly mediated by the 'duration' variable followed by the 'balance' variable, and the minimal mediation is by the 'default' variable.

> **Answer to RQ1 (Bank Dataset).** *As discussed above, the features such as 'age' and 'housing' show the maximal controlled direct impact on the model prediction compared to features 'campaign' and 'default'. Moreover, the 'age', 'sex', 'education', and 'balance' features demonstrate a higher total natural impact, primarily due to the two mediating variables: duration' and 'balance'.*

4.8.2 RQ2: Given these identified biased features, can we assert whether or not they are important to the model?

In this section, we aim to examine whether the features that we identified to be the potential source of bias to the machine learning model are also important feature to the model (Feature importance). We used the SHAP (SHapley Additive exPlanations) value to assess the feature's importance. SHAP value a state-of-the-art explainable method to help us explain the output of the machine learning model; refer to Section 4.7. First, we compute the SHAP

values for each instance of the sampled dataset. The similar instance of data used as inputs for our data swapping function was used in this step (i.e., $10\%, 30\%, 50\%,$ and 70% of the test set). Then, the importance of the feature is computed by taking the average of the absolute values of the SHAP values for all instances.

In the following, we present our results in the same order of the presentation for **RQ1** where each subsection corresponds to the dataset used in the experiment, described in Section 4.8.1 (i.e., Student Data, Cleveran Heart Dataset, COMPAS Dataset, and Bank Dataset).

Feature importance for Student Dataset

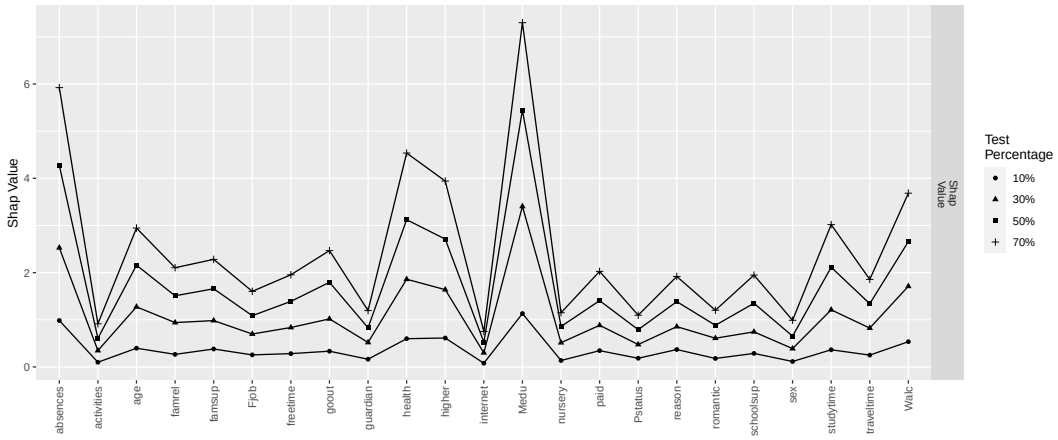


Figure 4.6 The feature importance for the Student dataset, computed by running SHAP on the similar data points used as input by the swapping functions.

Figure 4.6 shows the average absolute SHAP values for each feature in the Student Dataset, demonstrating the importance of a given feature in the model prediction. The higher SHAP value of a feature is an indication of the higher importance of that feature on the model outcome. As can be seen in Figure 4.6, the SHAP values of each features increases with the number of data points used, similar to the results of swapping function, discussed in Section 4.8.1.

From Figure 4.6, we can see that 'Medu' consistently show the highest SHAP values when tested on the different percentage of the test dataset. Furthermore, the ordering is generally consistent for the test percentages $30\%, 50\%, and 70\%$ and closely similar for the 10%, where the ordering from maximum is: 'Medu' , absences, health, higher, and Walc. On the other-hand, the features with the minimal SHAP values are in the order: 'activities', 'guardian', 'nursery' and 'sex'

If we compare Figure 4.6 with Figure 4.2, i.e., the results of SHAP values with the results

of our swapping function, we can clearly see that the ordering of the SHAP values and the statistical distance are closely similar for the single feature swapping function shown in Figure 4.2a. To better understand the relationship between the SHAP values and our swapping functions, we introduce the concept of feature ranking stability in the following Subsection.

Feature Ranking Stability: We compute the stability of the rankings of features returned by SHAP and those returned by our swapping function to understand the consistency in the ranking order of the features from the two results. Specifically, we used Spearmans' coefficient, following the idea from [160], to compare how the results of a ranking order are similar when different distance measures are used as follows:

$$(\text{stability}) = 1 - \sum_i \frac{(r_{\Phi_i} - r_{\Im_i})^2}{m \cdot (m^2 - 1)} \tag{4.17}$$

In Equation (4.17), m is the number of features, and $r_\Phi, r_\Im$ are the rankings by SHAP value and a given divergence measure, respectively. For example, SHAP Values and Hellinger distance, or SHAP Values and Jensen-divergence distance. r_{Φ_i} and $r_{\Im_i}$ also denote the ranks of feature i in rankings r_Φ and $r_\Im$, respectively. Specifically, we computed the ranking stability between the SHAP value and the results single feature swapping function, reported in Figure 4.2a.

Table 4.5 Feature rankings Stability measure between the ranking for SHAP value and our swapping functions for **Student dataset**

Distance Measure	**SHAP vs Single**	**SHAP vs Double**
Hellinger distance	0.974	0.881
Jensen-Shannon divergence	0.968	0.871
Total variation distance	0.959	0.868
Wasserstein distance	0.934	0.89

In Table 4.5, we report the feature ranking stability for the Student Dataset. The column 'SHAP vs. Single swapping' correspond to the stability measure between the ranking generated from the distance measure of the single feature swapping function and the SHAP values ranking. On the other hand, the column named 'SHAP vs. Single swapping' in Table 4.5 is between the SHAP value and the Double features swapping function. Intuitively, A higher stability coefficient (close to 1) indicates that the ranking of the features was close to each other, and the rankings are consistent between the two measures.

Table 4.6 Ranking of Feature by SHAP values vs. double features swapping, for the Student dataset. $r_{\Im}$ is the ranking score based on the result of double feature swapping function, r_{Φ} is the ranking based on SHAP value. The feature that is a potential source of bias (smaller $r_{\Im}$), yet is detected as less important to the model (large r_{Φ}) is highlighted in **bold face**.

Distance Measure	**Feature**	$r_{\Im}$	r_{Φ}	$\lvert r_{\Phi} - r_{\Im} \rvert$
Hellinger distance	sex	6	21	15
	age	3	6	3
	health	2	3	1
	Pstatus	4	20	16
	nursery	7	18	11
	Medu	1	1	0
	Fjob	8	16	8
	schoolsup	9	14	5
	absences	5	2	3
	activities	11	22	11
	higher	10	4	6
	traveltime	13	15	2
	paid	14	11	3
	guardian	18	19	1
	Walc	12	5	7
	freetime	15	12	3
	famsup	16	9	7
	romantic	19	17	2
	studytime	17	7	10
	goout	20	8	12

Looking at the stability measures in Table 4.5, we can say that the controlled direct impact of the feature can also be used to explain the important features of the model. This is, however, not true when the mediating variables are considered when estimating the feature that is a potential source of bias. Table 4.6 shows the ranking of the features based on the results of double feature swapping function (i.e., $r_{\Im}$), and ranking of SHAP values (i.e., column r_{Φ}), the magnitude of the difference (i.e., $|r_{\Phi} - r_{\Im}|$). The lower value of $r_{\Im}$ indicates that the respective feature is estimated to be more likely the respective feature introduces bias to the model. For the feature importance, the lower the value of r_{Φ}), the more important the feature is to the model.

> ◉ **Answer to RQ2 (Student Dataset).** *The features 'PStatus', and 'nursery' are less important according to its SHAP value. Yet, 'Pstatus', and 'nursery' are also potential source of bias. These kinds of features could be removed from the Student dataset to improve the trained model's fairness.*

Feature importance for Cleveran Heart Dataset

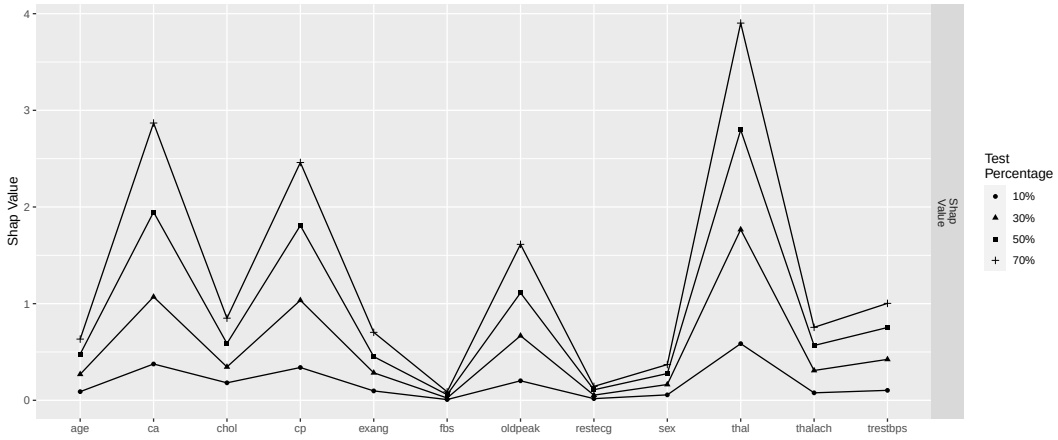


Figure 4.7 The feature importance for the Cleveran Heart dataset, computed by running SHAP on the similar data points used as input by the swapping functions.

Figure 4.7 shows the summary of the average absolute SHAP values indicating the importance of each feature in the Student Dataset. The feature with the large SHAP value show its more important. In Figure 4.7, we see that feature can be arranged in the order of importance as 'thal', 'ca', 'cp', and 'oldpeak'. The close ordering was also observed for the single feature swapping function reported in Figure 4.3a

Table 4.7 Feature rankings Stability measure between the ranking for SHAP value and our swapping functions for **Cleveran Heart dataset**

Distance Measure	**SHAP vs Single**	**SHAP vs Double**
Hellinger distance	0.985	0.879
Jensen-Shannon divergence	0.988	0.865
Total variation distance	0.99	0.846
Wasserstein distance	0.986	0.883

Table 4.8 Ranking of Feature by SHAP values vs. double features swapping, for the bank dataset. $r_{\Im}$ is the ranking score based on the double feature swapping function, r_{Φ} is the ranking based on SHAP value, for **Cleveran Heart dataset**

Distance Measure	Feature	$r_{\Im}$	r_{Φ}	$\lvert r_{\Phi} - r_{\Im} \rvert$
Hellinger distance	**sex**	4	10	6
	age	2	8	6
	thalach	3	7	4
	ca	1	2	1
	thal	5	1	4
	exang	7	9	2
	cp	6	3	3
	trestbps	8	5	3
	restecg	11	11	0
	fbs	9	12	3
	oldpeak	10	4	6
	chol	12	6	6

Feature Ranking Stability: Table 4.7 report the feature ranking stability between the SHAP value and feature swapping functions using Equation 4.17. According to Table 4.7, the ordering of the ranking of the features is closely similar between the SHAP value and the single feature swapping function. The closest similarity in the ordering was observed between the SHAP value and the Total variation distance measure for single figure swapping. For instance, the features detected to be more important to the model are similar detected to impose the controlled impact on the model prediction.

When comparing the double features swapping functions with SHAP values, we observe that the ranking of some of the features is not close, implying a feature that might be important could also be biased, or vice-versa. In Table 4.8 reports the ranking of the features by the double feature swapping function (i.e., $r_{\Im}$), and ranking of SHAP values (i.e., column r_{Φ}), and their difference. Feature (s) detected as less important and more biased are highlighted in **bold face**.

> ◉ **Answer to RQ2 (Cleveran Heart Dataset).** *The feature 'age', and 'sex' are less important according to its SHAP value. Yet, they are potential source of bias to the model. These features could be removed from the Cleverant Heart dataset in order to*

improve the model's fairness.

Feature importance for COMPAS Recidivism Dataset

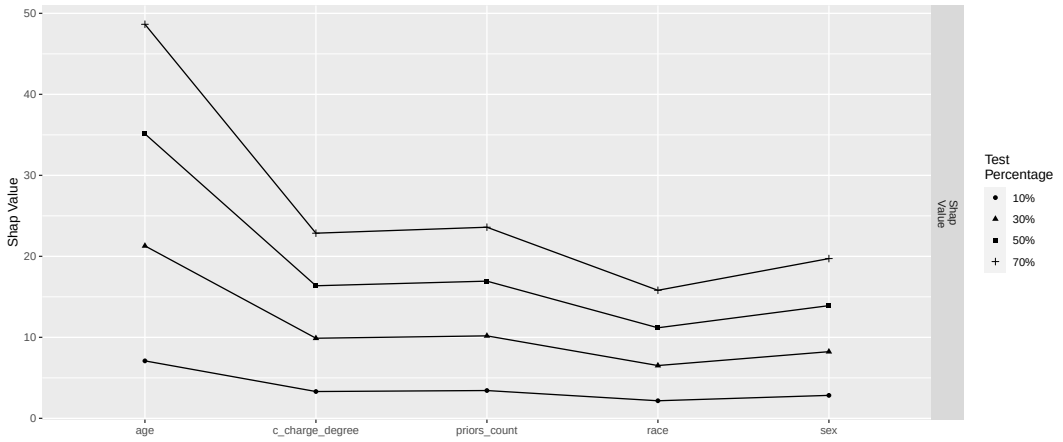


Figure 4.8 The feature importance for the COMPAS Recidivism dataset, computed by running SHAP on the similar data points used as input by the swapping functions.

In Figure 4.8, we reported the results of averaging the SHAP values for each features of COMPAS dataset. The feature with the maximal SHAP value are more important. Looking at Figure! its more important. In Figure 4.8, the feature 'age' is shown to be more important followed by the c_charge_degree.

Table 4.9 Feature rankings Stability measure between the ranking for SHAP value and our swapping functions for **COMPAS Recidivism dataset**

Distance Measure	**SHAP vs Single**	**SHAP vs Double**
Hellinger distance	0.95	0.717
Jensen-Shannon divergence	0.95	0.717
Total variation distance	0.95	0.717
Wasserstein distance	0.933	0.8

Comparing the SHAP values with the swapping functions, we compute the feature ranking stability using Equation (4.17) to examine the consistency in the two ranking orders. The results of Equation (4.17) is shown in Table 4.9. The divergence measures that closely agree with the SHAP value ranking are Hellinger distance, Jensen-Shannon divergence, and Total variation distance, when measured for single feature swapping function results. The double

Table 4.10 Ranking of Feature by SHAP values vs. double features swapping, for the COMPAS Recidivism dataset. $r_{\Im}$ is the ranking score by the double feature swapping function, r_{Φ} is the ranking based on the SHAP value.

Distance Measure	Feature	$r_{\Im}$	r_{Φ}	$\lvert r_{\Phi} - r_{\Im} \rvert$
Hellinger distance	**race**	1	5	4
	sex	2	4	2
	age	3	1	2
	c_charge_degree	4	3	1
	priors_count	5	2	3

features swapping functions results do not closely agree with the ranking order of the features detected as bias vs. those detected by SHAP as important features.

Table 4.10 details the ranking order of SHAP value (column r_{Φ}) and the ranking order for double feature swapping function (i.e., $r_{\Im}$), and the absolute difference as $|r_{\Phi} - r_{\Im}|$. The less important feature (s) and yet are more biased are highlighted in **bold face**. According to Table 4.10, 'race' consistently show the highest bias and less important between three of the four distance measures vs. SHAP values.

> ◉ **Answer to RQ2 (COMPAS Dataset).** *The features 'race', and 'sex' in COMPAS dataset are potential source of bias and yet less important to the model prediction. As interpreted by the domain expert, such features could be removed (or constrainted) from the data to ensure a more fairer ML model.*

Feature importance for Bank Dataset

Finally, here we want to examine the feature's importance and relation to the bank dataset's potential bias.

Figure 4.8 report the average absolute SHAP values for the features in Bank dataset. The features with higher SHAP values that the rest indicate are more important to the model prediction. From Figure 4.8, we can see that 'duration' consistently shows the highest SHAP value; hence is a more important feature.

Next, we compare the relation in the ranking between the feature importance and bias measures using the swapping functions (Using Equation (4.17)), and report the feature ranking stability in Table 4.11. The results show the higher stability between the SHAP values and single feature swapping function, especially for the Jensen-Shannon divergence and Total

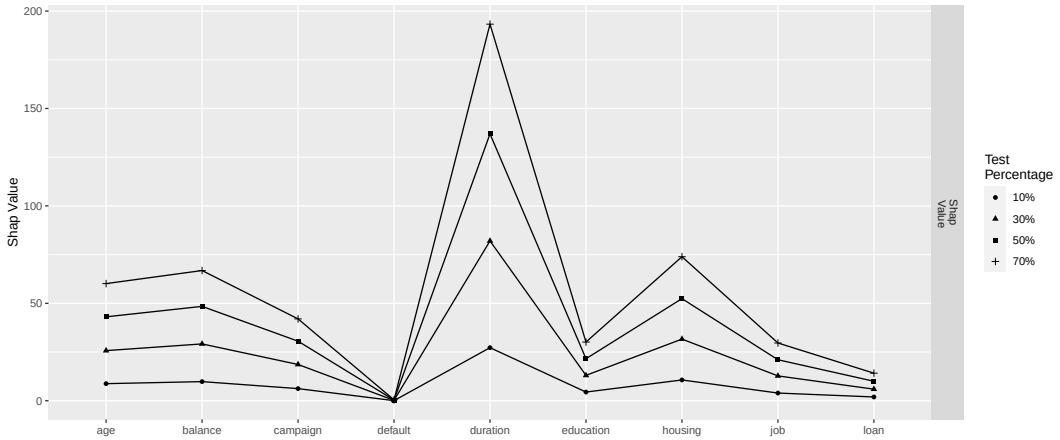


Figure 4.9 The feature importance for the Bank dataset, computed by running SHAP on the similar data points used as input by the swapping functions.

Table 4.11 Feature rankings Stability measure between the ranking for SHAP value and our swapping functions for **Bank dataset**

Distance Measure	**SHAP vs Single**	**SHAP vs Double**
Hellinger distance	0.931	0.831
Jensen-Shannon divergence	0.961	0.831
Total variation distance	0.978	0.831
Wasserstein distance	0.947	0.842

variation distance. We detailed the summary ranking of the feature importance, and bias detected using double features swapping in Table 4.12. The results show that features like education, job, and loan are more biased and have minimal importance to the model prediction.

> ◉ **Answer to RQ2 (Bank Dataset).** *The features such as 'education', 'loan', and 'job' might be more biased yet are less important to the model prediction. These features could be removed from the dataset to improve the model's fairness. On the other hand, features such as 'duration', 'housing', and 'balance' are more important and less biased. The 'default' is the least important to the model, and neither is a potential source of bias to the model.*

Table 4.12 Ranking of Feature by SHAP values vs. double features swapping, for the bank dataset. $r_{\Im}$ is the ranking score by the double feature swapping function, r_{Φ} is the ranking of the SHAP value

Distance Measure	Feature	$r_{\Im}$	r_{Φ}	$\lvert r_{\Phi} - r_{\Im} \rvert$
Hellinger distance	age	1	4	3
	education	2	6	4
	job	3	7	4
	loan	4	8	4
	balance	5	3	2
	housing	6	2	4
	duration	7	1	6
	campaign	8	5	3
	default	9	9	0

4.9 Discussion

The controlled direct impact can help us explain the impact of each feature on the model prediction closely similar to SHAP value. However, the conclusion drawn from such analysis should be supported by further analysis. For example, we have shown in this study that concluding that such features are biased by a simple perturbation of given feature values when other variables are kept unchanged can be misleading (such as [20–22]). A genuine conclusion should consider the effect of confounding.

Indeed, if we assume causation in which some of the relevant features are influenced by a potential bias feature, then computing counterfactuals for this biased feature would require altering downstream features [132, 134–137, 140, 141]. Changing the values of the potential bias feature alone will only correspond to a counterfactual if this feature does not have any mediator variables, which is unlikely [140].

Motivated by the plaintiff's expert report [161,162] that claimed that race played a significant role in admissions decisions of the Harvard's machine learning model [1] *"Consider the case of a male applicant who is an Asian-American, is not disadvantaged, and has other characteristics that result in a 25% chance. The chance of the applicant was observed to increase to 36% when only the applicant's race was changed to white— and keeping all his other characteristics the same. On the other hand, the applicant's chance of being admitted increased to 77% when his race is changed to Hispanic while keeping his other characteristics constant). When*

[1]Plaintiff's expert report of Peter S. Arcidiacono, Professor of Economics at Duke University.

the race was changed to African-American, leaving all other features constant increased the chance of his admission to 95%.". A logistic regression model was fitted against the historical admissions decisions regarding features considered relevant for Harvard's admission decision to model Harvard's decision rule above. The plaintiff's report above, however, can be translated into the technical claim that the model used admissions decisions did not satisfy the conditional statistical parity. Formally, let X be the set of applicant features and S as the applicant's reported race. If we use $\hat{Y} = \hat{y}$ to denote the model admissions decision deemed relevant for admission, then we say:

$$\frac{\rho(\hat{y}|X = x, S \neq s)}{\rho(\hat{y}|X = x, S = s)} \leq \tau$$

where $\rho(\hat{y}|X = x, S \neq s)$ denotes the conditional probability (evaluated over X) that the class outcome is $\hat{y} \in \hat{Y}$ given protected race group $S \neq s$ and $P(\hat{y}|X = x, S = s)$ is the conditional probability given non-protected group $S := s$, and τ is the allowed ratio, such as the 80% rule. For this condition to be violated depends solely on which feature was considered relevant for the admission, which to a large extent will correspond to the defendant's expert [163].

In this study, we propose an approach for systematically identifying all bias-inducing features of a model to help support the decision-making of domain experts. The proposed method considers both the case of using direct and indirect impact of each features on the model prediction using a novel technique of data swapping. We defined a single feature swapping function that modifies the values of the single feature keeping other features unchanged, and the function's output is used to estimate the direct impact of the feature on the model prediction. On the other hand, we proposed the double features swapping function will switch the values of pairs consisting of the feature and all the mediating variables used to estimate the total natural impact of the feature on the model prediction.

We demonstrate the usefulness of our approach in the decision-making by the domain experts on how our framework can spearhead the development, testing, maintenance, and deployment of fair machine learning systems. Specifically, we show how our proposed framework can be used to rank all the features based on the level of bias-inducing capability. We showed that when the single feature swapping function is used, the resulting conclusion can be used to explain the feature importance with the result closely similar to the SHAP value, a state-of-the-art method to explain the impact of each feature on the model prediction based on the famous Shapley values from game-theory. Then, we showed how the double feature swapping functions that consider the mediating variables following the concept of probabilistic causal; the resulting outcome can be used to estimate the total natural impact of the features on the model prediction. We demonstrated that the total natural impact, when integrated with the

state-of-the-art model interpretability tool for explaining the most relevant or least relevant features, will provide better insights into what features are most relevant or least relevant to the model prediction and are least bias-inducing as interpreted by the domain experts. The resulting insights will help the domain expert choose the features that improve predictive performance and fairer machine learning models.

4.10 Summary

In this study, we proposed approach for systematical identification bias inducing feature of the machine learning model based on data swapping technique. Two different kinds of swapping functions are proposed. First, a single feature swapping function that modifies the values of the single feature keeping other features unchanged, and to help estimate the direct impact of the feature on the model prediction. Second, the double features swapping functions which switch the values of pairs consisting of the feature and all the mediating variables and the resulting outputs are used to estimate the total natural impact of the feature on the model prediction. Four different distance measures (i.e., Hellinger distance, Jensen-Shannon divergence, total variation distance, and Wasserstein distance) are used to evaluate the impact of swapping the features using our proposed functions on the model prediction.

We demonstrate by answering two main research questions how the domain experts can use our proposed approach to identify the potentially bias-inducing features. We showed with the help of the state-of-the-art model explainability SHAP tool that the potential bias-inducing features that are less important to the model can be removed to improve the fairness of the machine learning model. Our study is the first step in helping domain experts make an informed decision by following a systematic identification of bias-inducing features. The domain experts can use our approach to visualize the cause of bias in the model, systematic features selection, prioritizing the fixes, and therefore helping contribute to the standard procedure when developing, maintaining, and deploying fairer machine learning systems.

CHAPTER 5 FAIRLFREP: FAIRNESS AWARE FAULT LOCALIZATION AND REPAIR OF DEEP NEURAL NETWORKS

This Chapter introduces FairFLRep, an automated fairness-aware fault localization and repair technique that identifies and corrects potentially bias-inducing neurons in DNN classifiers. FairFLRep focuses on adjusting neuron weights associated with sensitive attributes, such as race or gender, that contribute to unfair decisions. By analyzing the input-output relationships within the network, FairFLRep corrects neurons responsible for disparities in predictive quality parity. We evaluate FairFLRep on four image classification datasets using two DNN classifiers, and four tabular datasets with a DNN model. The results show that FairFLRep consistently outperforms existing methods in improving fairness while preserving accuracy. An ablation study confirms the importance of considering fairness during both fault localization and repair stages. Our findings also show that FairFLRep is more efficient than the baseline approaches in repairing the network.

5.1 Context of the Study

DNNs are prone to inheriting biases present in their training data. These biases can manifest in unfair predictions, disproportionately affecting certain sensitive groups. For example, a facial recognition model may show higher misclassification rates for darker-skinned individuals compared to lighter-skinned individuals, as shown in [8, 9]. In such cases, DNN models may perpetuate or even amplify societal inequalities. This is particularly concerning for Convolutional Neural Networks (CNNs), which are widely used, as they may unintentionally learn to prioritize features correlated with sensitive attributes, such as race or gender, leading to unfair outcomes [10–12].

As DNNs are increasingly integrated into decision-making systems that impact people's lives, addressing fairness issues has become critical. Unfair models can cause considerable harm, particularly in sensitive applications like hiring, loan approvals, or facial recognition. These biases not only lead to unjust outcomes but also erode public trust in AI systems [49, 164]. Fairness of AI systems is now required by different international regulations; for example, the EU AI Act, in its Recital 27, requires that AI systems avoid "discriminatory impacts and unfair biases"; the U.S. Equal Employment Opportunity Commission (EEOC) in its guidelines requires that the selection rate for an unprivileged group should be at least 80% of the selection rate for the privileged group [165]; GDPR requires that data is "processed lawfully, fairly and in a transparent manner in relation to the data subject" [166].

Traditionally, methods to address fairness in DNNs have focused on *pre-processing* (modifying the training data), *in-processing* (modifying the model's training process), or *post-processing* (adjusting predictions after training) [24]. While these techniques can mitigate some bias, they come with limitations. Pre-processing, for example, balances the training data but does not always resolve the root cause of bias encoded in the network's learned representations, as neural networks can still pick up subtle correlations between features and sensitive attributes. In-processing methods, which modify the training process, or post-processing techniques, which adjust predictions, can also be computationally expensive, require retraining, and may fail to generalize well across different datasets. Moreover, retraining the model—while capable of modifying weights—might not efficiently address the specific neurons responsible for biased behavior unless fairness-specific objectives are incorporated into the training process. Retraining can be computationally costly, especially for large models, and may have unintended consequences such as degrading overall model accuracy without fully addressing fairness concerns. This highlights the need for more targeted approaches that directly intervene at the level of bias-inducing neurons to achieve fairer outcomes without the overhead of retraining the entire model.

In response to these challenges, we propose applying repair techniques that have proven effective in other contexts—such as model debugging [39, 40, 167–173] (i.e., process of diagnosing, and fixing issues or errors in a DL model) and improving robustness [174]—to mitigate fairness issues in DNNs. Instead of retraining the model or modifying the data, these approaches focus on repairing the trained model by directly targeting and repairing the problematic neurons responsible for bias. This method has been successful in improving model accuracy and robustness in other domains, as demonstrated by `Arachne` [39]. We believe that repair techniques can also effectively mitigate fairness issues in DNNs by targeting specific neural weights responsible for biased decisions. This approach enables precise, neuron-level interventions that improve fairness without significantly affecting model performance. It is particularly advantageous in real-world scenarios where retraining a model from scratch may be computationally infeasible or impractical due to time constraints.

Given the complexity of CNNs and their widespread use, an efficient repair technique that targets biased neurons is particularly necessary. In addition to improving fairness, this method should minimize disruptions to the model's accuracy and be computationally efficient, making it practical for deployment in real-world systems. To address these challenges, we introduce `FairFLRep`, an automated fairness-aware fault localization and repair technique specifically designed for DNN classifiers. In this context, a *fault* refers to a *biased behavior* (or *bias bug*) within the model, where certain neuron weights contribute to unfair predictions against specific sensitive groups. `FairFLRep` identifies potentially biased neurons in a DNN by ana-

lyzing the input-output relationships within the network. The method prioritizes identifying and correcting neuron weights associated with sensitive attributes, such as race or gender, which contribute to unfair decisions. By focusing on the neurons most responsible for biased behavior, `FairFLRep` offers a targeted repair strategy that minimizes the impact on overall model accuracy. `FairFLRep` operates in two key stages:

1. *Fairness-aware fault localization* identifies neural weights in the network that contribute the most to unfair behavior. In this step, the fault is understood as the bias bug embedded in the neural weights, which causes unfair treatment of certain subgroups. The goal is to pinpoint the specific neuron weights that amplify these biases.
2. *Fairness-aware repair* corrects these neural weights to mitigate their contribution to bias, ensuring that the model produces fairer predictions.

We conduct an extensive empirical evaluation of `FairFLRep` across multiple image classification benchmarks (i.e., `FairFace` [175], `UTKFace` [176], Labeled Faces in the Wild (`LFW`) [177], and `CelebA` [178]), tabular classification benchmarks (`Student` [179], `COMPAS` [180], `Adult` [181], and `MEPS` [182]), and fairness metrics (*SPD*, *DI*, *FPR* and *EOD*). Our results reveal that `FairFLRep` not only improves fairness, but also preserves model accuracy, consistently outperforming existing methods such as `Arachne`. Moreover, `FairFLRep` is designed to be computationally efficient, enabling faster execution times and making it scalable for large-scale applications.

In summary, we make the following main contributions:

- We propose `FairFLRep`, a novel fairness-aware fault localization and repair technique designed to address bias in DNNs by analyzing input-output relationships and correcting biased neurons. `FairFLRep` provides flexibility in addressing biases directly tied to sensitive attributes (e.g., race, gender) and those indirectly influencing predictions, ensuring comprehensive bias mitigation.
- We conduct an extensive empirical evaluation across four widely used image classification datasets and four tabular classification benchmarks, demonstrating `FairFLRep`'s superiority over existing methods in improving fairness across multiple fairness metrics (*SPD*, *DI*, *EOD*, *FPR*) while preserving accuracy.
- We perform an ablation study confirming the importance of applying fairness constraints during both the fault localization and repair stages, resulting in more robust fairness improvements.
- `FairFLRep` offers computational efficiency, with significantly faster execution times compared to the baseline approaches, ensuring scalability for large datasets and complex models.

- By explicitly targeting biased neural behaviors and reducing subgroup disparities, `FairFLRep` advances the development of AI systems that are not only technically robust but also legally and ethically compliant with fairness mandates such as those in EU AI Act, GDPR, and EEOC regulation.
- We make the code of `FairFLRep` available online at `https://github.com/openjamoses/FairFLRep`.

Chapter organization Section 5.2 provides background information on DNNs and fairness, offering the foundational concepts necessary for understanding the fairness issues in DNNs and also discussed fairness metrics. Section 3.2 discusses related works on fairness testing and mitigation techniques in the context of DNNs, reviewing existing approaches and their limitations. Section 5.3 details the proposed `FairFLRep` technique introduced in this study, outlining the methodology for fairness-aware fault localization and repair in DNNs. Section 5.4 presents the experiment design used to evaluate `FairFLRep`, including the datasets, models, and the compared approaches. Section 5.5 reports the evaluation results. Section 5.6 outlines potential threats to the validity of this work, acknowledging limitations and areas for further research. Section 5.7 further highlights key insights of our findings, lessons learned, the broader implications and finally concludes the Chapter.

5.2 Preliminary

This section provides the preliminary definitions and technical background required to describe `FairFLRep`. We formalize the classification setting, fairness metrics, and neural network components relevant to fairness-aware fault localization and repair. These preliminaries support an engineering view of fairness, in which unfair behavior is analyzed through the forward and backward dynamics of DNNs and addressed via targeted, post-training interventions.

5.2.1 Notation and Definition

In this study, we focus on addressing the fairness problem within the context of binary classification. Unless otherwise specified when describing a given algorithm or implementation, the following notations are used consistently throughout this Chapter:

- $\mathcal{M}$ – *Model*: denotes the DNN model for binary classification.
- $\mathcal{D}$ – *Dataset*: it is the dataset used in the problem (e.g., training, validation, and test sets). The dataset used in this study consists of binary classification labels (e.g., race,

gender).

- $(\mathcal{X}, Y, \mathcal{A})$: they are the variables associated with the DNN model $\mathcal{M}$ being repaired:
  - $\mathcal{X}$ – *Input Feature Vector*: it is the input data to the model.
  - $Y \in \{0, 1\}$ – *True Label*: the ground truth binary label (class label) for the instance.
  - $\hat{Y}$ – *Predicted label*: it is the label predicted by the model (i.e., $\hat{Y} = \mathcal{M}(\mathcal{X})$).
  - $\mathcal{A} \in \{a_0, a_1\}$ – *Sensitive Attribute* (or *protected attribute*): it is a binary feature in the dataset that is considered sensitive due to its potential to contribute to unfair treatment, discrimination, or bias (e.g., race, gender). For example, if $\mathcal{A}$ is gender, a_0 can represent the female community and a_1 can represent the male community. Unless otherwise specified, we assume that objects satisfying: $\mathcal{A} = a_0$ represent the deprived community (e.g., female), while objects satisfying $\mathcal{A} = a_1$ represent the favored community (e.g., male). However, our method dynamically determines whether a sub-group belongs to the deprived or favored community (see Section 5.3.5).
- In our method, these two datasets are used:
  - $\mathcal{D}rep$ – *Repair Dataset*: it represents the dataset used during the repair process. This dataset is not seen during training but is used to guide the fairness improvements in the model. The data points of $\mathcal{D}rep$ are further categorized into positive and negative sets, defined as follows:
    - $\mathcal{D}rep_{pos}$ – *Positive set*: it is the set of instances in the dataset that are correctly classified by the model, i.e., $\mathcal{D}rep_{pos} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep \mid \hat{Y} = Y\}$.
    - $\mathcal{D}rep_{neg}$ – *Negative set*: it is the set of instances in the dataset that are misclassified by the model (predicted label differs from the true label), i.e., $\mathcal{D}rep_{neg} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep \mid \hat{Y} \neq Y\}$.
  - $\mathcal{D}_{test}$ – *Test Dataset*: it is the dataset used for evaluating the effectiveness of the repair approach and comparing it to baseline models. This dataset is also unseen during training and repair.
- $\mathcal{F}$ – *Fairness Metric*: it represents the fairness metric (e.g., *SPD*, *DI*, *FPR*, *EOD*) used to evaluate fairness in binary classification.

5.2.2 Deep Neural Network

Deep Neural Networks (DNNs) $\mathcal{M}$ are a class of artificial neural networks (ANNs) characterized by multiple layers between the input and output layers. The depth of a DNN allows it to capture intricate patterns and relationships in large datasets, making it suitable for various applications, such as autonomous systems, image and speech recognition, and natural

language processing. A typical DNN consists of: (i) *Input Layer*: it receives the input data. (ii) *Hidden Layers*: they consist of multiple neurons. These layers progressively extract more abstract features as data flows through the network. For example, in image classification tasks, early hidden layers capture basic visual patterns (edges, textures), while deeper hidden layers detect higher-level features (shapes, objects). (iii) *Output Layer*: it produces the final output, typically through a softmax function for classification tasks or a linear function for regression tasks.

Neurons in DNN

Neurons are the fundamental units of a DNN, analogous to biological neurons in the human brain. Each of the neurons in a DNN plays a critical role in extracting and representing features from the input data. As data passes from the input layer to the final prediction layer, the features extracted by neurons become increasingly abstract. In the following, we provide a detailed breakdown of how neurons and their associated weights function within a DNN, and how bias in the model can affect its performance:

Feature Extraction: Neurons receive raw data through the input layer. For example, in image recognition, these might be pixel values. As data moves through hidden layers, neurons extract progressively higher-level features. Early layers might capture simple patterns like edges or textures, while deeper layers might capture complex patterns like shapes or objects. The final layer compiles these abstract features to make a prediction. In classification tasks, this layer outputs probabilities of different classes.

Representation of Features: Each neuron represents a combination of features it receives from the previous layer. The activation function of the neuron determines the output based on the weighted sum of inputs. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU). Weights are parameters that define the importance of each input feature. During training, the model adjusts these weights to minimize error, thereby learning which features are more significant for the task.

Feature Importance and Unfairness

The importance of different features is encoded in the weights. Neurons with higher weights for certain features indicate those features are more critical for the decision-making process. If a model is biased, it might focus on inappropriate features. For instance, in a hiring AI, if the model gives high weight to the gender feature instead of skills or experience, it indicates bias. This can lead to unfair and unethical decisions.

This research focuses on addressing the fairness problem in Convolutional Neural Networks (CNNs). CNNs utilize both forward and backward propagation for learning. Below, we describe how the forward impact and gradient loss are computed, on which our proposed method is based.

Forward impact of the DNN

During the forward propagation, the input data is processed sequentially through each layer of the network to produce an output. Each neuron in the hidden layers accepts the data, performs a computation using its activation function (and weight and bias), and passes the result to the next layer. This process continues until the final output is generated.

Loss Function

Once the neural network output is generated, the loss function $\mathcal{L}$ is used to quantify the difference between the predicted output $\hat{Y}$ and the actual output Y. For example, the cross-entropy loss for classification tasks is defined as: $\mathcal{L}(Y, \hat{Y}) = -\sum Y log(\hat{Y})$.

Backpropagation

It is the method by which the network computes the gradient of the loss function with respect to the model's weights and biases. Using this gradient, the network updates its weights through gradient descent, enabling it to learn from its errors and improve its predictions.

Section 2.4 sets the foundation for understanding the fairness concerns related to the forward and backward propagation processes in DNNs, which play a pivotal role in how bias is embedded and addressed during training. The Background Section 2.3.3 describes how fairness issues manifest in DNNs and methods to measure them.

5.3 Proposed Approach

This section presents the proposed `FairFLRep` framework. We first formalize the problem addressed by `FairFLRep`, then provide a high-level overview of the framework, the repair datasets preparation, and finally describe in detail its two core components: the fairness-aware fault localization stage and the targeted repair stage.

5.3.1 Problem Definition

The goal of this study is to address fairness issues in a binary classification where the *sensitive attribute*, such as gender or race, is either the same as or different from the *target classification label*. The challenge is to mitigate biased behaviour in DNNs, particularly when the network disproportionately misclassifies or treats one group less favorably than another.

The bias bug being localized refers to the systematic unfair treatment or misclassification of certain groups, such as minority races, age, or genders. This bias often manifests as differences in misclassification rates or predictive performance between sensitive groups [183, 184]. For example, a model might misclassify individuals from minority groups at a higher rate than those from the majority group. Addressing these misclassification patterns, especially when they correlate with sensitive attributes, helps reduce bias by forcing the model to treat different groups more equitably [184, 185]. While maintaining accuracy is important, the primary goal of `FairFLRep` is to improve fairness by reducing disparities in predictive outcomes between groups. The objective is to develop a method that identifies and repairs the neuron weights responsible for this biased behavior while preserving the overall accuracy of the model $\mathcal{M}$.

The proposed approach seeks to localize faulty neuron weights (those contributing to biased outcomes) using both gradient loss and forward impact metrics, prioritizing the weights that contribute the most to the unfair treatment of deprived groups. By targeting these biased weights, the goal is to improve fairness in the model while maintaining acceptable accuracy. Before formally stating the optimization problem, we review the key components of our method.

5.3.2 Overview of `FairFLRep`

The overview of the proposed fairness-aware fault localization and repair approach, `FairFLRep`, depicted in Figure 8.3, aims to address bias bugs (unfairness) in DNN models. The approach operates by first identifying biased neuron weights responsible for unfair behavior and then repairing them to ensure fairness, while preserving the model's accuracy. `FairFLRep` operates as follows:

1. ***During the fault localization step***, the process identifies the set of neuron weights that potentially lead to biased behavior in the model's decisions by performing gradient loss analysis and forward impact analysis on each data point for each sub-group in the repair dataset. Both correctly classified (positive) and misclassified (negative) inputs are analyzed, with a stronger emphasis on the negative inputs to correct biased

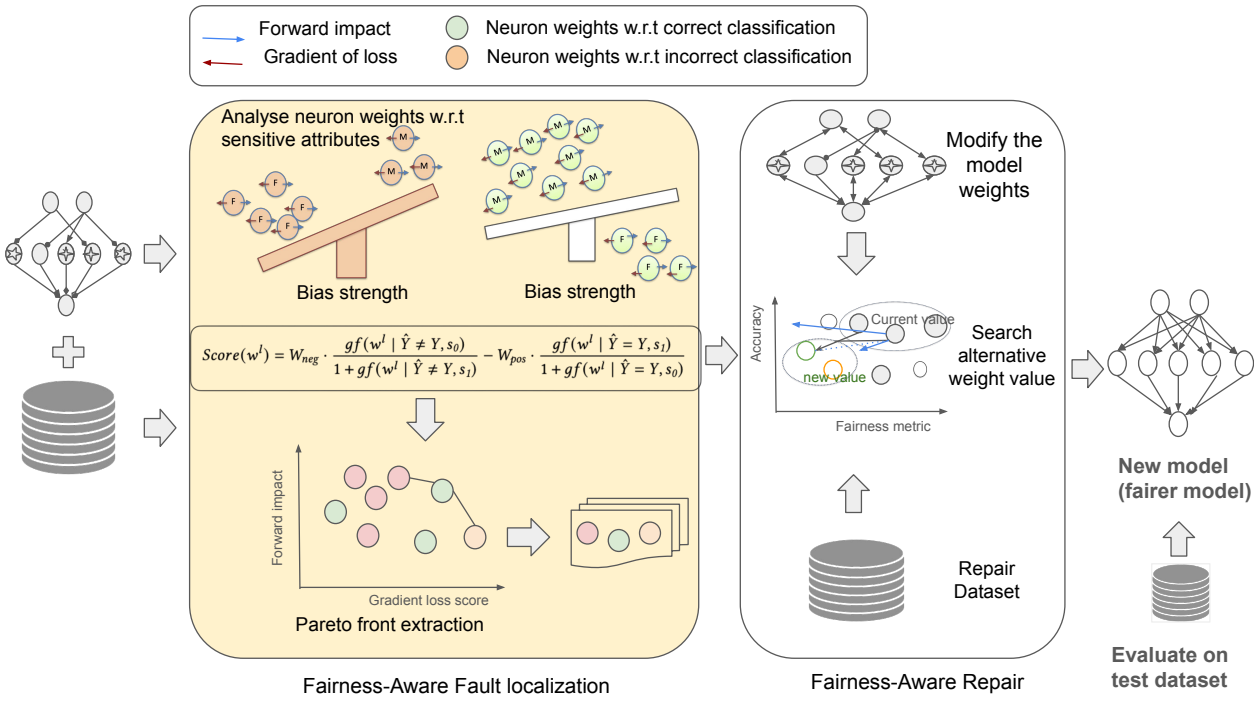


Figure 5.1 Overview of `FairFLRep`

behaviors while preserving correct classifications.

2. ***During the repair phase***, the approach searches for a set of neural weights that would correct the behavior of the DNN under repair based on Particle Swarm Optimization (PSO) [186]. PSO was selected for its effectiveness in optimization and its relatively fast convergence, essential in high-dimensional DNN weight adjustment tasks. During this search, a new model is defined by the search individual (i.e., an assignment to the search variables) that updates the original model with the new weights. A search individual is assessed using a fitness function that defines specific fairness metrics, such as *SPD*, *DI*, *FPR* and *EOD*, provided by the user.

Rationale for Using PSO PSO is utilized in `FairFLRep` due to its advantages in continuous optimization and lower computational demands compared to some other evolutionary algorithms [187], making it a practical choice for fairness repairs where resources may be constrained. PSO's versatility across a range of optimization tasks also makes it effective for balancing fairness and accuracy during model repair. However, alternative evolutionary algorithms like Genetic Algorithms (GA) or Differential Evolution (DE) could also be suitable. For example, GA may offer robustness for tuning discrete parameters [188], while DE is well-suited for complex multimodal optimization landscapes due to its high conver-

gence speed and accuracy [189–191]. This flexibility indicates that `FairFLRep`'s framework could accommodate various optimization algorithms, allowing adaptability based on specific application needs and resource limitations.

We detail the localization and repair phases in the following sections.

5.3.3 Proposed Fault Localization method

In this study, we build on the fault localization method proposed in [39], with a specific focus on fairness. We propose a novel fair-aware fault localization method designed to identify neural weights that are suspected of contributing to unfair behavior in the model. This allows for more effective repairs aimed at addressing unfairness.

Definition 5.3.1. *(Fair-Aware Fault localization approach), denoted as* ***FairFL****, is an approach that identifies the neural weights at layer l in a DNN model $\mathcal{M}$ that contribute to unfair behavior with respect to a sensitive attribute $\mathcal{A}$. Given a repair dataset $\mathcal{D}rep$, the method combines gradient analysis and forward impact to localize these problematic weights and returns top$-k$ weights $\mathbf{w_r}$ with the highest unfairness contributions.*

The fair-aware fault-localization steps proceed as follows:

- *Categorization of data*: The repair dataset $\mathcal{D}rep$ is divided into groups based on the correctness of the model's predictions and sensitive attribute values. The sensitive attribute $\mathcal{A}$ takes on values in $\{a_0, a_1\}$, representing the deprived and favored community, respectively. For instance, if the sensitive attribute is gender, a_0 might correspond to female and a_1 to male. Data points are grouped accordingly based on the sensitive attribute. These subsets are used to examine unfair model predictions and the neural weights impacting such behavior.
- *Estimate the level of bias from input samples*: The method estimates the level of bias based on the prediction outcomes of model $\mathcal{M}$ for different sensitive subgroups, such as the deprived and favored communities. This bias quantification is then used as a constraint when selecting neuron weights corresponding to model behaviors.
- *Neuron analysis*: It is performed with respect to the data samples that differ by the sensitive attribute $\mathcal{A}$. The goal is to identify neuron weights that are highly influential for the deprived and favored sensitive attribute and assess their impact on classification decisions.

  Notably, for each weight connected to the last layer, we calculate the forward impact and gradient loss, respectively. The focus is on the last layer because it is most directly connected to classification, and this also reduces computational complexity, given that

DNNs typically contain thousands of weight parameters. Prior work, such as adversarial debiasing techniques [10, 180], has demonstrated that directly applying mitigation techniques to the output layer promotes model fairness.

- *Scoring function*: A scoring function is employed to prioritize neural weights that maximize the gradient loss and forward impact for the deprived sensitive community relative to the favored community. This function ensures that neural weights contributing to misclassification, especially in the deprived community, are given higher priority. The rationale behind this approach is twofold: to address fairness issues by focusing on correcting biases affecting the deprived community, and to correct misclassifications without negatively impacting the correctly classified instances during the repair process.
- *Pareto front extraction*: The neuron weights $\mathbf{w_r}$ that form a Pareto front in terms of both high forward impact and high gradient loss are selected. A Pareto front consists of weights that cannot be improved in both metrics simultaneously. These weights are more likely to contribute to biased behavior, and improving them can lead to a reduction in bias without compromising model accuracy. Neurons on the Pareto front represent the most influential and biased neurons, making them ideal targets for repair.

Our fair-aware fault localization approach `FairFL` considers two forms of the sensitive attributes:

- `SettSame`: Attribute under prediction (Y) = Sensitive Attribute ($\mathcal{A}$). This case applies when the sensitive attribute (e.g., gender) is the same as the class label.
- `SettDiff`: Class label (Y) $\neq$ Sensitive Attribute ($\mathcal{A}$). This case applies when the sensitive attribute (e.g., race) differs from the class label (e.g., predicting gender but using race as the sensitive attribute).

In the next sections, we explain the details of the categorization, bias estimation, scoring, and repair processes, which together form the `FairFLRep` framework.

5.3.4 Categorization of data

The categorization step provides a framework for analyzing bias in models across two different cases where the attribute under prediction Y and the sensitive attribute $\mathcal{A}$ are the same or differ. This dual-categorization setup enables flexibility in the fault localization approach, allowing for more nuanced identification of unfair behavior of the model $\mathcal{M}$ based on different relationships between the sensitive attribute and class labels. This flexible setup enables the fault localization and repair methods to be adaptable to different scenarios of bias, thus enhancing the fairness of the model across different sensitive attributes and class

labels. Moreover, by isolating and analyzing targeted subgroups (e.g., misclassified minority samples), this categorization significantly reduces the search space during fault localization, thereby lowering the computational cost and improving the efficiency of the repair process.

`SettSame`

In this case, the sensitive attribute is the same as the class label. For example, if the model is predicting gender, the sensitive attribute would also be gender. The categorization is based on whether the data points are correctly or incorrectly classified:

- *Correctly Classified Samples by Sensitive Attribute*:
  - $\mathcal{D}rep_{pos,a_0} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{pos} \mid \mathcal{A} = a_0\}$: This set contains all correctly classified samples where the sensitive attribute $\mathcal{A} = a_0$ (e.g., female).
  - $\mathcal{D}rep_{pos,a_1} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{pos} \mid \mathcal{A} = a_1\}$: This set contains all correctly classified samples where the sensitive attribute $\mathcal{A} = a_1$ (e.g., male).
- *Misclassified Samples by Sensitive Attribute*:
  - $\mathcal{D}rep_{neg,a_0} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{neg} \mid \mathcal{A} = a_0\}$: This set contains all misclassified samples where the sensitive attribute $\mathcal{A} = a_0$ (e.g., female).
  - $\mathcal{D}rep_{neg,a_1} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{neg} \mid \mathcal{A} = a_1\}$: This set contains all misclassified samples where the sensitive attribute $\mathcal{A} = a_1$ (e.g., male).

This setup directly targets the bias in the sensitive attribute by grouping the data points into misclassified and correctly classified categories. The goal is to identify neuron weights that contribute to misclassification, particularly for instances where the model disproportionately misclassifies data points from certain sensitive groups.

`SettDiff`

In this case, the sensitive attribute and class label are different. For example, the model might be predicting gender, but the sensitive attribute is race. The categorization is based on the class label $Y \in \{0, 1\}$, and sensitive attribute $\mathcal{A} \in \{a_0, a_1\}$.

- Samples belonging to class $Y = 1$ (e.g., gender=male).
  - $\mathcal{D}rep_{1,a_0} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{pos} \mid \mathcal{A} = a_0, Y = 1\}$: instances where $Y = 1$ and the sensitive attribute $\mathcal{A} = a_0$ (e.g., gender=male, race = minority).
  - $\mathcal{D}rep_{1,a_1} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{pos} \mid \mathcal{A} = a_1, Y = 1\}$: instances where $Y = 1$ and the sensitive attribute $\mathcal{A} = a_1$ (e.g., gender=male, race = majority).
- Samples belonging to class $Y = 0$ (e.g., gender=female).
  - $\mathcal{D}rep_{0,a_0} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{pos} \mid \mathcal{A} = a_0, Y = 0\}$: instances where $Y = 0$ and

the sensitive attribute $\mathcal{A} = a_0$.

– $\mathcal{D}rep_{0,a_1} = \{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{pos} \mid \mathcal{A} = a_1, Y = 0\}$: instances where $Y = 0$ and the sensitive attribute $\mathcal{A} = a_1$.

5.3.5 Estimating the level of bias

`SettSame` – Estimating the level of bias

We compute the level of bias in both the correctly classified instances and the misclassified instances with respect to the sensitive attribute $\mathcal{A} \in \{a_0, a_1\}$. This involves estimating how classifications are biased toward or against specific sensitive groups. The weighing scores add constraints to ensure neurons are selected fairly, preventing overcompensation toward a specific sensitive group.

• **Estimating the level of bias W_{neg} in misclassified data $\mathcal{D}rep_{neg}$**

We aim to estimate the level of bias in the misclassified data points produced by model $\mathcal{M}$ for different sub-groups, such as the deprived community a_0 and the favored community a_1. This bias quantification can then be used as a constraint when selecting the neuron weights corresponding to misclassification behaviors.

— ***Expected probability of bias-free misclassification***: If the model $\mathcal{M}$ were unbiased with respect to the dataset (i.e., $\mathcal{A}$ is statistically independent of misclassification), the expected probability $\mathcal{P}_{exp}$ of misclassifying data points with sensitive attribute $\mathcal{A} = a_0$ would be:

$$\mathcal{P}_{exp}(\mathcal{D}rep_{neg,a_0}) := \frac{|\{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep \mid \mathcal{A} = a_0\}|}{|\mathcal{D}rep|} \times \frac{|\mathcal{D}rep_{neg}|}{|\mathcal{D}rep|} \tag{5.1}$$

This term computes the expected proportion of deprived community members who would be misclassified by an unbiased model $\mathcal{M}$. Essentially, it calculates the fraction of deprived samples in the entire dataset and scales this by the overall misclassification rate.

— ***Observed probability of misclassification***: In reality, the observed probability $\mathcal{P}_{obs}$ of misclassifying data points with the sensitive attribute $\mathcal{A} = a_0$ may differ from the expected probability. The observed probability $\mathcal{P}_{obs}$ is defined as:

$$\mathcal{P}_{obs}(\mathcal{D}rep_{neg,a_0}) := \frac{|\mathcal{D}rep_{neg,a_0}|}{|\mathcal{D}rep|} \tag{5.2}$$

i.e., the proportion of misclassified instances in the dataset that belong to the deprived

community.

— ***Cost of misclassified instances***: If the observed probability is lower than the expected probability, $\mathcal{M}$ exhibits bias against the deprived community. To quantify this bias, we assign the following cost to the neuron weights when the input is the misclassified instances from the deprived community:

$$cost(\mathcal{D}rep_{neg,a_0}) := \frac{\mathcal{P}_{exp}(\mathcal{D}rep_{neg,a_0})}{\mathcal{P}_{obs}(\mathcal{D}rep_{neg,a_0})} \tag{5.3}$$

This cost represents the level of bias correction needed to ensure the deprived group is not unfairly penalized in the model's misclassifications. The cost scales inversely with the observed probability—if the deprived group is misclassified less frequently than expected, this cost will increase.

Similarly, the cost for misclassified instances from the favored community $\mathcal{A} = a_1$ is:

$$cost(\mathcal{D}rep_{neg,a_1}) := \frac{\mathcal{P}_{exp}(\mathcal{D}rep_{neg,a_1})}{\mathcal{P}_{obs}(\mathcal{D}rep_{neg,a_1})} \tag{5.4}$$

— ***Level of bias in misclassified instances***: The strength of bias in the misclassified instances $\mathcal{D}rep_{neg}$ can be quantified by comparing the bias costs between the deprived and favored groups. This is expressed as the ratio of the cost for the deprived community (Eq. 5.3) to the cost for the favored community (Eq. 5.4):

$$W_{neg} := \frac{cost(\mathcal{D}rep_{neg,a_1})}{cost(\mathcal{D}rep_{neg,a_0})} \tag{5.5}$$

The ratio W_{neg} quantifies the level of bias between the deprived and favored groups in the misclassified data points. If $W_{neg} > 1$, the model is more biased against the deprived group (i.e., their misclassification cost is higher than expected), and if $W_{neg} < 1$, the model is more biased against the favored group. W_{neg} can be used as a guiding metric when selecting neuron weights corresponding to deprived relative to favored group.

• **Estimating the level of bias W_{pos} in correctly classified data $\mathcal{D}rep_{pos}$**

Given the correctly classified data points, we estimate the level of bias represented by each sub-group, a_0 (deprived group) and a_1 (favored group), by calculating the cost per group following the same steps as in the previous analysis (Section 5.3.5).

— ***Cost for the deprived community a_0:*** The cost for the deprived community a_0 in the correctly classified set $\mathcal{D}rep_{pos}$ is calculated using the following formula:

$$cost(\mathcal{D}rep_{pos,a_0}) := \frac{\mathcal{P}_{exp}(\mathcal{D}rep_{pos,a_0})}{\mathcal{P}_{obs}(\mathcal{D}rep_{pos,a_0})} \tag{5.6}$$

This formula accounts for the proportion of the dataset that belongs to the deprived community and compares that to the proportion of correctly classified instances from the deprived community. This cost reflects how well the deprived group is represented in the correctly classified set relative to its overall presence in the dataset.

— ***Cost for the Favored Community*** a_1***:*** Similarly, the cost for the favored community a_1 in the correctly classified set $\mathcal{D}rep_{pos}$ is calculated as:

$$cost(\mathcal{D}rep_{pos,a_1}) := \frac{\mathcal{P}_{exp}(\mathcal{D}rep_{pos,a_1})}{\mathcal{P}_{obs}(\mathcal{D}rep_{pos,a_1})} \tag{5.7}$$

This formula mirrors the calculation for the deprived community, measuring the bias relative to the correctly classified instances for the favored group.

— ***Level of bias in correctly classified instances***: Once the costs for both the deprived and favored communities are computed, the strength of bias in the correctly classified instances $\mathcal{D}rep_{pos}$ is given by the ratio of the cost for the deprived community to the cost for the favored community:

$$W_{pos} := \frac{cost(\mathcal{D}rep_{pos,a_0})}{cost(\mathcal{D}rep_{pos,a_1})} \tag{5.8}$$

This ratio W_{pos} quantifies the relative bias between the deprived and favored groups among the correctly classified instances. A higher value of W_{pos} indicates a greater level of bias on the deprived community for the correct results. In this case: (i) $W_{pos} > 1$ indicates more biased toward favored group; this means the deprived community is likely to be least classified correctly by the model $\mathcal{M}$ than expected. (ii) $W_{pos} < 1$ indicates fairness on the favored group; this indicates the favored community is more likely to be correctly classified by the model $\mathcal{M}$ as expected, given their presence in the dataset.

This measure of bias in the correctly classified instances complements the bias measure computed in the misclassified instances $\mathcal{D}rep_{neg}$, providing a more holistic understanding of how the model behaves across different sensitive groups. The value of bias strength W_{pos} can then be used to guide the selection of neuron weights from the correctly classified inputs with respect to sensitive attribute $\mathcal{A} \in \{a_0, a_1\}$

Theorem 5.3.1. *To illustrate how the bias estimation works, we apply the above formulas to the simple repair dataset $\mathcal{D}rep$ shown in Table 5.1a. In this example, the sensitive attribute is gender and the class label is also gender; we consider 10 samples to illustrate the application*

Table 5.1 Examples for `SettSame` and `SettDiff`.

(a) `SettSame`

#	Y	$\mathcal{A}$	**Correct**	**Miss**
1	1	male(1)	Yes	No
2	1	male(1)	Yes	No
3	1	male(1)	Yes	No
4	0	female(0)	No	Yes
5	0	female(0)	No	Yes
6	1	male(1)	No	Yes
7	0	female(0)	Yes	No
8	0	female(0)	No	Yes
9	1	male(1)	Yes	No
10	0	female(0)	Yes	No

(b) `SettDiff`

#	Y	$\mathcal{A}$	**Correct**	**Miss**
1	1	white	Yes	No
2	0	white	No	Yes
3	0	black	Yes	No
4	1	white	No	Yes
5	0	black	No	Yes
6	1	white	No	Yes
7	0	white	Yes	No
8	1	black	Yes	No
9	1	white	Yes	No
10	0	black	Yes	No

of the formulas. So, according to the table, the sizes of the subsets of $\mathcal{D}rep$ are as follows:

- *Total samples.* $|\mathcal{D}rep| = 10$*;*
- *Female samples.* $|\{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep \mid \mathcal{A} = a_0\}| = 5$*;*
- *Misclassified samples.* $|\mathcal{D}rep_{neg}| = 4$*;*
- *Misclassified female samples.* $|\mathcal{D}rep_{neg,a_0}| = 3$*;*
- *Misclassified male samples.* $|\mathcal{D}rep_{neg,a_1}| = 1$*;*
- *Correctly classified samples.* $|\mathcal{D}rep_{pos}| = 6$*;*
- *Correctly classified female samples.* $|\mathcal{D}rep_{pos,a_0}| = 2$*;*
- *Correctly classified male samples.* $|\mathcal{D}rep_{pos,a_1}| = 4$.

Given these values, the levels of bias are computed as follows:

- *Bias in Misclassified Data:*

$$\mathcal{P}_{exp}(\mathcal{D}rep_{neg,a_0}) := \frac{5}{10} \times \frac{4}{10} = 0.2 \qquad \mathcal{P}_{obs}(\mathcal{D}rep_{neg,a_0}) := \frac{3}{10} = 0.3$$

$$cost(\mathcal{D}rep_{neg,a_0}) := \frac{0.2}{0.3} \approx 0.67 \qquad cost(\mathcal{D}rep_{neg,a_1}) := \frac{0.2}{0.1} = 2.0$$

$$W_{neg} := \frac{2.0}{0.67} \approx 2.99$$

- *Bias in Correctly Classified Data:*

$$cost(\mathcal{D}rep_{pos,a_1}) := \frac{0.3}{0.2} = 1.5 \qquad cost(\mathcal{D}rep_{pos,a_1}) := \frac{0.3}{0.4} = 0.75$$

$$W_{pos} := \frac{1.5}{0.75} = 2.0$$

`SettDiff` – Estimating the level of bias

This section outlines the procedure for estimating bias in `SettDiff`, where the sensitive attribute $\mathcal{A}$ differs from the class label Y. The bias estimation follows a similar approach as previously described but focuses on class-sensitive attribute combinations.

• Bias in the class label

We first estimate bias in predicting class labels $Y = 0$ and $Y = 1$, as follows:

— ***Expected probability of bias-free prediction for*** $Y = 0$:

$$\mathcal{P}_{exp}(Y = 0) := \frac{|\{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep \mid Y = 0\}|}{|\mathcal{D}rep|} \times \frac{|\mathcal{D}rep_{pos}|}{|\mathcal{D}rep|} \tag{5.9}$$

This term computes the expected proportion of correctly classified instances for $Y = 0$.

— ***Observed probability within the class*** $Y = 0$:

$$\mathcal{P}_{obs}(Y = 0) := \frac{|\{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep_{pos} \mid Y = 0\}|}{|\mathcal{D}rep|} \tag{5.10}$$

This reflects the actual classification rate of the model $\mathcal{M}$ for instances with $Y = 0$.

— ***Estimating the cost of classification for class label*** $Y = 0$***:***

$$cost(Y = 0) := \frac{\mathcal{P}_{exp}(Y = 0)}{\mathcal{P}_{obs}(Y = 0)} \tag{5.11}$$

This cost quantifies how much correction is needed for the model to achieve bias-free classification of $Y = 0$.

Similarly, the classification cost for the class label $Y = 1$ is calculated as:

$$cost(Y = 1) := \frac{\mathcal{P}_{exp}(Y = 1)}{\mathcal{P}_{obs}(Y = 1)} \tag{5.12}$$

— ***The level of bias in class label***

The strength of bias reflected in the two class labels can be quantified by comparing the bias costs between the $Y = 0$ and $Y = 1$: if $cost(Y = 0) > cost(Y = 1)$, the model

is biased towards $Y = 0$, otherwise, it is biased towards $Y = 1$. This information will guide the scoring function in prioritizing class labels to achieve equitable outcomes across class-sensitive attribute combinations.

• **Bias strength for deprived community**

We aim to estimate the level of bias for the subset of data where $Y = 0$ and the sensitive attribute is from deprived community, $\mathcal{A} = a_0$.

— ***Expected probability of deprived community*** a_0 ***in*** $Y = 0$:

$$\mathcal{P}_{exp}(\mathcal{D}rep_{0,a_0}) := \frac{|\{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep \mid \mathcal{A} = a_0\}|}{|\mathcal{D}rep|} \times \frac{|\{(\mathcal{X}, Y, \mathcal{A}) \in \mathcal{D}rep \mid Y = 0\}|}{|\mathcal{D}rep|} \tag{5.13}$$

— ***Observed probability of deprived community*** a_0 ***within the class*** $Y = 0$:

$$\mathcal{P}_{obs}(\mathcal{D}rep_{0,a_0}) := \frac{|\mathcal{D}rep_{0,a_0}|}{|\mathcal{D}rep|} \tag{5.14}$$

— ***Estimating the cost for deprived community in*** $Y = 0$:

$$cost(\mathcal{D}rep_{0,a_0}) := \frac{\mathcal{P}_{exp}(\mathcal{D}rep_{0,a_0})}{\mathcal{P}_{obs}(\mathcal{D}rep_{0,a_0})} \tag{5.15}$$

Similarly, the cost for the favored community a_1 in $Y = 0$ is calculated as:

$$cost(\mathcal{D}rep_{0,a_1}) := \frac{\mathcal{P}_{exp}(\mathcal{D}rep_{0,a_1})}{\mathcal{P}_{obs}(\mathcal{D}rep_{0,a_1})} \tag{5.16}$$

— ***Computing the level of bias in class*** $Y = 0$***:***

$$W_0 := \frac{cost(\mathcal{D}rep_{0,a_0})}{cost(\mathcal{D}rep_{0,a_1})} \tag{5.17}$$

This ratio W_0 quantifies the level of bias between the deprived and favored groups in the instance of dataset for class labeled category $Y = 0$.

• **Estimating the level of bias within class** $Y = 1$

The bias for $Y = 1$ is computed similarly to $Y = 0$, using the same methodology to assess bias between the deprived and favored communities within $Y = 1$.

— Cost for the deprived community a_0 in class $Y = 1$:

$$cost(\mathcal{D}rep_{1,a_0}) := \frac{\mathcal{P}_{exp}(\mathcal{D}rep_{1,a_0})}{\mathcal{P}_{obs}(\mathcal{D}rep_{1,a_0})}$$

and similarly for favored community as:

$$cost(\mathcal{D}rep_{1,a_1}) := \frac{\mathcal{P}_{exp}(\mathcal{D}rep_{1,a_1})}{\mathcal{P}_{obs}(\mathcal{D}rep_{1,a_1})}$$

— ***Computing the level of bias in*** $Y = 1$***:***

$$W_1 := \frac{cost(\mathcal{D}rep_{1,a_0})}{cost(\mathcal{D}rep_{1,a_1})} \tag{5.18}$$

This framework estimates the bias strength for both $Y = 0$ and $Y = 1$, offering a comprehensive view of how the model behaves with respect to different sensitive attributes in each class. These bias measures can then guide the selection of neuron weights to be adjusted in the repair process to ensure equitable treatment of both deprived and favored communities.

Theorem 5.3.2. *Consider the scenario in Table 5.1b, where we are predicting gender* ($Y \in \{0, 1\}$*, where* $1 =$ *male,* $0 =$ *female), and the sensitive attribute* $\mathcal{A} \in \{a_0, a_1\}$ *represents race* ($s_0 =$ *black,* $s_1 =$ *white). In this case,* Y *and* $\mathcal{A}$ *are not the same, and gender prediction should be independent of race. This fits the* `SettDiff` *condition.*

We compute the Bias in Class Label as follows:

- *For* $Y = 0$

$$cost(Y = 0) := \frac{\frac{5}{10} \times \frac{5}{10}}{\frac{2}{10}} = \frac{0.25}{0.2} = 1.25$$

- *For* $Y = 1$

$$cost(Y = 1) := \frac{\frac{5}{10} \times \frac{5}{10}}{\frac{3}{10}} = \frac{0.25}{0.3} \approx 0.83$$

- *Bias in class* $Y = 0$

$$\mathcal{P}_{exp}(\mathcal{D}rep_{0,black}) := \frac{4}{10} \times \frac{5}{10} = 0.20 \qquad \mathcal{P}_{exp}(\mathcal{D}rep_{0,white}) := \frac{6}{10} \times \frac{5}{10} = 0.30$$

$$\mathcal{P}_{obs}(\mathcal{D}rep_{0,black}) := \frac{2}{10} = 0.2 \qquad \mathcal{P}_{obs}(\mathcal{D}rep_{0,white}) := \frac{1}{10} = 0.1$$

$$W_0 := \frac{\frac{0.20}{0.20}}{\frac{0.30}{0.1}} = \frac{1.0}{3.0} \approx 0.333$$

- *Bias in class* $Y = 1$

$$\mathcal{P}_{exp}(\mathcal{D}rep_{1,black}) := \frac{1}{10} \times \frac{5}{10} = 0.05 \qquad \mathcal{P}_{exp}(\mathcal{D}rep_{1,white}) := \frac{4}{10} \times \frac{5}{10} = 0.2$$

$$\mathcal{P}_{obs}(\mathcal{D}rep_{1,black}) := \frac{1}{10} = 0.1 \qquad \mathcal{P}_{obs}(\mathcal{D}rep_{1,white}) := \frac{2}{10} = 0.2$$

$$W_1 := \frac{\frac{0.05}{0.1}}{\frac{0.2}{0.2}} = 0.5$$

5.3.6 Neuron Analysis with respect to data samples

As depicted in Figure 5.2, we conduct neuron analysis with respect to the data samples that differ by the sensitive attribute $\mathcal{A} \in \{a_0, a_1\}$. The goal is to understand how the weights of

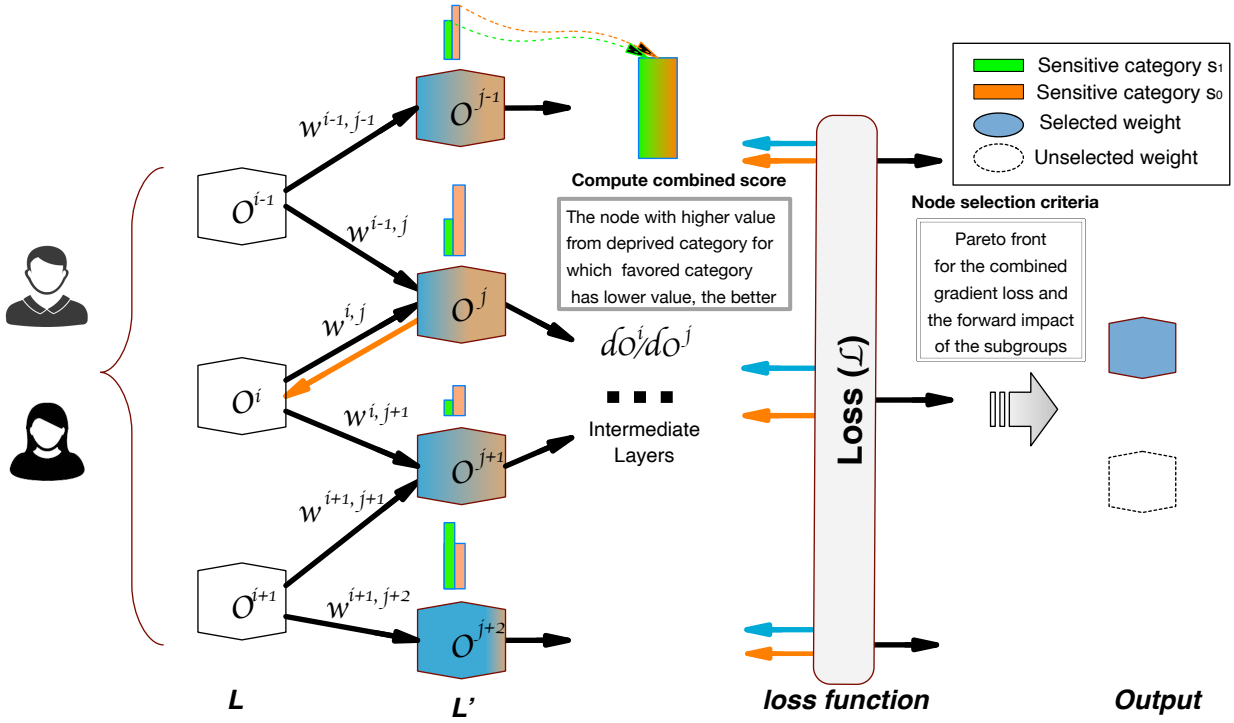


Figure 5.2 Propose `FairFL` fault localization technique. We first compute the individual gradient loss and forward impact of each individual weight for sub-groups within a selected layer. Based on the Pareto front, we then select the weights from the group with combined higher gradient and forward impact. In the figure, black arrows represent forward pass, whereas blue and orange arrows are backpropagation. Black and blue arrows combined compute the forward impact, whereas the orange arrow denotes the gradient loss. The gradient loss and the forward impact combined evaluate the likelihood of $w^{i,j}$ being localized by our technique.

individual neurons contribute to the model's incorrect behavior (i.e., unfair predictions). To achieve this, we calculate both the *forward impact* and *gradient loss* of each neuron weight in a given layer of the neural network.

Gradient Loss

The gradient loss measures the responsibility that a particular neural weight has for the misclassification within a specific sensitive attribute group (e.g., the deprived group). A higher gradient loss indicates that the weight is more strongly associated with causing misclassification errors. The intuition is that, by fixing the misclassification with respect to such neurons, we have the potential of improving the unfair treatment for the different sub-groups of sensitive attribute.

Given an input sample from a deprived subset (e.g., $\mathcal{D}rep_{neg,a_0}$), we compute the gradient loss with respect to each weight w^l in layer l (typically, the last layer is used in this study) of the neural network by taking the gradient of the loss function $\mathcal{L}$ with respect to these weights. We introduce the following notation:

- $\mathcal{L}_{neg,a_0} = \mathcal{L}(\hat{Y}_{neg,a_0}, Y_{neg,a_0})$: The loss (e.g., cross-entropy) between the predicted output $\hat{Y}_{neg,a_0}$ and the true label Y_{neg,a_0}, for the input sample from the deprived community $\mathcal{D}rep_{neg,a_0}$.
- $w^l \in W^l$: The individual weights of the layer l of the neural network $\mathcal{M}$.
- z^l : The activations (pre-activation outputs) of the layer l before applying the output activation function (e.g., softmax).

  The gradient loss with respect to the neuron weight w^l in layer l for a given input sample $\mathcal{D}rep_{neg,a_0}$ is calculated as:

$$grad\_loss_{w^l(\mathcal{D}rep_{neg,a_0})} = \frac{\delta \mathcal{L}_{neg,a_0}}{\delta w^l} = \frac{\delta \mathcal{L}_{neg,a_0}}{\delta z^l} \cdot \frac{\delta z^l}{\delta w^l}$$

  where:
  - $\frac{\delta \mathcal{L}_{neg,a_0}}{\delta w^l}$ is the gradient of the loss function with respect to the pre-activation outputs z^l. This quantifies how changes in the activations of layer l affect the loss for the missclasified instance of deprived sample.
  - $\frac{\delta z^l}{\delta w^l}$ is the gradient of the pre-activation outputs with respect to the weights in layer l. This quantifies how changes in the weights affect the activations in layer l.

  A high value of the gradient loss indicates that the weight w^l is contributing significantly to the incorrect behavior (i.e., misclassifications) of the input from the deprived group.

Adjusting such weights can help reduce unfair misclassifications, thereby improving fairness for the sensitive attribute group.

Similarly, the gradient loss of each neuron weights w^l in layer l is computed using the same methodology for other subsets of data. For an input sample from the favored community $\mathcal{D}rep_{neg,a_1}$, the gradient loss is calculated as:

$$grad\_loss_{w^l(\mathcal{D}rep_{neg,a_1})} = \frac{\delta \mathcal{L}_{neg,a_1}}{\delta z^l} \cdot \frac{\delta z^l}{\delta w^l} = \frac{\delta \mathcal{L}_{neg,a_1}}{\delta w^l}$$

Forward impact analysis

The forward impact measures the influence of a neural weight on the final classification decision. This metric evaluates how strongly each weight affects the prediction outcome, regardless of whether the prediction is correct or incorrect. Similar to gradient loss, forward impact is calculated for each neural weight with respect to different sensitive groups (i.e., $\mathcal{D}rep_{a_0}$, $\mathcal{D}rep_{a_1}$).

Estimating the forward impact of a neural weight w^l on the final output is more complex than computing the gradient loss. Neural weights that strongly influence the activation value of the connected output neuron, where the output neuron subsequently influences the final classification result, are considered to have a significant impact on the model's predictions. We compute forward impact as follows:

1. We begin by estimating the influence of a neural weight on the activation of the output neuron, denoted as $o^l w^l$. This is done by multiplying the neural weight w^l by the average activation value z_μ^{l-1} of the neuron it is connected to in the previous layer, resulting in:

   $$o^l w^l = w^l * z_\mu^{l-1} + b^l$$

   where b^l is the bias vector term on the output neuron.

   The computed influence is then normalized by dividing it by the total influence of all neural weights connected to the same output neuron:

   $$Normalized\ influence = \frac{o^l w^l}{\sum_i^{|L|} o^l w^l}$$

   where $|L|$ is the total number of weights connected to the output neuron.

   This normalization step ensures that the forward impact is measured as a proportion of the total influence on the output neuron.
2. Next, we calculate the influence of the output neuron's activation on the final output

$\mathcal{O}$. This is done by computing the gradient of the final output $\mathcal{O}$ with respect to the activation of the output neuron o'_j.

$$\textit{Influence on final output} = \frac{\delta \mathcal{O}}{\delta o'_j}$$

This gradient captures how much the final output changes in response to changes in the activation of the output neuron.

Finally, we compute the forward impact ($fwd\_impact_{w^l}$) of a neural weight w^l as:

$$fwd\_impact_{w^l} = \frac{o^l w^l}{\sum_i^{|L|} o^l w^l} \times \frac{\delta \mathcal{O}}{\delta o'_j} \tag{5.19}$$

This provides a comprehensive measure of the impact the weight has on the final classification decision.

As with the gradient loss, we estimate the forward impact of the neural weights with respect to different sub-groups of sensitive attributes. For example:

- $fwd\_impact_{w^l(\mathcal{D}rep_{neg,a_0})}$: Forward impact of neuron weight w^l when the input is the misclassified samples from the deprived community $\mathcal{D}rep_{neg,a_0}$.
- $fwd\_impact_{w^l(\mathcal{D}rep_{pos,a_1})}$: Forward impact of neuron weight w^l when the input is the correctly classified samples from the favored community $\mathcal{D}rep_{pos,a_1}$.

5.3.7 Scoring of neuron weights

We aggregate the values of the forward impact independently of the gradient loss based on a scoring function that identifies faulty neurons contributing to the biased behavior. The information from both the gradient loss and forward impact of each neuron weight allows us to treat these as two competing objectives to optimize. Neurons with higher bias impact on the deprived community are prioritized for repair. The bias strength scores (See Sections 5.3.5 and 5.3.5) help control how much each neuron is selected for adjustment, ensuring that the selected weights is proportional to the bias reflected in $\mathcal{M}$.

As pointed out in Section 5.3.4, the dual categorization of the data are based on the version of fault-location technique: `SettSame` when $Y = \mathcal{A}$, and `SettDiff` when $Y \neq \mathcal{A}$. Similarly, the scoring function differs slightly depending on the version of the fault-location technique used.

`SettSame` (fair-aware fault localization when $Y = \mathcal{A}$)

The scoring function focuses on selecting neuron weights contributing more to misclassifications from the deprived community relative to those contributing to correct classifications from the favored community. The goal is to prioritize the neuron weights that disproportionately impact deprived groups, relative to the favored groups. Thus, for each weight w^l, we compute:

$$score_{w^l} = W_{neg} \cdot \frac{gf(w^l \mid \mathcal{D}rep_{neg,a_0})}{1 + gf(w^l \mid \mathcal{D}rep_{neg,a_1})} - W_{pos} \cdot \frac{gf(w^l \mid \mathcal{D}rep_{pos,a_0})}{1 + gf(w^l \mid \mathcal{D}rep_{pos,a_1})}, \tag{5.20}$$

where:

- w^l is the specific weight being scored
- W_{neg} and W_{pos} are the level of bias in misclassified and correctly classified instances, computed based on Eqs. 5.5 and 5.8,
- gf is either the measure of gradient of loss or forward impact of weight w^l with respect to the input samples.

The numerator in these equations captures the gradient loss and forward impact on the deprived community, while the denominator captures the impact on the favored community. Neuron weights with higher gradient loss and forward impact on the deprived community are maximized hence are more likely to be selected for repair, while neurons with high impact on the favored community are minimize hence scored low.

The pseudo-code of the `FairFL` method, given $Y = \mathcal{A}$, is presented in Algorithm 2. The algorithm works as follows.

1. It starts by randomly sampling the same number of positive inputs as negative ones since, typically, a fully trained model has far more positive inputs than negative ones (Line 2).
2. It computes the bias cost (both positive and negative) for each sensitive group (Lines 5-6).
3. For each neuron weight w^l, it computes the gradient loss with respect to the deprived and favored groups for both misclassified and correctly classified instances, according to procedure in Section 5.3.6. This is done by calling the function *Compute_Gradient_Loss* (Lines 11-14).
4. Similarly, for each neuron weight w^l, it computes the forward impact with respect to the deprived and favored groups for both misclassified and correctly classified instances, using the *Compute_Forward_Impact* function (Lines 16-19).

Algorithm 2 FairFL fault localization when $Y = \mathcal{A}$

INPUT: A DNN model to be repaired $\mathcal{M}$; set of inputs correctly and incorrectly classified data instance $\{\mathcal{D}rep_{pos}, \mathcal{D}rep_{neg}\} \in \mathcal{D}rep$; the sensitive attributes $\mathcal{A} \in \{a_0, a_1\}$; the target layer l.
OUTPUT: a set of neural weights to target for repair, $\mathbf{w_r}$

1: **procedure** FairFL
2: $\mathcal{D}rep_{pos} \leftarrow$ RandomSample($\mathcal{D}rep_{pos}$, sizeof=$|\mathcal{D}rep_{neg}|$)
3: $\{\mathcal{D}rep_{pos,a_0}, \mathcal{D}rep_{pos,a_1}\} \leftarrow$ Inputs belonging to deprived and favored group in $\mathcal{D}rep_{pos}$
4: $\{\mathcal{D}rep_{neg,a_0}, \mathcal{D}rep_{neg,a_1}\} \leftarrow$ Inputs belonging to deprived and favored group in $\mathcal{D}rep_{neg}$
5: $W_{neg} \leftarrow$ compute_cost_of_bias($\mathcal{D}rep_{neg}, a_0, a_1$)
6: $W_{pos} \leftarrow$ compute_cost_of_bias($\mathcal{D}rep_{pos}, a_0, a_1$)
7: *scoredWeights* $\leftarrow$ []
8: $W^l \leftarrow$ Set of weights in the target layer l
9: $\mathcal{L} \leftarrow$ a loss function used in $\mathcal{M}$
10: **for** $w^l \in W^l$ **do**
11: $grad\_loss_{w^l(\mathcal{D}rep_{neg,a_0})} \leftarrow Compute\_Gradient\_Loss(w^l, \mathcal{M}, \mathcal{D}rep_{neg,a_0}, \mathcal{L})$
12: $grad\_loss_{w^l(\mathcal{D}rep_{neg,a_1})} \leftarrow Compute\_Gradient\_Loss(w^l, \mathcal{M}, \mathcal{D}rep_{neg,a_1}, \mathcal{L})$
13: $grad\_loss_{w^l(\mathcal{D}rep_{pos,a_0})} \leftarrow Compute\_Gradient\_Loss(w^l, \mathcal{M}, \mathcal{D}rep_{pos,a_0}, \mathcal{L})$
14: $grad\_loss_{w^l(\mathcal{D}rep_{pos,a_1})} \leftarrow Compute\_Gradient\_Loss(w^l, \mathcal{M}, \mathcal{D}rep_{pos,a_1}, \mathcal{L})$
15: $grad\_lossScore_{w^l} \leftarrow W_{neg} * \frac{grad\_loss_{w^l(\mathcal{D}rep_{neg,a_0})}}{1+grad\_loss_{w^l(\mathcal{D}rep_{neg,a_1})}} - W_{pos} * \frac{grad\_loss_{w^l(\mathcal{D}rep_{pos,a_0})}}{1+grad\_loss_{w^l(\mathcal{D}rep_{pos,a_1})}}$
16: $fwd\_impact_{w^l(\mathcal{D}rep_{neg,a_0})} \leftarrow Compute\_Forward\_Impact(w^l, \mathcal{M}, \mathcal{D}rep_{neg,a_0})$
17: $fwd\_impact_{w^l(\mathcal{D}rep_{neg,a_1})} \leftarrow Compute\_Forward\_Impact(w^l, \mathcal{M}, \mathcal{D}rep_{neg,a_1})$
18: $fwd\_impact_{w^l(\mathcal{D}rep_{pos,a_0})} \leftarrow Compute\_Forward\_Impact(w^l, \mathcal{M}, \mathcal{D}rep_{pos,a_0})$
19: $fwd\_impact_{w^l(\mathcal{D}rep_{pos,a_1})} \leftarrow Compute\_Forward\_Impact(w^l, \mathcal{M}, \mathcal{D}rep_{pos,a_1})$
20: $fwd\_impact_{w^l} \leftarrow W_{neg} * \frac{fwd\_impact_{w^l(\mathcal{D}rep_{neg,a_0})}}{1+fwd\_impact_{w^l(\mathcal{D}rep_{neg,a_1})}} - W_{pos} * \frac{fwd\_impact_{w^l(\mathcal{D}rep_{pos,a_1})}}{1+fwd\_impact_{w^l(\mathcal{D}rep_{pos,a_1})}}$
21: **Add tuple** $(w^l, grad\_loss, fwd\_impact)$ **to** *scoredWeights*
22: **end for**
23: $\mathbf{w_r} \leftarrow$ *ExtractParetoFront(scoredWeights)*
24: **Return** $\mathbf{w_r}$
25: **end procedure**

5. The values of the gradient loss and forward impact are aggregated to create a combined score for each neuron weight in Line 15 and Line 20, respectively. This aggregation gives higher priority to neuron weights that contribute more to negative inputs, with a specific focus on the deprived community relative to the favored group.
6. The algorithm identifies the set of neuron weights $\mathbf{w_r}$ that constitute the Pareto front based on the combined scores for gradient loss and forward impact (Line 23). The Pareto front consists of neuron weights for which no other weight has both a greater gradient loss and greater forward impact simultaneously. These neurons are the most

impactful for repair, as improving them will likely lead to the greatest reduction in bias without negatively affecting model accuracy.

`SettDiff` (fair-aware fault localization when $Y \neq \mathcal{A}$)

Similarly to `SettSame`, W_0 measures bias in the underrepresented class label (Eq. 5.17), and W_1 measures bias in the more represented class label (Eq. 5.18). The neuron scoring function prioritizes underrepresented groups based on their class labels and sensitive attributes. The scoring function for `SettDiff` balances the need to address bias in the underrepresented class while maintaining fairness in the more represented class. The goal is to prioritize neuron weights contributing to biased outcomes for underrepresented classes while ensuring that the adjustment does not unfairly penalize the more represented class.

Algorithm 3 presents the pseudo-code of the `FairFL` method given $Y \neq \mathcal{A}$.

1. It starts by randomly sampling equal numbers of instances from the less represented class (e.g., $Y = 0$) and the more represented class (e.g., $Y = 1$), ensuring balanced input data for the scoring process (Line 2). The algorithm assumes that the cost of misclassification for $Y = 0$ is greater than that for $Y = 1$, meaning that the underrepresented group ($Y = 0$) faces more bias.
2. It computes the bias cost (both $Y = 1$ and $Y = 0$) for each sensitive group (Lines 5-6).
3. For each neuron weight w, it computes the gradient loss with respect to the deprived and favored groups for classes, according to procedure in Section 5.3.6. This is done by calling the function *Compute_Gradient_Loss* (Lines 11-14).
4. Similarly, for each neuron weight w, it computes the forward impact with respect to the deprived and favored groups for both classes, using the *Compute_Forward_Impact* function (Lines 16-19).
5. The values of the gradient loss and forward impact are aggregated to create a combined score for each neuron weight in Lines 15 and 20, respectively. This aggregation gives higher priority to neuron weights that contribute more to class-sensitive input combination, with a specific focus on the deprived community relative to the favored group.
    - The scoring function prioritizes the underrepresented class $Y = 0$ (the term on the left of the equations, in Line 15 and Line 20). The numerators in the scoring functions capture the impact of neuron weights on the deprived community, while the denominators capture their impact on the favored community. The scoring maximizes the effect on underrepresented groups (with higher bias, as measured by W_0) and minimizes the impact on more represented groups (measured by W_1).

Algorithm 3 `FairFLRep` fault localization when $Y \neq \mathcal{A}$

INPUT: A DNN model to be repaired $\mathcal{M}$; set of inputs correctly and incorrectly classified data instance $\{\mathcal{D}rep_{pos}, \mathcal{D}rep_{neg}\} \in \mathcal{D}rep$; the sensitive attributes $\mathcal{A} \in \{a_0, a_1\}$; the target layer l.
OUTPUT: a set of neural weights to target for repair, $\mathbf{w_r}$

1: **function** FAIRFL
2: $\mathcal{D}rep_{pos} \leftarrow$ RandomSample($\mathcal{D}rep_{pos}$, sizeof=$|\mathcal{D}rep_{neg}|$)
3: $\{\mathcal{D}rep_{1,a_0}, \mathcal{D}rep_{1,a_1}\} \leftarrow$ Inputs belonging to deprived and favored group, respectively, in $\mathcal{D}rep_1$
4: $\{\mathcal{D}rep_{0,a_0}, \mathcal{D}rep_{0,a_1}\} \leftarrow$ Inputs belonging to deprived and favored group, respectively, in $\mathcal{D}rep_0$
5: $W_0 \leftarrow$ compute_cost_of_bias($\mathcal{D}rep_0, a_0, a_1$)
6: $W_1 \leftarrow$ compute_cost_of_bias($\mathcal{D}rep_1, a_0, a_1$)
7: *scoredWeights* $\leftarrow []$
8: $W^l \leftarrow$ Set of weights in the target layer l
9: $\mathcal{L} \leftarrow$ a loss function used in $\mathcal{M}$
10: **for** $w^l \in W^l$ **do**
11: $grad\_loss_{w^l(\mathcal{D}rep_{0,a_0})} \leftarrow Compute\_Gradient\_Loss(w^l, \mathcal{M}, \mathcal{D}rep_{0,a_0}, \mathcal{L})$
12: $grad\_loss_{w^l(\mathcal{D}rep_{0,a_1})} \leftarrow Compute\_Gradient\_Loss(w^l, \mathcal{M}, \mathcal{D}rep_{0,a_1}, \mathcal{L})$
13: $grad\_loss_{w^l(\mathcal{D}rep_{1,a_0})} \leftarrow Compute\_Gradient\_Loss(w^l, \mathcal{M}, \mathcal{D}rep_{1,a_0}, \mathcal{L})$
14: $grad\_loss_{w^l(\mathcal{D}rep_{1,a_1})} \leftarrow Compute\_Gradient\_Loss(w^l, \mathcal{M}, \mathcal{D}rep_{1,a_1}, \mathcal{L})$
15: $grad\_lossScore_{w^l} \leftarrow W_0 * \frac{grad\_loss_{w^l(\mathcal{D}rep_{0,a_0})}}{1+grad\_loss_{w^l(\mathcal{D}rep_{0,a_1})}} - W_1 * \frac{grad\_loss_{w^l(\mathcal{D}rep_{1,a_0})}}{1+grad\_loss_{w^l(\mathcal{D}rep_{1,a_1})}}$
16: $fwd\_impact_{w^l(\mathcal{D}rep_{0,a_0})} \leftarrow Compute\_Forward\_Impact(w^l, \mathcal{M}, \mathcal{D}rep_{0,a_0})$
17: $fwd\_impact_{w^l(\mathcal{D}rep_{0,a_1})} \leftarrow Compute\_Forward\_Impact(w^l, \mathcal{M}, \mathcal{D}rep_{0,a_1})$
18: $fwd\_impact_{w^l(\mathcal{D}rep_{1,a_0})} \leftarrow Compute\_Forward\_Impact(w^l, \mathcal{M}, \mathcal{D}rep_{1,a_0})$
19: $fwd\_impact_{w^l(\mathcal{D}rep_{1,a_1})} \leftarrow Compute\_Forward\_Impact(w^l, \mathcal{M}, \mathcal{D}rep_{1,a_1})$
20: $fwd\_impactScore_{w^l} \leftarrow W_0 * \frac{fwd\_impact_{w^l(\mathcal{D}rep_{0,a_0})}}{1+fwd\_impact_{w^l(\mathcal{D}rep_{0,a_1})}} - W_1 * \frac{fwd\_impact_{w^l(\mathcal{D}rep_{1,a_0})}}{1+fwd\_impact_{w^l(\mathcal{D}rep_{1,a_1})}}$
21: **Add tuple** $(w^l, grad\_loss, fwd\_impact)$ **to** *scoredWeights*
22: **end for**
23: $\mathbf{w_r} \leftarrow$ *ExtractParetoFront(scoredWeights)*
24: **Return** $\mathbf{w_r}$
25: **end function**

- If the cost for $Y = 1$ is greater, then the same equations apply but reverse the roles of W_0 and W_1. Similarly, if the deprived community is a_1, the terms for $\mathcal{A} = a_1$ appear in the numerator, and a_0 in the denominator.

6. The algorithm identifies the set of neuron weights $\mathbf{w_r}$ that form the Pareto front based on the combined scores for gradient loss and forward impact (Line 23). The Pareto front consists of neuron weights for which no other weight has both a greater gradient loss and greater forward impact simultaneously. These neurons are the most impactful

for repair, as improving them will likely lead to the greatest reduction in bias without negatively affecting model accuracy.

5.3.8 Repair of the fairness in DNNs

In the repair phase of the `FairFLRep` method, the objective is to modify the suspicious neuron weights $\mathbf{w_r}$ identified during the fault localization step to reduce bias and improve fairness in the DNN $\mathcal{M}$. This process is designed to adjust the model without causing a significant drop in overall accuracy. The method utilizes Particle Swarm Optimization (PSO) to explore possible weight changes that enhance fairness while correcting misclassifications.

Initialization Strategy

The repair process starts by initializing a population of candidate solutions, where each solution represents an alternative configuration of neuron weights. To avoid drastic changes that might degrade model performance, the search focuses on regions of the weight space near the original values.

For each candidate solution, the initial value of each neuron weight is sampled from a Gaussian distribution. The mean and standard deviation of the distribution are computed using the existing weights within the same layer. This ensures that the search begins in a localized region of the parameter space, close to the original weight values, allowing for gradual adjustments.

Particle Swarm Optimization (PSO)

PSO is a population-based optimization technique inspired by the social behavior of birds flocking or fish schooling. Each individual in the population (called a "particle") represents a potential solution, which in this context is a set of neuron weights. The particles explore the search space by adjusting their positions based on their own experience and the experience of neighboring particles. The goal of PSO in this context is to find an optimal configuration of neuron weights that improves fairness in the model while maintaining or improving classification accuracy.

Each particle's position (representing a weight configuration) is updated based on two factors:

1. *pbest* - (personal best): The best position (i.e., the set of weights in the neural network) that an individual particle has found so far during the optimization process.
2. *gbest* - (global best): The best position found by any particle in the swarm. This serves

as a reference point for all particles.

The velocity of each particle is adjusted in each iteration based on its distance from *pbest* and *gbest*, ensuring that particles move toward a combination of their own best experience and the swarm's collective best experience. The position of each particle is updated as follows:

$$pbest_i = \begin{cases} x_i^{t+1}, & \text{if Fitness}(x_i^{t+1}, \mathcal{F}) \leq \text{Fitness}(pbest_i, \mathcal{F}) \\ pbest_i, & \text{otherwise} \end{cases} \tag{5.21}$$

where:

- x_i^t is the current position of particle i,
- x_i^{t+1} is the updated position of particle i based on its velocity,
- $pbest_i$ is the personal best position of particle i,
- $\mathcal{F}$ is the fairness metric e.g., *EOD*.

The global best position *gbest* is updated if any particle finds a better solution than the current *gbest*:

$$gbest = \begin{cases} x_i^{t+1}, & \text{if Fitness}(x_i^{t+1}, \mathcal{F}) \leq \text{Fitness}(gbest, \mathcal{F}) \\ gbest, & \text{otherwise} \end{cases} \tag{5.22}$$

Fitness Function

The fitness function plays a crucial role in guiding the PSO optimization process. It evaluates how well each candidate solution (set of neuron weights) meets the specified fairness criteria while also ensuring that the model's classification performance remains acceptable.

The fitness function is defined in terms of a fairness metric, $\mathcal{F}$, such as *SPD*, *EOD*, *FPR* or *DI*. The fairness definition is application-specific, and the user can specify which fairness metric to minimize.

The computation of the fitness for a given particle p works as follows. We first build the corresponding model $\mathcal{M}_p$ by updating the neuron weights $\mathbf{w_r}$ with p. Then, we evaluate the repair dataset $\mathcal{D}rep$ using $\mathcal{M}_p$. Based on the results, we identify which is the deprived community (and the favored community as a consequence) of the sensitive attribute; indeed, it could be that, due to the repair action, the deprived community changes w.r.t. that of the original model. Given this information, we compute the fitness value using the fairness metric $\mathcal{F}$.

Note that the fitness function aims at improving the fairness metric, and it does not explicitly

consider overall accuracy. Therefore, it could happen that some previously correct input is misclassified by the modified model. In the experiments (RQ2), we will assess how much accuracy is affected by repair.

Stopping criterion

The search continues until a predefined stopping criterion is met (e.g., a maximum number of iterations or a satisfactory reduction in unfairness).

5.4 Experiment design

In this section, we introduce the design of the experiments we conducted to assess `FairFLRep`. The approach has been implemented in Keras [192] with a TensorFlow back-end [193]. The code and all the experimental results are available at [194].

5.4.1 Research questions (RQs)

We will evaluate `FairFLRep` using these five research questions (RQs).

RQ1: *Can `FairFLRep` effectively improve the fairness of the DNN?*
This question is critical to validating the method's core objective: mitigating bias in DNNs. We aim to determine whether `FairFLRep` can accurately detect and reduce bias, leading to enhanced fairness across various sensitive groups.

RQ2: *Can `FairFLRep` effectively improve the fairness without harming the accuracy of the DNN?*
While fairness is a key concern, maintaining high predictive performance is crucial in real-world deployments. We aim to assess whether `FairFLRep` can reduce bias while preserving the model's classification accuracy, ensuring that the method is practical for application in real-world systems.

RQ3: *Which fairness metric is better for repairing the fairness of DNN?*
Fairness can be measured using various metrics such as *EOD*, *SPD*, *DI*, and *FPR*, each capturing different aspects of fairness. Optimizing one metric may impact others and model's predictive performance. Understanding which fairness metric leads to the most effective bias mitigation is crucial for guiding the repair process and ensuring alignment with specific fairness concerns in different applications. This RQ aims to identify the most suitable metric for fairness optimization, as well as to explore how these metrics relate with one another, to help uncover potential correlations or trade-offs.

RQ4: *Is repairing only the last layer more effective than repairing other layers?*

One key question in model repair is whether fairness can be improved effectively by modifying only the last layer (fully connected layer) instead of adjusting other layers in the network, which is a more costly operation, due to the higher number of weights. This RQ aims to assess whether modifications to the last layer alone are sufficient for bias mitigation, or if fairness-aware repair must target deeper layers (e.g., convolutional layers in CNNs). Understanding this trade-off is crucial for minimizing computational overhead while maximizing fairness improvements.

RQ5: *How efficient is* ***FairFLRep****?*

This question examines the computational efficiency of FairFLRep to ensure it can repair fairness quickly and efficiently. This is especially important for large-scale or time-sensitive deployments. By addressing this RQ, we can evaluate whether FairFLRep can scale to large datasets and complex models without imposing significant computational overhead.

5.4.2 Benchmark datasets and models

Image datasets: We use four well-known image classification datasets for fairness testing including FairFace [175], UTKFace [176], CelebA [178], and Labeled Faces in the Wild (LFW) [177]. The images exhibit a broad range of variations in pose, facial expression, illumination, occlusion, resolution, and more. All four datasets provide annotations for binary gender. FairFace and UTKFace also include annotations for race (*black*, *white*, *Asian*, *Middle Eastern*, and *Indian*); for these datasets, we only select samples from the *black* and *white* categories. The CelebA dataset includes a predefined binary "Young" attribute to represent age, where "1" signifies "*young*" and "0" signifies *"not-young"* (or *"old"*), as originally assigned by the dataset authors.

Tabular datasets: We also include four real-world well-known tabular datasets for fairness assessment: Student Performance (Student) [179], COMPAS Recidivism [180], Adult Income (Adult) [181], and Medical Expenditure Panel Survey (MEPS) [182]. The Student dataset contains academic performance data from Portuguese secondary schools, with final grades as the target variable. COMPAS includes over $10,000$ criminal defendant records used for recidivism risk assessment, known for racial bias in false positive rates [159]. The Adult Income dataset is used for income classification, with protected attributes including *gender*, *race*, and *age*. MEPS is a healthcare dataset for predicting medical expenditure, with *gender* as the primary protected attribute.

As explained in Section 5.3.3, the fault localization of FairFLRep (i.e., FairFL) differs depending on whether the sensitive attribute is the same as attribute under classification (SettSame),

or the sensitive attribute and the class label are different (`SettDiff`). We experiment both cases as follows:

`SettSame`: we use all image datasets `UTKFace`, `FairFace`, `LFW`, and `CelebA`, using *gender* as attribute under classification, and also as sensitive attribute. We do not use tabular datasets, as they are not applicable in this setting;

`SettDiff`: for image datasets, we use datasets `UTKFace`, `FairFace`, and `CelebA` (`LFW` is omitted due to lack of race or age group labels), using *gender* as attribute under classification for all of them; the sensitive attribute is race (*black* or *white*) for the `UTKFace` and `FairFace`, and *age* group (*young* or *old*) for the `CelebA` dataset. For tabular datasets, we use all of them (i.e., `Student`, `COMPAS`, `Adult`, and `MEPS`). The attribute under classification is: *exam result* for `Student` (*passed* or *failed*), *reoffending* for `COMPAS` (*yes* or *no*), *high income* for `Adult` (*yes* or *no*), and *utilize* for `MEPS` (*yes* or *no*). We experiment with two sensitive attributes: *gender* for all datasets, and *race* for `Adult`, `COMPAS`, and `MEPS`.

Models. We experiment with two DNN models of different architectures and complexities:

Image datasets: For the image datasets, we experiment with two DNN models of different architectures and complexities. 1. `VGG16`: A CNN composed of a series of convolutional and pooling layers, followed by three dense layers (fully connected layers). 2. `LeNet-5`: A relatively small network with seven layers, consisting of convolutional, pooling, and fully connected layers, making it a foundational model for deep learning.
Each model is trained separately on the four datasets, resulting in a total of eight models. These models are then used as inputs to evaluate the compared approaches.

Tabular datasets: For tabular datasets, we train a 1D CNN to process structured data effectively. Instead of traditional fully connected layers, we leverage Conv1D layers to extract feature patterns from tabular inputs, treating each sample as a 1D sequence of numerical features. This architecture enables efficient feature extraction from tabular datasets, leveraging spatial dependencies among features while maintaining computational efficiency.

5.4.3 Compared approaches

`FairFLRep`

It is our proposed fairness-aware method where both the fault localization and repair phases are explicitly optimized with fairness in mind. The method takes as input a fairness metric $\mathcal{F}$ that needs to be addressed (*EOD*, *SPD*, *DI*, or *FPR*); we indicate as $\texttt{FairFLRep}_{\mathcal{F}}$ the

approach initialized with fairness metric $\mathcal{F}$. As explained in Section 5.2.1, the approach takes in input a repair dataset $\mathcal{D}rep$, split between misclassified inputs $\mathcal{D}rep_{neg}$ and correctly classified inputs $\mathcal{D}rep_{pos}$.

`FairArachne`

In order to assess whether considering fairness both in fault localization and repair is necessary, we experiment with an approach (named `FairArachne`) that does not use our proposed fault localization approach, but the one from `Arachne`; the repair phase, instead, uses our `FairRep` repair technique (which is fairness-aware). This setup helps isolate the impact of using fairness-aware fault localization. We indicate with $\texttt{FairArachne}_{\mathcal{F}}$ the approach that uses the metric $\mathcal{F}$ (e.g., *SPD*, *EOD*, etc.) during the repair process. For consistency, we use the same repair dataset $\mathcal{D}rep$ used in `FairFLRep`; the only difference is that, by following `Arachne`'s approach [39], the negative samples $\mathcal{D}rep_{neg}$ used in fault localization only contain the deprived community.

`FairMOON`

It is a variant of `FairFLRep` that combines `MOON`'s spectrum-based fault localization [92] with `FairFLRep`'s fairness-aware repair. We indicate with $\texttt{FairMOON}_{\mathcal{F}}$ the approach that uses the fairness metric $\mathcal{F}$ during the repair process. We use the same repair data as in `FairArachne`.

`Arachne`

It is a widely recognized repair technique [39] primarily designed to address misclassification errors in DNNs. Although originally intended for general model repair, the authors have noted that `Arachne` can be applied to fairness-related domains. We follow the guidelines from the original study to adapt `Arachne` for fairness repair. As repair dataset, we use the same used in `FairArachne`.

`LIMI`

It is a fairness testing method based on generating natural discriminatory instances through surrogate boundary approximation in a generative model's latent space [95]. To adapt `LIMI` for fairness repair evaluation, we follow the retraining strategy proposed by the original authors and by Zhang et al. [**?**]: discriminatory instances generated by `LIMI` are randomly selected at 30% the size of the original training dataset, then combined with the original data, and the model is retrained on this augmented dataset. We use `LIMI` as a baseline to assess

how retraining-based bias mitigation compares to direct fault localization and neuron-level repair, as proposed in `FairFLRep`.

5.4.4 Experimental setup

An *experiment* is the execution of one approach (see Section 8.3.3), over a benchmark (i.e., a model trained over a dataset), for one of the fault localization settings. For image datasets, there are eight benchmark models (two models trained over four datasets); `SettSame` is applicable for all the eight benchmark models, while `SettDiff` for six benchmark models. For tabular datasets, there are four benchmarks (one model trained over four datasets) that can only experimented over `SettDiff`; for three datasets, we experiment with two sensitive attributes, while for a dataset we experiment with only one. So, in total, there are 21 experiments. Note that `LIMI` did not produce any discriminative input for the dataset `Student` and, therefore, cannot be used for this dataset. To account for the randomness of the search algorithm used in repair, by following [195], each experiment has been executed 30 times.

We configure the PSO algorithm used during `FairFLRep` repair using the same setting as in the original `Arachne` study [39]. Each individual repair is represented by a swarm of 100 particles, where each particle corresponds to a candidate solution (i.e., a specific weight adjustment). The optimization runs for a maximum of 100 generations, but includes an early stopping criterion: if the best candidate solution does not improve for 10 consecutive generations, the search terminates early. Regarding `FairMOON`, for fault localization, we use the *DStar* formula with exponent 3, as recommended in the original `MOON` study [92]. We select the top 15% suspicious neurons ($\lambda = 0.15$) for repair based on suspiciousness scores. Test input selection is guided by `MOON`'s Suspicious Neuron Activation (SNA) and Test Input Diversity (TID) objectives. Regarding `LIMI` [95], we run `LIMI` using the implementation and default hyperparameters provided in the original replication package. Specifically, the candidate probing parameter λ is set to 0.3, as recommended by the authors to balance between avoiding duplicate test cases (λ too small) and preventing samples from deviating too far from the decision boundary (λ too large). For generative modeling, we use CTGAN with the default settings: 300 training epochs and batch size of 500 for each tabular dataset.

5.4.5 Evaluation metrics and statistical analysis

To validate the effectiveness of our `FairFLRep` technique against baseline methods, we compare their results in terms of:

Fairness Metrics ($\mathcal{F}$): We employ four group fairness metrics: *DI*, *SPD*, *EOD* and *FPR*. These metrics allow us to capture different aspects of group fairness in DNN classifiers.

Accuracy: The accuracy of the repaired model is crucial to assess its overall utility. `FairFLRep` aims to improve fairness while preserving the model's accuracy as much as possible.

We will report the distributions of the values of the metrics across different runs and compare them using appropriate statistical tests [195]. Specifically, given an experiment (i.e., a given benchmark model and a repair setting (`SettSame` or `SettDiff`)) and a metric M (a fairness metric or accuracy), we compare the distributions of M of `FairFLRep` and a baseline approach across the 30 runs. Specifically, we employ the Mann-Whitney U test to assess the statistical significance difference of the distributions. If the p-value is less than 0.05, we reject the null hypothesis that there is no statistically significant difference. In case of significant difference, we use Vargha and Delaney's $\hat{A}_{12}$ [195] effect size statistic to quantify the magnitude of the difference. A value of $\hat{A}_{12}$ greater than 0.5 means that then first approach produces significantly higher values of M; a value of $\hat{A}_{12}$ lower than 0.5 means the opposite. We further categorize the effect size as follows [196]: *negligible* if $0.5 \leq \hat{A}_{12} < 0.556$ or $0.494 \leq \hat{A}_{12} < 0.5$; *small* if $0.556 \leq \hat{A}_{12} < 0.638$ or $0.362 \leq \hat{A}_{12} < 0.494$; *medium* if $0.638 \leq \hat{A}_{12} < 0.714$ or $0.286 \leq \hat{A}_{12} < 0.286$; *large* if $\hat{A}_{12} \geq 0.714$ or $\hat{A}_{12} \leq 0.286$.

We categorize the results of the comparison of `FairFLRep` with a baseline approach into:

Win (`W`): `FairFLRep` outperforms the baseline approach with at least small $\hat{A}_{12}$ value. We will use ✓, ✓✓, and ✓✓✓ to indicate a small, medium, and large $\hat{A}_{12}$ value.

Tied (`T`): The difference between `FairFLRep` and the baseline is not significant or negligible.

Loss (`L`): The baseline outperforms `FairFLRep` with at least small $\hat{A}_{12}$ value. We will use $\times$, $\times\times$, and $\times\times\times$ to indicate a small, medium, and large $\hat{A}_{12}$ value.

5.5 Experimental Results

This section presents the results of our empirical evaluation, addressing the research questions outlined in Section 8.3.1. It is important to note that all reported results are based on the test sets.

5.5.1 RQ1 – Can `FairFLRep` effectively improve fairness in DNN?

We evaluate the effectiveness of our proposed approach `FairFLRep` in the task of repairing model fairness. The evaluation in this section is broken into two sub-experiments for two fault localization settings (`SettSame` and `SettDiff`); for image datasets, we compare `FairFLRep` with `Arachne`, `FairArachne`, and `FairMOON`; for tabular datasets, we compare it

with `Arachne`, `FairArachne`, `FairMOON`, and `LIMI`.

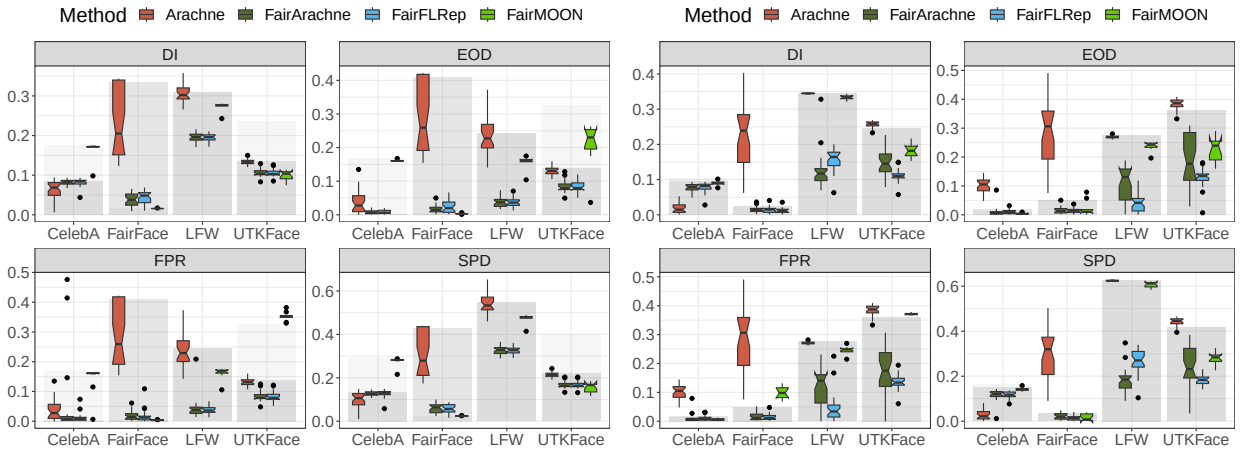
`SettSame` – Image datasets

Here, we present the results of repairing bias and unfairness in gender prediction for the two DNN models trained on the four image datasets.

Figure 5.3 shows the results of the repaired models using `FairFLRep`, `FairArachne`, `Arachne`, and `FairMOON` for each benchmark. The figure groups results based on the fairness metrics used during the repair process (*DI*, *EOD*, *SPD*, and *FPR*) applied to the `VGG16` and `LeNet-5` models. The gray bars represent the fairness score of the original model.

Overall, `FairFLRep` outperforms `Arachne` in most of the cases. `FairFLRep` exhibits greater stability, with smaller deviations across different runs, indicating that it produces consistent results. This stability is observed across multiple datasets. The consistent results suggest that integrating fairness considerations into both the fault localization and repair stages leads to more reliable and fair models across sensitive groups. `FairArachne` shows moderate improvements in fairness, outperforming `Arachne` in several cases. However, it does not perform as well as `FairFLRep`, particularly in dataset like `UTKFace`.

`FairMOON`, similar to `FairArachne`, occasionally matches or slightly outperforms `FairFLRep` on specific cases (e.g., `FairFace` with `VGG16`). However, its performance is highly inconsistent across datasets and fairness metrics. For example, `FairMOON` significantly underperforms on



(a) `VGG16` (b) `LeNet-5`

Figure 5.3 RQ1 – `SettSame` – Image datasets – Fairness of the repaired models (the lower the value, the better). The gray-bar plot indicates the fairness of the original model.

.

DI, *EOD*, and *FPR* on `UTKFace` and `CelebA`, especially with the `VGG16` model. Moreover, while `FairMOON` shows competitive performance on `FairFace` (`VGG16`), this is not the case for `LeNet-5`, where it severely degrades the original model's fairness, particularly on *FPR*. These inconsistencies highlight `FairMOON`'s limited reliability as a comprehensive fairness repair method.

Unlike `FairFLRep`, `Arachne` occasionally degrades the fairness of the original model. In certain cases, the repaired model performs worse than the original, especially on metrics like *DI*, *FPR*, and *EOD* for `LeNet-5` models trained on the `FairFace` and `CelebA` datasets (see the exceptions discussed below). This degradation is likely because `Arachne` focuses more on optimizing accuracy, without sufficiently addressing fairness disparities. In contrast, `FairFLRep` preserves the original fairness of the model.

According to the 80% rule [51], which considers a *DI* value > 0.2 as unfair, some of the models repaired using `Arachne` on the `FairFace` dataset had *DI* values ranging between 0.24 and 0.4, indicating significant unfairness. These results suggest that `FairFLRep` is more effective at reducing unfairness in DNNs while preserving the overall performance, whereas `Arachne` may not provide the same level of fairness improvement.

Consistency Across Datasets The results from multiple datasets highlight that `FairFLRep` provides more reliable fairness improvements compared to `Arachne`, `FairArachne`, and `FairMOON`. While `Arachne` shows some improvement in specific cases, its focus on accuracy often leads to inconsistent fairness outcomes. The following points summarize the key insights observed across the datasets:

1. `FairFLRep` consistently provides more balanced and comprehensive fairness improvements across various metrics and datasets, demonstrating its versatility and effectiveness in addressing fairness concerns across diverse settings.
2. Interestingly, on the `CelebA` dataset, `Arachne` performs relatively well for *DI* and *SPD*, where it achieves lower fairness scores compared to other datasets and metrics. One potential reason for this is that the original `VGG16` and `LeNet-5` models trained on `CelebA` already exhibit relatively fair behavior, which gives `Arachne` a better starting point to improve accuracy without significantly harming fairness.
   However, for other metrics such as *EOD* and *FPR*, and across other datasets, `FairFLRep` consistently outperforms `Arachne`. This suggests that while `Arachne` may excel in specific scenarios—particularly when the original model is already relatively fair—it lacks robustness across a broader range of fairness measures and datasets.
3. While `FairArachne` generally underperforms compared to `FairFLRep`, there are a few

instances where FairArachne outperforms FairFLRep on specific metrics, such as *DI* and *SPD* on the LFW dataset. This outcome may be due to FairArachne's ability to focus on fairness during the repair phase alone, which may allow it to more directly target group-level disparities (as measured by *DI* and *SPD*). Additionally, LFW might present simpler bias patterns, where fairness issues are more straightforward (e.g., fewer confounding factors), making FairArachne more effective in optimizing these specific fairness metrics.

4. FairMOON demonstrates competitive results on certain datasets and metrics but fails dramatically on others, particularly UTKFace and CelebA, reflecting instability across diverse settings.

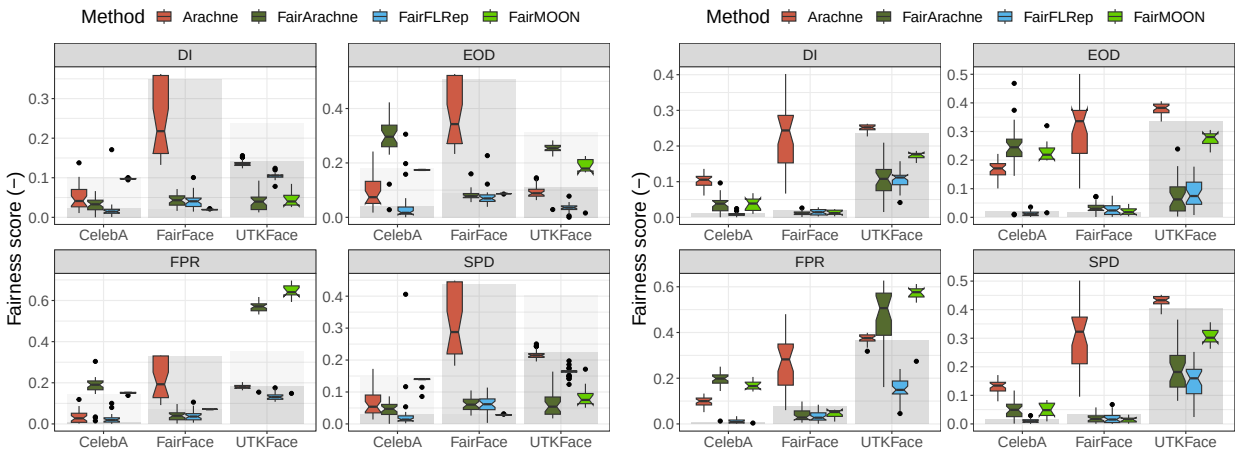
> **Answer to RQ1 (SettSame – Image datasets).** *Across most datasets (3 out of 4), using **FairFLRep** provides a more consistent and robust solution for improving fairness in DNNs. It outperforms **Arachne** across most fairness metrics and datasets. While **FairArachne** provides some gains in fairness, it lacks the robustness and consistency achieved by **FairFLRep**, especially in reducing bias across a broader range of fairness metrics and datasets. **FairMOON**, though often better than **Arachne**, shows variable performance across datasets and metrics. This validates the effectiveness of applying fairness-aware techniques in both stages (fault localization and repair), resulting in significantly better fairness outcomes for the models.*

SettDiff — Image datasets

We present the results of repairing bias in image datasets: race disparity in the UTKFace and FairFace datasets, and age bias in the CelebA dataset. In all the cases, the class being predicted is gender. Figure 5.4 reports the results. As in the previous figure (Figure 5.3), the boxplots report the value of the fairness metric $\mathcal{F}$ under repair in the repaired model, and the gray bars represent the fairness scores of the original model.

According to Figure 5.4, Arachne shows some improvement over the original model, but its impact remains limited across all datasets and fairness metrics. Overall, Arachne consistently struggles to reduce fairness scores, highlighting its limitations in fairness-related tasks.

FairArachne, which incorporates fairness only during the repair phase, is more effective than Arachne across most metrics and datasets. However, it performs worse than the original model on the CelebA dataset for all fairness metrics. This suggests that FairArachne, although it is fairness-aware during the repair phase, may not localize the correct set of faulty neurons responsible for fairness disparities, especially in a dataset like CelebA where



(a) `VGG16` (b) `LeNet-5`

Figure 5.4 RQ1 – `SettDiff` – Image datasets – Fairness of the repaired models (the lower the value, the better). The gray-bar plot indicates the fairness of the original model.

the original models already show relatively fair behavior. In such cases, fairness improvements may be insufficient, or in some instances, the adjustments may degrade the original fairness of the model, as seen in both `Arachne` and `FairArachne` performing worse on `CelebA`.

In contrast, `FairFLRep` outperforms both `Arachne` and `FairArachne` across all metrics and datasets, consistently lowering fairness scores with minimal variation. `FairFLRep` demonstrates higher consistency and robustness in mitigating bias across multiple datasets and metrics. This reinforces the advantage of using `FairFLRep`, which incorporates fairness awareness in both the fault localization and repair phases, effectively reducing bias and with minimal variation. `Arachne`, by comparison, shows greater variability and, in some cases, even performs worse than the original model in terms of fairness. `Arachne`'s emphasis on optimizing accuracy may contribute to this issue, potentially leading to greater disparities in treatment across sensitive groups.

`FairMOON` sometimes produces competitive results on some metrics and datasets but shows higher variability compared to `FairFLRep`. For example, on `FairFace` in `VGG16`, `FairMOON` outperforms in *SPD* and *DI*, but fails to match `FairFLRep`'s consistency across *EOD* and *FPR*, especially on `CelebA`. This further reinforces the importance of a methodical integration of fairness objectives across the full repair pipeline.

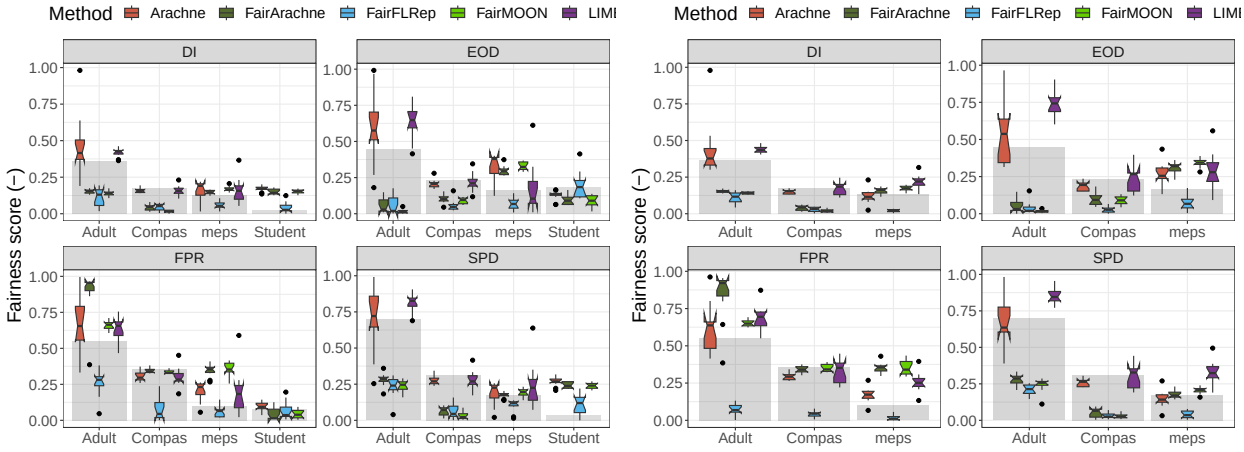
> ◎ **Answer to RQ1 (`SettDiff` – Image datasets).** *The consistently lower fairness scores with with minimal variation across all metrics in* ***`FairFLRep`*** *indicate that this*

> *method is particularly effective at reducing bias, while remaining robust across datasets. For both fairness repair tasks (addressing disparities in age and race), these improvements are more pronounced in* ***FairFLRep****, that is fair-aware both in the fault localization and repair phases.*
> *While* ***FairArachne*** *and* ***FairMOON*** *offers moderate improvements, they lacks the robustness and consistency of* ***FairFLRep****, particularly when dealing with more complex fairness issues such as race and age disparities. This observation suggests that incorporating fairness mechanisms in both the fault localization and repair stages is crucial to achieving comprehensive fairness in DNNs.*

`SettDiff` – Tabular datasets

We now consider the `SettDiff` scenario using tabular data. We evaluate fairness repair for two cases: *gender* bias and *race* bias, using the four tabular datasets. For both bias cases, the predicted class is the same.

Gender bias repair Figure 5.5a presents the fairness scores for models repaired for *gender* bias across the four tabular datasets and the four fairness metrics. Each bar group in the figure represents the results for one dataset and one fairness metric. The compared methods are `FairFLRep` against `Arachne`, `FairArachne`, `FairMOON`, and `LIMI`. According to Figure 5.5a, `FairFLRep` achieves the lowest fairness scores (i.e., best fairness) across most datasets and



(a) Sensitive attribute $\mathcal{A}$: *Gender* (b) Sensitive attribute $\mathcal{A}$: *Race*

Figure 5.5 RQ1 – `SettDiff` – Tabular datasets – Fairness of the repaired models (the lower the value, the better). The gray-bar plot indicates the fairness of the original model.

metrics, outperforming all baselines, particularly in *DI*, *EOD*, and *FPR*. `FairArachne` performs moderately better than `Arachne` and `LIMI` in some cases, though its improvements are not consistent across datasets. `FairMOON` shows competitive performance for *SPD*, especially on the `COMPAS` dataset, but tends to under-perform compared to `FairFLRep` in *EOD* and *FPR*. `LIMI` performs inconsistently; while it sometimes improves fairness compared to `Arachne`, it is generally less stable and less effective overall. Similarly, `Arachne` shows some marginal improvement in certain metrics (notably on `MEPS` for *SPD*), but often falls short or even increases bias due to its lack of fairness awareness.

Race bias repair Figure 5.5b displays the results for repairing *race* bias using the tabular datasets, except for the `Student` dataset which does not include race as a sensitive feature. The results indicate that `FairFLRep` continues to provide the most significant fairness reductions, especially in *DI* and *FPR*, across all datasets, affirming its advantage in handling race-related bias. `FairArachne` again outperforms `Arachne` in several cases but shows increased variance in its results, indicating lower stability across fairness metrics. Notably, `FairArachne` and `Arachne` perform worse than the original model on some metrics in the `COMPAS` and `MEPS` datasets, highlighting their potential to degrade fairness under distribution shifts. `FairMOON` provides competitive results on *SPD* in the `Adult` dataset but shows inconsistency in other metrics. `LIMI` struggles to consistently match the fairness gains of `FairFLRep`, particularly on datasets like `MEPS` and `Adult`.

Results of both biases reinforce the benefit of incorporating fairness considerations into both the fault localization and repair phases, as done in `FairFLRep`. The method reliably localizes fairness-related faults and performs targeted interventions that generalize across broader datasets, thereby achieving better fairness outcomes for both gender and race biases.

> ◎ **Answer to RQ1 (`SettDiff` – Tabular datasets).** *Results show the consistent superiority of* ***`FairFLRep`*** *across all tabular datasets, demonstrating its robustness in identifying and repairing fairness issues.* ***`FairFLRep`*** *achieves more reliable and significant improvements across multiple fairness metrics when compared to baselines, for both gender and race.*

5.5.2 RQ2 – Can `FairFLRep` effectively improve the fairness without harming the accuracy?

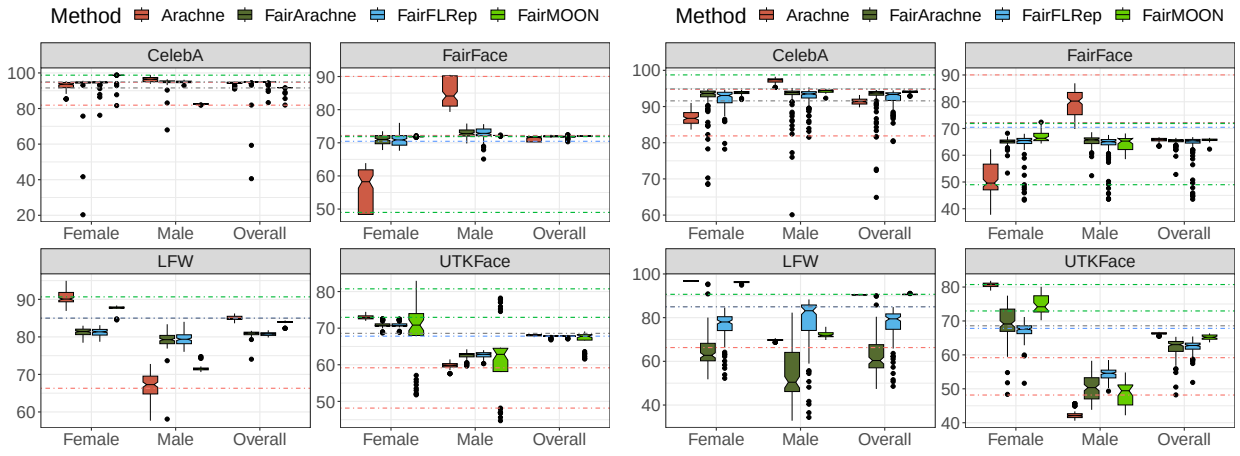
So far, we have observed that `FairFLRep` can effectively improve fairness in DNN models. However, it is equally important to consider the impact of these fairness improvements on the

model's predictive performance. Indeed, it could happen that, in order to increase fairness, some previously correctly classified input is misclassified in the repaired model. If fairness constraints can be optimized while maintaining or minimally affecting accuracy, this would demonstrate the effectiveness of the proposed repair technique in balancing fairness and performance. Below, we assess the accuracy outcomes of `FairFLRep` compared to the baseline approaches.

`SettSame` – Image datasets

This section reports the accuracy metrics for *gender* prediction for all the models and datasets. The accuracy is broken down by sensitive categories (*female* and *male*) as well as the overall accuracy for each dataset. Figure 5.6 illustrates the accuracy metrics of the repaired models using the four compared methods. The figure also reports the value of the accuracy of the original model (three dashed lines showing overall accuracy, and accuracy for *male* and *female*). We observe that the original model shows varied performance across the datasets, with a notable gender gap in accuracy. In most cases, the model performs better for one gender group over the other, indicating a potential imbalance in accuracy before any repair is applied.

Regarding the results of the repaired models, `Arachne` tends to exhibit significant differences in accuracy between *female* and *male* groups across most datasets, although it improves overall accuracy in some cases. In contrast, `FairFLRep` achieves more balanced accuracy between *male* and *female* groups in the `LFW`, `CelebA`, and `FairFace` datasets. For instance,



(a) `VGG16` (b) `LeNet-5`

Figure 5.6 RQ2 – `SettSame` – Image datasets – Accuracy of the repaired models

VGG16 repaired with FairFLRep achieves a median accuracy of 70% for *female* and 72% for *male*, significantly reducing the gender disparity. This more balanced accuracy within the sensitive attribute is due to the improvement of fairness (i.e., they are correlated).

FairMOON produces more balanced accuracy than Arachne in some datasets (particularly FairFace) but struggles in others. It shows notable gender gaps in UTKFace and CelebA, especially with the LeNet-5 model on UTKFace, and with both architectures on CelebA and LFW. While FairMOON improves overall accuracy, the results highlight its limited ability to ensure balance across sensitive groups. FairArachne delivers moderate improvements in balanced accuracy, generally outperforming Arachne and FairMOON, but still falling short of the performance of FairFLRep in most datasets. There are some datasets where the accuracy for the *male* or *female* categories is less balanced, which suggests that the combination considering fairness only in repair provides some benefits but lacks the precision and consistency of considering fairness in both stages.

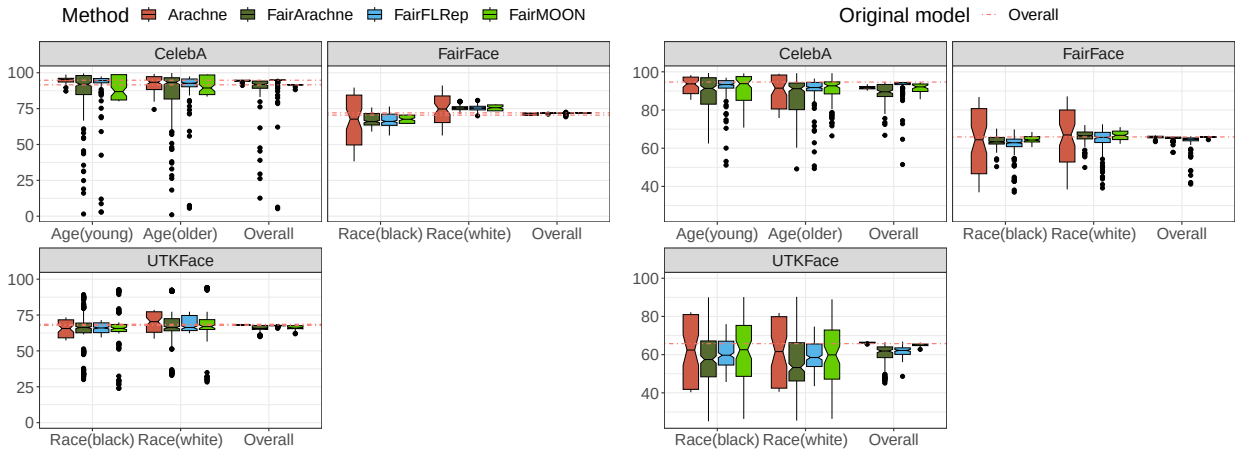
In some cases, particularly for the LFW dataset with LeNet-5 model, FairArachne resulted in a noticeable drop in accuracy when fairness was improved in some datasets (see fairness score in Figure 5.3b). This suggests that FairArachne might not be accurately identifying the problematic weights for fairness repair, causing a conflict between accuracy and bias mitigation. However, this issue is not present in the FairFLRep method, where fairness is explicitly considered in both the fault-localization and repair phases. This ensures that the correct weights are targeted for adjustment, allowing the model to maintain a balanced accuracy while effectively addressing bias. These results validate the effectiveness of the FairFLRep, confirming its superiority in achieving fairness without compromising performance.

> ◉ **Answer to RQ2 (SettSame – Image datasets).** *The results indicate a clear pattern: (i) While **Arachne** improves accuracy, it fails to fully address fairness, as disparities often persist. (ii) **FairArachne**, which introduces fairness-awareness solely in the repair phase, improves fairness, but a significant drop in accuracy on some datasets (e.g., **LFW**) suggests a mismatch between **Arachne**'s fault localization and fairness-aware repair. (iii) **FairMOON**, although it improves overall accuracy, has a limited ability to achieve balance across sensitive groups, as it shows notable gender gaps in **UTKFace** and **CelebA**. (iv) In contrast, **FairFLRep** effectively reduces bias across all datasets (especially in highly biased ones like **LFW** and **UTKFace**) while maintaining or enhancing overall accuracy. This demonstrates **FairFLRep**'s ability to target the appropriate weights, achieving fairness without compromising performance.*

`SettDiff` – Image datasets

Figure 5.7 compares the performance of `FairFLRep` against baselines in the context of `SettDiff`, where the prediction class (*gender*) is different from the sensitive attribute (*race* or *age*). For this setting, only datasets `UTKFace`, `CelebA`, and `FairFace` are used; the accuracy is presented for different sensitive groups (*race*: *black* vs. *white*; *age*: *young* vs. *old*) and overall accuracy.

Across all datasets, `FairFLRep` consistently maintains competitive accuracy compared to `Arachne`, and even outperforms `FairArachne` in some cases (e.g., in `UTKFace` with `VGG16`, and `CelebA` with both `VGG16` and `LeNet-5`). These results further validate the effectiveness of `FairFLRep` in targeting the appropriate weights for repair without compromising accuracy across diverse fairness repair tasks, similarly to the observations in `SettSame`. The `FairFLRep` method successfully minimizes disparities between sensitive groups (*race*, *age*), ensuring more balanced accuracy while also maintaining or improving overall model performance. `FairMOON` shows mixed results. While it demonstrates relatively strong overall accuracy in some cases—such as `FairFace` with `VGG16`–it frequently shows wider disparities across some sensitive groups. On the `UTKFace` dataset, `FairMOON` presents noticeable accuracy fluctuations between sensitive groups, suggesting instability and a less reliable group-level performance. On `CelebA`, `FairFLRep` slightly outperforms `FairMOON` both in terms of group-level and overall accuracy, indicating that `FairFLRep` achieves better trade-offs between fairness and model performance, even in competitive scenarios.



(a) `VGG16` (b) `LeNet-5`

Figure 5.7 RQ2 – `SettDiff` – Image datasets – Accuracy of the repaired models

> ◉ **Answer to RQ2 (`SettDiff` – Image datasets).** *Our evaluation indicates that* ***`FairFLRep`*** *can effectively minimize disparities between race and age sub-groups. It ensures more balanced accuracy while maintaining or improving overall model performance.*

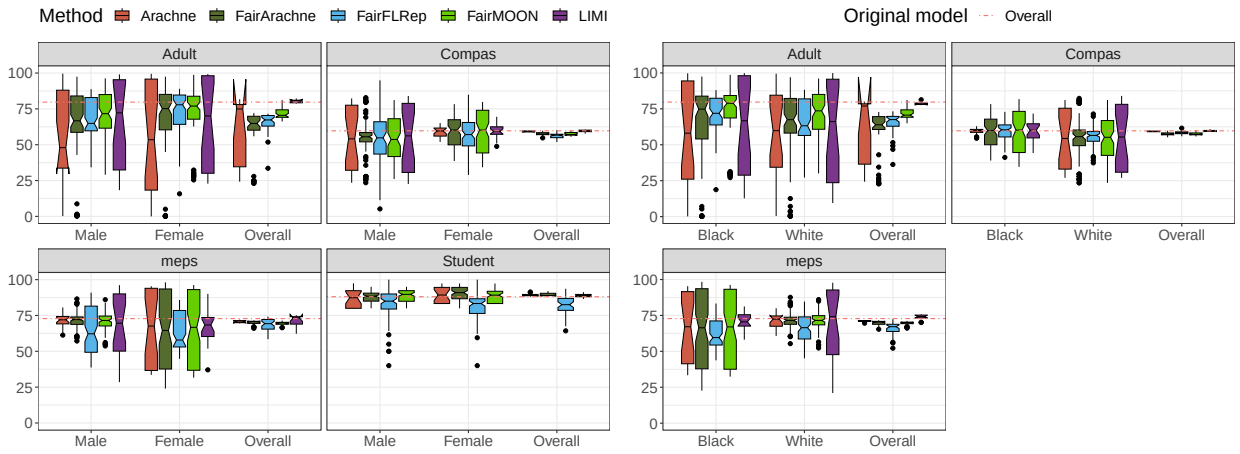
`SettDiff`– Tabular datasets

Figure 5.8 compares the accuracy performance of `FairFLRep` against the baseline methods under `SettDiff` settings across tabular datasets, with a focus on sub-groups disparities (`Gender` and `Race`) and overall accuracy.

According to the results, `Arachne` and `LIMI` exhibit pronounced `Gender` and `Race` accuracy gaps, revealing a contrasting performance pattern compared to `FairFLRep`. While `Arachne` and `LIMI` improves the overall accuracy, they often introduce or exacerbate disparities between subgroups across most datasets. This suggest that their emphasis on maximizing the global accuracy may come at the cost of individual accuracies, especially for minority groups.

In contrast, `FairFLRep` consistently demonstrates a more balanced accuracy between subgroups across all datasets. It achieves this without significantly sacrificing overall accuracy, underscoring its strength in reducing subgroups disparities while maintaining competitive model performance.

When compared to `FairArachne`, the results show that `FairArachne` achieves relatively more



(a) Gender (b) Race

Figure 5.8 RQ2 – `SettDiff` – Tabular datasets – Accuracy of the repaired models

balanced subgroup accuracy than `Arachne` or `LIMI`. However, `FairArachne` also exhibits instability in certain cases– for example, large accuracy variations are observed in the *Female* and *Black* subgroups for the `MEPS` dataset–indicating a lack of robustness.

`FairMOON` performs competitively in terms of overall accuracy, but shows noticeable variability within minority groups, particularly for *Female* and *Black* subgroups in the `MEPS` and `COMPAS` datasets. This subgroups disparities suggest that `FairMOON`, while promising, may not consistently deliver equitable outcomes across sensitive attributes.

> ◉ **Answer to RQ2 (`SettDiff` – Tabular datasets).** *`FairFLRep` demonstrates superior subgroups balance and competitive overall accuracy across the tabular datasets. Unlike `Arachne` and `LIMI`, which often trade subgroup fairness for overall accuracy, `FairFLRep` achieves both. While `FairArachne` and `FairMOON` show some promise, their inconsistency across minority subgroups limits their reliability. `FairFLRep` remains the most stable and effective solution for mitigating `Gender` and `Race` disparities in tabular classification tasks.*

5.5.3 Statistical analysis of the model fairness and model accuracy

In this section, we present the statistical analysis of the fairness and accuracy of models repaired using different techniques, to summarize the observations of RQ1 and RQ2 presented in Section 5.5.1 and Section 5.5.2. As before, the analysis is split for the two fault localization settings.

`SettSame` – Image datasets

In Table 5.2, we report the statistical comparison between the different repair methods in terms of fairness and overall accuracy of the repaired models. As explained in Section 5.4.5, we use the categorization `W` to indicate where `FairFLRep` outperforms (i.e., wins) the baseline (columns); `T` where the comparison of the two methods is tied; and `L` when the baseline outperforms `FairFLRep`. Each subtable reports the results for the different fairness metrics $\mathcal{F}$ used as fitness functions. For example, $\texttt{FairFLRep}_{EOD}$ indicates the case where `FairFLRep` used the *EOD* as the fairness definition. Each subtable reports the results for the fairness metric and for the accuracy. At the bottom of the table, the summary provides an overall count of wins, ties, and losses across different datasets and models, giving a comprehensive view of the effectiveness of `FairFLRep`.

Table 5.2 RQ2 – SettSame – Image datasets – Statistical comparison between $\texttt{FairFLRep}_{\mathcal{F}}$ and the baselines. W: $\texttt{FairFLRep}_{\mathcal{F}}$ wins against the baseline; T: no statistical significant difference; L: $\texttt{FairFLRep}_{\mathcal{F}}$ loses to the baseline. The number of symbols ✓ and × represents the difference magnitude (1: small, 2: medium, 3: large).

Dataset	Model	$\texttt{FairFLRep}_{EOD}$ vs Arachne *EOD*	$\texttt{FairFLRep}_{EOD}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairArachne}_{EOD}$ *EOD*	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairArachne}_{EOD}$ Accuracy	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairMOON}_{EOD}$ *EOD*	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairMOON}_{EOD}$ Accuracy
UTKFace	VGG16	W (✓✓)	L (× × ×)	T	T	W (✓✓✓)	W (✓✓✓)
	LeNet-5	W (✓✓)	L (× × ×)	W (✓✓)	T	W (✓✓✓)	L (× × ×)
Fair Face	VGG16	W (✓✓✓)	W (✓✓✓)	T	T	L (× × ×)	L (× × ×)
	LeNet-5	W (✓✓✓)	T	T	T	T	T
LFW	VGG16	W (✓✓✓)	L (× × ×)	T	T	W (✓✓✓)	L (× × ×)
	LeNet-5	W (✓✓✓)	L (× × ×)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	L (× × ×)
CelebA	VGG16	W (✓✓✓)	W (✓✓✓)	T	T	W (✓✓✓)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	T	T	L (× × ×)	L (× × ×)	L (× × ×)

Dataset	Model	$\texttt{FairFLRep}_{SPD}$ vs Arachne *SPD*	$\texttt{FairFLRep}_{SPD}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairArachne}_{SPD}$ *SPD*	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairArachne}_{SPD}$ Accuracy	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairMOON}_{SPD}$ *SPD*	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairMOON}_{SPD}$ Accuracy
UTKFace	VGG16	W (✓✓✓)	L (× × ×)	T	T	T	W (✓✓✓)
	LeNet-5	W (✓✓✓)	L (× × ×)	W (✓✓✓)	T	W (✓✓✓)	L (× × ×)
Fair Face	VGG16	W (✓✓✓)	W (✓✓)	T	T	L (× × ×)	L (× × ×)
	LeNet-5	W (✓✓✓)	L (× × ×)	W (✓)	T	T	L (× × ×)
LFW	VGG16	W (✓✓✓)	L (× × ×)	T	T	W (✓✓✓)	L (× × ×)
	LeNet-5	W (✓✓✓)	L (× × ×)	L (× × ×)	W (✓✓✓)	W (✓✓✓)	L (× × ×)
CelebA	VGG16	W (✓✓)	T	T	T	W (✓✓✓)	W (✓✓✓)
	LeNet-5	L (× × ×)	W (✓✓✓)	T	T	W (✓✓✓)	L (× × ×)

Dataset	Model	$\texttt{FairFLRep}_{DI}$ vs Arachne *DI*	$\texttt{FairFLRep}_{DI}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairArachne}_{DI}$ *DI*	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairArachne}_{DI}$ Accuracy	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairMOON}_{DI}$ *DI*	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairMOON}_{DI}$ Accuracy
UTKFace	VGG16	W (✓✓✓)	L (× × ×)	T	T	T	W (✓✓✓)
	LeNet-5	W (✓✓✓)	L (× × ×)	W (✓✓✓)	T	W (✓✓✓)	L (× × ×)
Fair Face	VGG16	W (✓✓✓)	W (✓✓)	T	T	L (× × ×)	L (× × ×)
	LeNet-5	W (✓✓✓)	L (××)	T	T	T	T
LFW	VGG16	W (✓✓✓)	L (× × ×)	T	T	W (✓✓✓)	L (× × ×)
	LeNet-5	W (✓✓✓)	L (× × ×)	L (× × ×)	W (✓✓✓)	W (✓✓✓)	L (× × ×)
CelebA	VGG16	L (× × ×)	W (✓✓)	T	T	W (✓✓✓)	W (✓✓✓)
	LeNet-5	L (× × ×)	W (✓✓✓)	T	T	W (✓✓✓)	T

Dataset	Model	$\texttt{FairFLRep}_{FPR}$ vs Arachne *FPR*	$\texttt{FairFLRep}_{FPR}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairArachne}_{FPR}$ *FPR*	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairArachne}_{FPR}$ Accuracy	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairMOON}_{FPR}$ *FPR*	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairMOON}_{FPR}$ Accuracy
UTKFace	VGG16	W (✓✓✓)	L (× × ×)	T	T	W (✓✓✓)	T
	LeNet-5	W (✓✓✓)	L (× × ×)	W (✓)	T	W (✓✓✓)	L (× × ×)
Fair Face	VGG16	W (✓✓✓)	W (✓✓✓)	T	T	T	T
	LeNet-5	W (✓✓✓)	L (××)	T	T	W (✓✓✓)	T
LFW	VGG16	W (✓✓✓)	L (× × ×)	T	L (××)	W (✓✓✓)	L (× × ×)
	LeNet-5	W (✓✓✓)	L (× × ×)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	L (× × ×)
CelebA	VGG16	W (✓✓✓)	W (✓✓)	T	T	W (✓✓✓)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	W (✓✓✓)	T	T	T	T
TOTAL	**Win**	**29**	**10**	**7**	**4**	**15**	**7**
	Tie	**0**	**3**	**23**	**26**	**7**	**7**
	Loss	**3**	**19**	**2**	**2**	**4**	**18**

According to results in Table 5.2, FairFLRep achieved a high number of wins, often by large margins, when compared to Arachne in terms of fairness results (29 out of 32 cases). There were only 3 cases where the FairFLRep approach lost to Arachne, and no ties were observed in the fairness metrics. However, this trend does not hold when comparing the overall accuracy of models repaired by FairFLRep to those repaired by Arachne. In this case, FairFLRep

only won in 10 out of 32 cases, while losing in 19 cases. This outcome aligns with the focus of Arachne on correcting misclassification problems. Conversely, our results aim to address both fairness and model predictive performance without significantly sacrificing either.

The comparison between FairFLRep and FairArachne further supports this claim: in terms of fairness metrics, FairFLRep won in 7 out of 32 cases, and tied in 23; in terms of overall accuracy, it won in 4 cases, and tied in 26 cases. This demonstrates that FairFLRep can achieve favorable fairness outcomes while maintaining competitive model accuracy, highlighting its effectiveness in balancing these critical aspects.

When comparing with FairMOON, the results are more mixed. In terms of fairness, FairFLRep achieved 15 wins, 7 ties, and 4 losses out of 32 comparisons, indicating that FairMOON underperforms in fairness compared to FairFLRep. Notably, 3 our 4 fairness loses occurred on the VGG16 model repaired on the FairFace dataset, highlighting cases where FairMOON may outperform FairFLRep on specific datasets or metrics. However, FairFLRep maintains overall superiority in fairness, as reflected in the higher total number of wins, while also remaining competitive in accuracy, with 7 ties recorded in accuracy comparisons. These results suggest that while FairMOON may excel in certain configurations, FairFLRep offers more reliable fairness improvements across diverse datasets and model architectures.

> ◉ **Answer to RQ1 and RQ2 (SettSame – Image datasets).** *In summary, FairFLRep can improve model fairness across various metrics, often by large margins, in a statistical significant way. This improvement is achieved with minimal impact on overall accuracy when compared to the baselines Arachne, FairArachne and FairMOON. While FairFLRep tends to sacrifice some accuracy, it remains a highly effective for balancing fairness with predictive performance.*

SettDiff – Image datasets

In Table 5.3, we report the statistical comparison of the repair approaches using the image datasets in SettDiff setting. The results highlight FairFLRep's strong performance, particularly in fairness outcomes across different datasets and model architectures. FairFLRep demonstrates significant wins across all fairness metrics, consistently outperforming Arachne, FairArachne, and FairMOON in fairness metrics. This confirms FairFLRep's effectiveness in reducing group-level bias under complex fairness scenarios.

Compared to Arachne, FairFLRep achieved 23 fairness comparisons across the three datasets and both models, without a single loss. However, FairFLRep falls short in overall accuracy

Table 5.3 RQ1 and RQ2 – `SettDiff` – Image datasets – Statistical comparison between $\texttt{FairFLRep}_{\mathcal{F}}$ and the baselines (Same notation used in Table 5.2)

Dataset	Model	$\texttt{FairFLRep}_{EOD}$ vs Arachne ***EOD***	$\texttt{FairFLRep}_{EOD}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairArachne}_{EOD}$ ***EOD***	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairArachne}_{EOD}$ Accuracy	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairMOON}_{EOD}$ ***EOD***	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairMOON}_{EOD}$ Accuracy
UTKFace	VGG16	W (✓✓✓)	L (× × ×)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	L (× × ×)	T	W (✓✓✓)	W (✓✓✓)	L (× × ×)
FairFace	VGG16	W (✓✓✓)	W (✓✓)	T	T	T	T
	LeNet-5	W (✓✓✓)	W (✓✓✓)	T	T	T	L (× × ×)
CelebA	VGG16	W (✓✓✓)	T	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)

Dataset	Model	$\texttt{FairFLRep}_{SPD}$ vs Arachne ***SPD***	$\texttt{FairFLRep}_{SPD}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairArachne}_{SPD}$ ***SPD***	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairArachne}_{SPD}$ Accuracy	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairMOON}_{SPD}$ ***SPD***	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairMOON}_{SPD}$ Accuracy
UTKFace	VGG16	W (✓✓✓)	L (× × ×)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	L (× × ×)	T	T	W (✓✓✓)	L (× × ×)
FairFace	VGG16	W (✓✓✓)	W (✓✓)	T	T	L (× × ×)	T
	LeNet-5	W (✓✓✓)	L (× × ×)	T	L (× × ×)	T	L (× × ×)
CelebA	VGG16	W (✓✓✓)	T	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓)	W (✓✓✓)	W (✓✓✓)

Dataset	Model	$\texttt{FairFLRep}_{DI}$ vs Arachne ***DI***	$\texttt{FairFLRep}_{DI}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairArachne}_{DI}$ ***DI***	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairArachne}_{DI}$ Accuracy	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairMOON}_{DI}$ ***DI***	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairMOON}_{DI}$ Accuracy
UTKFace	VGG16	W (✓✓✓)	L (× × ×)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	L (× × ×)	T	T	W (✓✓✓)	L (× × ×)
FairFace	VGG16	W (✓✓✓)	W (✓✓)	T	T	L (× × ×)	T
	LeNet-5	W (✓✓✓)	L (× × ×)	T	L (××)	T	L (× × ×)
CelebA	VGG16	W (✓✓✓)	W (✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓)	W (✓✓✓)	T

Dataset	Model	$\texttt{FairFLRep}_{FPR}$ vs Arachne ***FPR***	$\texttt{FairFLRep}_{FPR}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairArachne}_{FPR}$ ***FPR***	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairArachne}_{FPR}$ Accuracy	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairMOON}_{FPR}$ ***FPR***	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairMOON}_{FPR}$ Accuracy
UTKFace	VGG16	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	T
	LeNet-5	W (✓✓✓)	L (× × ×)	W (✓✓✓)	T	W (✓✓✓)	L (× × ×)
FairFace	VGG16	W (✓✓✓)	W (✓✓)	T	T	W (✓✓✓)	T
	LeNet-5	W (✓✓✓)	L (× × ×)	T	L (××)	T	L (× × ×)
CelebA	VGG16	T	T	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)
	LeNet-5	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)
TOTAL	**Win**	**23**	**10**	**11**	**12**	**15**	**10**
	Tie	**1**	**3**	**11**	**8**	**5**	**6**
	Loss	**0**	**11**	**2**	**4**	**4**	**8**

in 11 cases, compared to 10 wins, reflecting `Arachne`'s focus on accuracy rather than fairness.

Against `FairArachne`, `FairFLRep` achieves 11 wins in fairness, 11 ties, and only 2 losses. For accuracy, `FairFLRep` also wins more often (12 wins vs. 4 losses), showing it not only improves fairness more effectively but also retains or improves model accuracy in many cases. The higher number of fairness ties (11) compared to accuracy ties (8) highlights the consistent fairness performance of `FairFLRep` across models and metrics.

Compared to FairMOON, FairFLRep achieves 15 fairness wins, 5 ties, and 4 losses. Most losses occur on *DI* and *SPD* for the VGG16 model on the UTKFace and FairFace datasets, indicating FairMOON's effectiveness is limited to specific dataset–model configurations. In terms of accuracy, FairMOON outperforms FairFLRep in 8 out of 32 comparisons. However, these gains are offset by sharp fairness degradations—especially on UTKFace and CelebA—where FairMOON severely worsens the original model's fairness. These trade-offs, which are not observed in FairFLRep, suggest that FairMOON may prioritize accuracy at the expense of fairness, leading to inconsistent and potentially unstable repair outcomes.

> ◎ **Answer to RQ1 and RQ2 (SettDiff – Image datasets).** *FairFLRep excels at reducing bias across a range of datasets and models, although it occasionally sacrifices some accuracy, especially in comparison to Arachne and FairMOON. The high number of fairness ties and wins, along with competitive accuracy results, suggests that FairFLRep effectively balances fairness and accuracy, confirming its value as a robust fairness repair strategy in complex image-based DNNs.*

SettDiff – Tabular datasets

Table 5.4 summarizes how FairFLRep performs on tabular datasets compared to four baselines under the SettDiff setting. The notation is as in Table 5.2.

According to the results, Arachne and LIMI exhibit performance patterns that contrast with FairFLRep in complementary ways. Compared to Arachne, FairFLRep shows a stronger ability to reduce bias, achieving 26 fairness wins and no fairness losses. However, this improvement in fairness is accompanied by a slightly weaker or more variable performance in accuracy, with 10 losses to Arachne in the accuracy metric.

In comparison to LIMI, FairFLRep again demonstrates clear superiority in fairness, securing 23 fairness wins and zero losses. However, FairFLRep consistently underperforms in accuracy against LIMI, losing all 22 comparisons in accuracy. This tradeoff further illustrates LIMI's emphasis on preserving classification performance, whereas FairFLRep prioritizes fairness repair.

When evaluated against FairArachne, FairFLRep continues to outperform across fairness metrics, achieving 18 fairness wins with only 1 loss, showcasing its effectiveness in reducing bias. Accuracy-wise, the performance is largely balanced, with 11 tie cases, reflecting FairFLRep's ability to enhance fairness without drastically compromising model performance.

In comparison to FairMOON, the results show that while FairFLRep achieves 17 fairness wins

Table 5.4 RQ1 and RQ2 – `SettDiff` – Tabular datasets – Statistical comparison between $\texttt{FairFLRep}_{\mathcal{F}}$ and the baselines (Same notation used in Table 5.2)

Dataset	Model	$\texttt{FairFLRep}_{EOD}$ vs Arachne *EOD*	$\texttt{FairFLRep}_{EOD}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairArachne}_{EOD}$ *EOD*	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairArachne}_{EOD}$ Accuracy	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairMOON}_{EOD}$ *EOD*	$\texttt{FairFLRep}_{EOD}$ vs $\texttt{FairMOON}_{EOD}$ Accuracy	$\texttt{FairFLRep}_{EOD}$ vs LIMI *EOD*	$\texttt{FairFLRep}_{EOD}$ vs LIMI Accuracy
Adult	Gender	W (✓✓✓)	T	T	T	T	T	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	T	T	T	T	T	W (✓✓✓)	L (× × ×)
COMPAS	Gender	W (✓✓✓)	L (× × ×)	W (✓✓✓)	T	W (✓✓✓)	T	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	L (× × ×)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	L (× × ×)
Student	Gender	T	T	L (× × ×)	T	L (× × ×)	T		
MEPS	Gender	W (✓✓✓)	T	W (✓✓✓)	T	W (✓✓✓)	T	T	T
	Race	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)

Dataset	Model	$\texttt{FairFLRep}_{SPD}$ vs Arachne *SPD*	$\texttt{FairFLRep}_{SPD}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairArachne}_{SPD}$ *SPD*	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairArachne}_{SPD}$ Accuracy	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairMOON}_{SPD}$ *SPD*	$\texttt{FairFLRep}_{SPD}$ vs $\texttt{FairMOON}_{SPD}$ Accuracy	$\texttt{FairFLRep}_{SPD}$ vs LIMI *SPD*	$\texttt{FairFLRep}_{SPD}$ vs LIMI Accuracy
Adult	Gender	W (✓✓✓)	T	T	T	T	T	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	T	W (✓✓✓)	T	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)
COMPAS	Gender	W (✓✓✓)	L (× × ×)	T	T	T	T	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	L (× × ×)	T	W (✓✓✓)	T	W (✓✓✓)	W (✓✓✓)	L (× × ×)
Student	Gender	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)		
MEPS	Gender	W (✓✓✓)	T	W (✓✓✓)	T	W (✓✓✓)	T	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)

Dataset	Model	$\texttt{FairFLRep}_{DI}$ vs Arachne *DI*	$\texttt{FairFLRep}_{DI}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairArachne}_{DI}$ *DI*	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairArachne}_{DI}$ Accuracy	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairMOON}_{DI}$ *DI*	$\texttt{FairFLRep}_{DI}$ vs $\texttt{FairMOON}_{DI}$ Accuracy	$\texttt{FairFLRep}_{DI}$ vs LIMI *DI*	$\texttt{FairFLRep}_{DI}$ vs LIMI Accuracy
Adult	Gender	W (✓✓✓)	T	T	T	T	L (× × ×)	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	T	W (✓✓✓)	T	T	L (× × ×)	W (✓✓✓)	L (× × ×)
COMPAS	Gender	W (✓✓✓)	L (× × ×)	T	L (× × ×)	L (× × ×)	T	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	L (× × ×)	T	W (✓✓✓)	T	W (✓✓✓)	W (✓✓✓)	L (× × ×)
Student	Gender	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)		
MEPS	Gender	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)

Dataset	Model	$\texttt{FairFLRep}_{FPR}$ vs Arachne *FPR*	$\texttt{FairFLRep}_{FPR}$ vs Arachne Accuracy	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairArachne}_{FPR}$ *FPR*	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairArachne}_{FPR}$ Accuracy	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairMOON}_{FPR}$ *FPR*	$\texttt{FairFLRep}_{FPR}$ vs $\texttt{FairMOON}_{FPR}$ Accuracy	$\texttt{FairFLRep}_{FPR}$ vs LIMI *FPR*	$\texttt{FairFLRep}_{FPR}$ vs LIMI Accuracy
Adult	Gender	W (✓✓✓)	T	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	T	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)
COMPAS	Gender	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)
	Race	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)
Student	Gender	T	L (× × ×)	T	L (× × ×)	T	L (× × ×)		
MEPS	Gender	W (✓✓✓)	T	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	W (✓✓✓)	T
	Race	W (✓✓✓)	L (× × ×)	W (✓✓✓)	T	W (✓✓✓)	T	W (✓✓✓)	L (× × ×)
TOTAL	**Win**	**26**	**0**	**18**	**6**	**17**	**4**	**23**	**0**
	Tie	**2**	**3**	**7**	**11**	**9**	**10**	**1**	**2**
	Loss	**0**	**16**	**1**	**10**	**2**	**14**	**0**	**22**

and only 2 losses, `FairMOON` maintains a competitive stance in terms of accuracy, also resulting in 10 tie cases. This suggests that `FairMOON`, while less consistent in fairness outcomes, may retain or improve model accuracy in certain scenarios.

> ◉ **Answer to RQ1 and RQ2 (`SettDiff` – Tabular datasets).** *Overall, `FairFLRep` demonstrates a robust advantage in fairness repair across tabular datasets, while maintaining competitive performance in accuracy. These results underscore `FairFLRep`'s gen-*

> *eralizability across both fairness metrics and datasets.*

5.5.4 RQ3 – Which is the "best" fairness metric for repair?

The different fairness metrics capture different aspects of fairness, and several recent studies [26, 159, 197–199] have shown that it is almost impossible to satisfy multiple notions of fairness simultaneously. Some fairness mitigation techniques may excel at improving one metric, such as *SPD*, while being less effective at improving another, like *DI*. This suggests that selecting the appropriate fairness metric is crucial, depending on the specific fairness concerns of the application under test. Moreover, improving one fairness metric may not necessarily lead to proportional improvements across all metrics. For instance, a method that significantly reduces *SPD* might show limited improvement in *DI* or *EOD*. This analysis will highlight the potential trade-offs between different fairness objectives when applying these techniques.

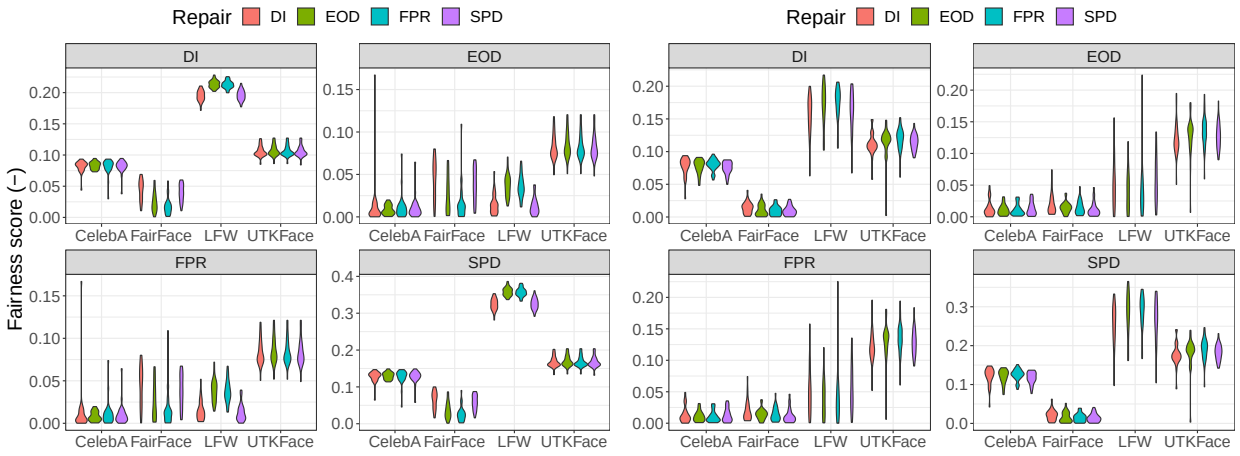
The intuition is that fairness-aware repair techniques should be designed to optimize the most appropriate fairness metric for the specific fairness problem, considering the context and trade-offs inherent in the decision-making domain. In this RQ, we use the term "best" to describe scenarios where optimizing one fairness metric also leads to improvements in another. Specifically, we investigate the impact of optimizing one fairness metric on others based on `FairFLRep` technique (e.g., how optimizing for *DI* affects *SPD*, *EOD*, and *FPR*). Consistently with RQ1 and RQ2, we evaluate performance in terms of both accuracy and fairness across 30 runs.

`SettSame` – Image datasets

In Figure 5.9, we visualize the results of this analysis, showing how optimizing for one fairness metric impacts the performance of other metrics. Each plot in a cell shows the distribution of fairness scores across the fairness metrics after repair.

- The vertical axis represents the fairness score, where lower values indicate a more favorable fairness outcome (i.e., less bias).
- Each cell corresponds to the distribution of a fairness score (shown in the header of the cell) after optimizing for a specific fairness metric (identified by the color of the violin plot), across different datasets. The width of the violin plot shows the density of scores at that level, with wider sections indicating more frequent occurrences.

Based on the patterns and distributions of the violin plots in Figure 5.9, we can gain insights into the relationships between various fairness metrics.



(a) `VGG16` (b) `LeNet-5`

Figure 5.9 RQ3 – `SettSame` – Image datasets – The violin plot visualizes the relationship between optimizing one fairness metric and its effect on other fairness metrics. Each plot in a cell shows the distribution of fairness scores across different fairness metrics after the repair process is applied.

Relationships Between Fairness Metrics: The results in Figure 5.9 indicate that fairness metrics like *DI* and *SPD*, or *FPR* and *EOD*, often have similarly shaped distributions across datasets due to their conceptual relationships. For instance, *DI* and *SPD* both address disparities in positive outcomes across groups, which explains why their violin plots often show similar distributions. Likewise, *EOD* requires equalization of *FPR* and false negative rates (*FNR*) across groups, so changes in *EOD* naturally impact *FPR*. This conceptual overlap between these metrics suggests that optimizing one fairness metric may lead to improvements in others, as verified by the statistical test in Table 5.5.

Consistency Across Datasets: Comparing the results across different datasets, the distributions of *DI* and *SPD*, as well as *EOD* and *FPR*, are consistently similar. This suggests that the relationship between these metrics remains stable across datasets, with no substantial variability. Therefore, improvements in one fairness metric, such as *DI*, are consistently reflected in the other (*SPD*), and the same holds true for *EOD* and *FPR* across all datasets. This consistency reinforces the conceptual overlap between these fairness metrics across various datasets.

Statistical analysis Table 5.5 provides a detailed analysis of the statistical comparison between different fairness metrics and model accuracy when the `FairFLRep` repair technique is applied across the four datasets and the two models. Each row reports the four versions

Table 5.5 RQ3 – SettSame – Image datasets – Statistical comparison between the performance of fairness metrics used in $\texttt{FairFLRep}_{\mathcal{F}}$. W: the fairness metric on the row wins against the fairness metric on the column; T: no statistical significant difference; L: the fairness metric on the row loses against the fairness metric on the column. The number of symbols ✓ and × represents the difference magnitude (1: small, 2: medium, 3: large).

			$\texttt{FairFLRep}_{EOD}$		$\texttt{FairFLRep}_{SPD}$		$\texttt{FairFLRep}_{DI}$		$\texttt{FairFLRep}_{FPR}$	
Dataset	**Model**	**Repair**	***EOD***	**Accuracy**	***SPD***	**Accuracy**	***DI***	**Accuracy**	***FPR***	**Accuracy**
UTKFace	VGG16	$\texttt{FairFLRep}_{EOD}$	-		T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-		T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	T	T	-	-
	LeNet-5	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	W (✓✓)	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	L (××)	T	-	-
FairFace	VGG16	$\texttt{FairFLRep}_{EOD}$	-	-	W (✓✓✓)	T	W (✓✓✓)	T	T	T
		$\texttt{FairFLRep}_{SPD}$	L (××)	T	-	-	T	T	L (× × ×)	T
		$\texttt{FairFLRep}_{DI}$	L (× × ×)	T	T	T	-	-	L (× × ×)	L (××)
		$\texttt{FairFLRep}_{FPR}$	T	T	W (✓✓✓)	T	W (✓✓✓)	W (✓✓)	-	-
	LeNet-5	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	W (✓✓)	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	L (××)	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	W (✓✓)	T	-	-
LFW	VGG16	$\texttt{FairFLRep}_{EOD}$	-	-	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	T	T
		$\texttt{FairFLRep}_{SPD}$	W (✓✓✓)	L (× × ×)	-	-	T	T	W (✓✓✓)	L (× × ×)
		$\texttt{FairFLRep}_{DI}$	W (✓✓✓)	L (× × ×)	T	T	-	-	W (✓✓✓)	L (× × ×)
		$\texttt{FairFLRep}_{FPR}$	T	T	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	-	-
	LeNet-5	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	L (××)	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	L (××)
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	L (××)	W (✓✓)	-	-
CelebA	VGG16	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	T	T	-	-
	LeNet-5	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	T	T	-	-
			W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L
Summary		$\texttt{FairFLRep}_{EOD}$	-	-	1/6/1	1/7/0	1/6/1	1/7/0	0/8/0	0/8/0
		$\texttt{FairFLRep}_{SPD}$	1/6/1	0/7/1	-	-	1/7/0	0/8/0	1/5/2	0/7/1
		$\texttt{FairFLRep}_{DI}$	1/6/1	0/7/1	0/7/1	0/8/0	-	-	2/5/1	0/5/3
		$\texttt{FairFLRep}_{FPR}$	0/8/0	0/8/0	1/6/1	1/7/0	1/4/3	3/5/0	-	-
TOTAL		**Win**	**2**	**0**	**2**	**2**	**4**	**4**	**3**	**0**
		Tie	**20**	**22**	**19**	**22**	**16**	**20**	**18**	**20**
		Loss	**2**	**2**	**3**	**0**	**4**	**0**	**3**	**4**

of FairFLRep instantiated with the four fairness metrics ($\texttt{FairFLRep}_{EOD}$, $\texttt{FairFLRep}_{SPD}$, $\texttt{FairFLRep}_{DI}$, $\texttt{FairFLRep}_{FPR}$); similarly, the last four columns also report the four versions of FairFLRep. Each cell reports the statistical comparison between the repair approach on the row and the one on the column (e.g., $\texttt{FairFLRep}_{EOD}$ vs. $\texttt{FairFLRep}_{SPD}$); the comparison is performed in terms of the fairness metric of the approach on the column (e.g., *SPD*) and in

terms of accuracy. The possible results of the comparison of $\texttt{FairFLRep}_{\mathcal{F}_1}$ and $\texttt{FairFLRep}_{\mathcal{F}_2}$ in terms of $\mathcal{F}_2$ are as follows:

- `W` (*Win*): it indicates that $\texttt{FairFLRep}_{\mathcal{F}_1}$ significantly improves $\mathcal{F}_2$ significantly better than $\texttt{FairFLRep}_{\mathcal{F}_2}$.
- `L` (*Loss*): it indicates that $\texttt{FairFLRep}_{\mathcal{F}_1}$ is significantly worse at improving $\mathcal{F}_2$ compared to $\texttt{FairFLRep}_{\mathcal{F}_2}$.
- `T` (*Tie*): it indicates that $\texttt{FairFLRep}_{\mathcal{F}_1}$ and $\texttt{FairFLRep}_{\mathcal{F}_2}$ achieve similar results in terms of $\mathcal{F}_2$.

The comparison in terms of accuracy, instead, tells which approach better maintains or improves model accuracy. A `W` (Win) or `L` (Loss) here suggests a substantial impact on accuracy.

At the bottom, the table provides a summary of the total wins, ties, and losses across each fairness metric used during repair. Based on the summary in Table 5.5, we can derive several key insights:

Fairness metrics are not independent: Improving one fairness metric often impacts others, either positively or negatively. For example, optimizing *SPD* can result in improvements in *EOD* and *DI*, but may negatively affect *FPR*. This highlights that fairness metrics are interrelated.

Trade-offs: There are inherent trade-offs in fairness optimization. Some metrics may improve at the expense of others, as seen when optimizing *FPR* frequently results in losses in *DI* and *SPD*, especially in challenging datasets like `LFW`.

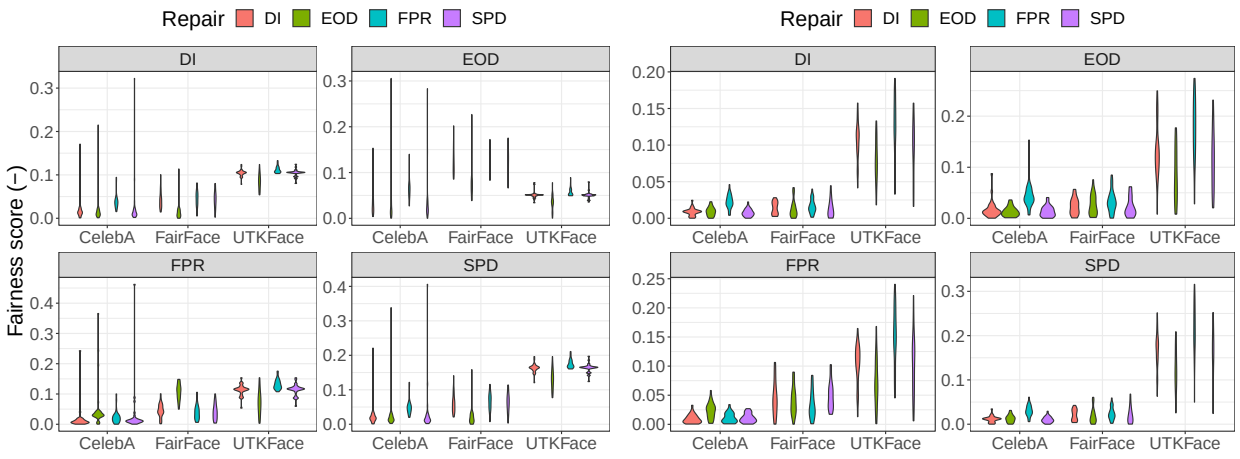
Consistency across the dataset and model architectures: Across the datasets, `UTKFace` and `CelebA` exhibit consistent patterns, where improvements in one fairness metric (like *SPD* or *DI*) often yield similar effects across others. In contrast, `FairFace` and `LFW` datasets show stronger trade-offs, particularly when optimizing *FPR*, with frequent losses in *DI* and *SPD*. Across the model architectures, both the `VGG16` and `LeNet-5` models exhibit trade-offs. However, `VGG16` tends to produce larger improvements in fairness but also larger trade-offs, while `LeNet-5` shows more stability, with more tied outcomes across fairness metrics. This suggests that complex models like `VGG16` may be more flexible but also more prone to variability in fairness gains and losses.

> ◉ **Answer to RQ3 (`SettSame` – Image datasets).** *Our analysis confirms the conceptual overlap between metrics like DI and SPD or EOD and FPR, suggesting that optimizing one can lead to similar outcomes. However, optimizing one metric can also*

> *have both positive and negative effects on others, and the impact of fairness optimization depends heavily on the model architecture. There is no universally optimal fairness metric. The "best" metric depends on the specific fairness concerns in the given context. A "one-size-fits-all" approach to fairness is ineffective. Careful selection of the most relevant fairness metric is crucial for achieving the desired improvements, with trade-offs, especially in challenging metrics like DI and FPR.*

`SettDiff` – Image datasets

The violin plots shown in Figure 5.10 illustrate the relationship between using one fairness metric for repair and its effects on other fairness metrics across the three datasets (`CelebA`, `FairFace`, `UTKFace`) for the task of addressing race disparities (in `UTKFace` and `FairFace`) and age disparities (`CelebA`). By focusing on one fairness metric for repair (e.g., *DI* or *SPD*), the figure reveals how that repair impacts other metrics. For example, repairing *DI* may improve fairness measured by *DI* but have mixed effects on other metrics like *FPR* or *EOD*. This is demonstrated by the spread and shape of the violins for each dataset and fairness metric. The effect of the repair slightly differs across datasets and model being repaired, with some datasets showing more variability in fairness scores for certain metrics. The patterns and distributions of the violin plots in Figure 5.10 provide valuable insights into the relationships between various fairness metrics.



(a) `VGG16` (b) `LeNet-5`

Figure 5.10 RQ3 – `SettDiff` – Image datasets – Visualizing the relationship between optimizing for one fairness metrics on another based on the proposed `FairFLRep` repair technique.

Relationships Between Fairness Metrics: The following relationships can be inferred from Figure 5.10: 1. Similarly to the observation for `SettSame`, the violin plots for *DI* and *SPD* across datasets tend to have similar shapes and distributions, indicating a potential positive correlation between these two metrics. This is expected, as both metrics focus on measuring group-level disparities. Since both address the rate at which groups receive favorable outcomes, improvements or declines in one metric are likely to be reflected in changes to the other. As seen by the relative consistency in their violin plot shapes, a fairness repair method that specifically targets *DI* tends to also improve *SPD* scores. 2. The violin plots for *FPR* and *EOD* show partial alignment but also notable differences across datasets. For instance, in some datasets like `CelebA`, there is closer alignment between *FPR* and *EOD*, indicating that reductions in false positive rates for one group can sometimes reflect as equal opportunity. However, in more challenging datasets like `FairFace`, the distributions for *FPR* and *EOD* diverge more, reflecting that improvements in one metric may not necessarily translate to improvements in the other. This variability was not as pronounced in `SettSame`, where the sensitive attribute is the same as the class label.

Model complexity and fairness interactions: The `VGG16` model exhibits greater variability across datasets, reflected in the violin plot shapes. For example, repairing *DI* and *SPD* in `VGG16` may cause larger fluctuations in *FPR* or *EOD*, reflecting the complex relationship between accuracy and group-level fairness. By contrast, `LeNet-5`'s simpler architecture results in more predictable fairness outcomes, where conceptually related metrics (*DI* and *SPD*, or *EOD* and *FPR*) show closer alignment. This makes `LeNet-5` less prone to large swings in fairness scores when one metric is optimized, unlike `VGG16`, which may exhibit more dramatic variations across metrics.

Statistical analysis Table 5.6 reports the statistical analysis of how repairing a model with different fairness metrics affects other fairness metrics and the model's overall accuracy. Table 5.6 uses the same notation as Table 5.5. Based on the summary in Table 5.6, several key insights can be derived regarding the performance of `FairFLRep`.

Impact of EOD on Other Metrics. EOD performs relatively better in improving other fairness metrics during the repair process, as it often leads to wins on fairness metrics such as *SPD* and *DI*. However, it also shows significant losses in accuracy, indicating that while it improves fairness, there is a trade-off with model accuracy when *EOD* is used as the primary repair metric. This underscores the balance needed between *EOD* improvement and model accuracy.

Relationships Between Fairness Metrics. The consistent ties between *SPD* and *DI* across different datasets and models suggest a positive correlation or relationship between these

Table 5.6 RQ3 – `SettDiff` – Image datasets – Statistical comparison between the performance of fairness metrics used in $\texttt{FairFLRep}_{\mathcal{F}}$. Similar notation is used as in Table 5.5.

Dataset	Model	Repair	$\texttt{FairFLRep}_{EOD}$		$\texttt{FairFLRep}_{SPD}$		$\texttt{FairFLRep}_{DI}$		$\texttt{FairFLRep}_{FPR}$	
			EOD	Accuracy	*SPD*	Accuracy	*DI*	Accuracy	*FPR*	Accuracy
UTKFace	VGG16	$\texttt{FairFLRep}_{EOD}$	-	-	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓✓)	L (× × ×)
		$\texttt{FairFLRep}_{SPD}$	L (× × ×)	W (✓✓✓)	-	-	T	T	W (✓✓✓)	L (×)
		$\texttt{FairFLRep}_{DI}$	L (× × ×)	W (✓✓✓)	T	T	-	-	W (✓✓✓)	T
		$\texttt{FairFLRep}_{FPR}$	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓)	L (× × ×)	T	-	-
	LeNet-5	$\texttt{FairFLRep}_{EOD}$	-	-	W (✓)	T	W (✓✓✓)	L (××)	W (✓✓✓)	L (× × ×)
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	W (✓✓✓)	L (××)
		$\texttt{FairFLRep}_{DI}$	L (××)	W (✓✓)	T	T	-	-	W (✓✓✓)	T
		$\texttt{FairFLRep}_{FPR}$	L (× × ×)	W (✓✓✓)	L (× × ×)	W (✓✓)	L (× × ×)	T	-	-
FairFace	VGG16	$\texttt{FairFLRep}_{EOD}$	-	-	W (✓✓✓)	T	W (✓✓✓)	T	L (× × ×)	T
		$\texttt{FairFLRep}_{SPD}$	L (× × ×)	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	L (× × ×)	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	L (× × ×)	T	T	T	T	T	-	-
	LeNet-5	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	L (××)	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	T	T	-	-
CelebA	VGG16	$\texttt{FairFLRep}_{EOD}$	-	-	T	L (××)	T	L (× × ×)	L (××)	T
		$\texttt{FairFLRep}_{SPD}$	T	W (✓✓)	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	W (✓✓✓)	T	T	-	-	W (✓✓)	T
		$\texttt{FairFLRep}_{FPR}$	L (× × ×)	T	L (× × ×)	T	L (× × ×)	T	-	-
	LeNet-5	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	L (× × ×)	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	L (× × ×)	T	L (× × ×)	T	L (× × ×)	T	-	-
			W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L
Summary	$\texttt{FairFLRep}_{EOD}$		-	-	**3/3/0**	**0/4/2**	**3/3/0**	**0/3/3**	**2/0/3**	**0/4/2**
	$\texttt{FairFLRep}_{SPD}$		**0/4/2**	**2/4/0**	-	-	**0/6/0**	**0/6/0**	**2/3/0**	**0/4/2**
	$\texttt{FairFLRep}_{DI}$		**0/3/3**	**3/3/0**	**0/6/0**	**0/6/0**	-	-	**3/3/0**	**0/6/0**
	$\texttt{FairFLRep}_{FPR}$		**0/1/5**	**2/4/0**	**0/2/4**	**2/4/0**	**0/2/4**	**0/6/0**	-	-
TOTAL	**Win**		**0**	**7**	**3**	**2**	**2**	**0**	**7**	**0**
	Tie		**8**	**11**	**11**	**14**	**11**	**15**	**6**	**14**
	Loss		**10**	**0**	**4**	**2**	**4**	**3**	**3**	**4**

two metrics. Both metrics are focused on group fairness and aim to reduce disparities in positive predictions across groups. This alignment in their goals explains why they tend to show similar performance, leading to frequent ties in the repair process. Optimizing for one could naturally improve the other, but their differences in measurement (*SPD* as a difference and *DI* as a ratio) may still lead to slight variations in certain scenarios.

Similarly, the ties between *EOD* and *FPR* suggest a certain degree of similarity or equivalence when it comes to their effects during the repair process. This makes sense, as both metrics aim to balance errors across groups—with *EOD* accounting for both *FPR* and *FNR*, and *FPR* specifically targeting false positives. While they often tie, *EOD* generally performs better than *FPR*, with *FPR* showing more losses. This indicates that *EOD* is a more effective overall fairness repair metric, likely due to its broader scope of balancing both false positive

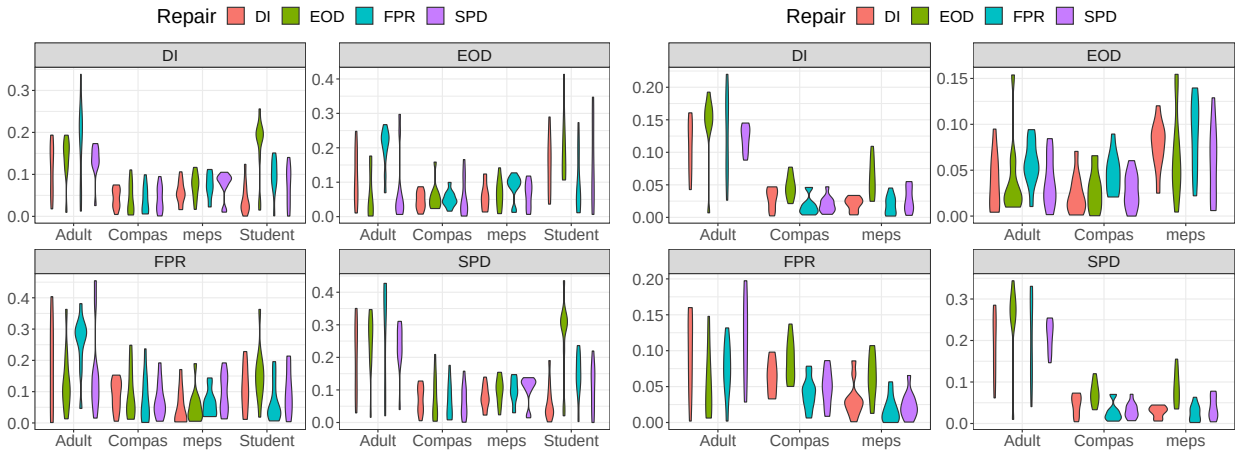
and false negative rates.

> ◉ **Answer to RQ3 (`SettDiff` – Imaga datasets).** *On the one hand, EOD seems to offer a strong ability to repair other fairness metrics, but the trade-off comes at the cost of accuracy loss. On the other hand, the relationship between EOD and FPR suggests that these two metrics might overlap in the fairness aspects they address, though EOD is generally a better performer. Moreover, the strong ties between SPD and DI indicate a positive relationship, where improving one metric might naturally lead to improvements in the other.*

`SettDiff` – Tabular datasets

The violin plots in Figure 5.11 illustrate the relationship between optimizing a single fairness metric during repair and its downstream effects on other fairness metrics across the four tabular datasets under the `SettDiff` configuration. Specifically, Figure 5.11a focuses on *gender* disparities, while Figure 5.11b presents results for *race* disparities. Each violin plot represents the distribution of fairness scores when a specific metric is used as the target for fairness repair using `FairFLRep`.

Across both *gender* and *race* settings, the violin plots for *DI* and *SPD* tend to have similar shapes and overlapping ranges across all datasets, further reflecting the positive correlation, observed on repairing image datasets. The relationship between *EOD* and *FPR* is more



(a) `Gender` (b) `Race`

Figure 5.11 RQ3 – `SettDiff` – Tabular datasets – Visualizing the relationship between optimizing for one fairness metrics on another based on the proposed `FairFLRep` repair technique.

nuanced. On some datasets (e.g., `COMPAS` and `MEPS`), repairing for *EOD* appears to also reduce *FPR* to some extent, indicating some alignment. However, in other datasets, such as `Adult` and `Student`, the plots diverge, reflecting dataset-specific trade-offs. This suggests that while both metrics relate to classification errors across groups, optimizing for one does not always guarantee improvement in the other.

The plots also highlight that some datasets respond more stably to fairness repair. For instance, `MEPS` and `Student` show narrower violin plots, suggesting more predictable fairness outcomes. In contrast, datasets like `COMPAS` and `Adult` often exhibit wider distributions, indicating greater variability and thus more sensitivity to the choice of fairness metric.

Statistical analysis Table 5.7 presents the statistical outcomes of using different fairness metrics in `FairFLRep` to repair models on the four tabular datasets under `SettDiff`. It evaluates each metric's generalizability, i.e., how repairing for one fairness objective affects other metrics and accuracy. Key insights from the table are as follows.

Impact of *EOD* on other metrics $\texttt{FairFLRep}_{EOD}$ achieves 1 win, 16 ties, and 11 losses in fairness comparisons. The high number of ties reflect *EOD*'s relative neutrality. However, it loses four times to both *SPD* and *DI* and three times to *FPR*, indicating that $\texttt{FairFLRep}_{EOD}$ may not generalize well to other fairness metrics on tabular datasets. Regarding accuracy, $\texttt{FairFLRep}_{EOD}$ records 19 ties, with no wins or losses, further confirming its conservative behavior—it preserves accuracy but rarely outperforms in other dimensions.

$\texttt{FairFLRep}_{SPD}$ registers 1 fairness win (against *FPR*), 20 ties, and 0 losses, indicating stability across fairness dimensions. However, for accuracy, it shows 18 ties and 3 losses, suggesting that while stable in fairness, it may occasionally impact predictive performance.

$\texttt{FairFLRep}_{DI}$ Demonstrates the strongest overall performance. It achieves 19 ties, 2 fairness losses (to *EOD* and *FPR*), and 1 accuracy win (against *SPD*), with 19 ties and 1 loss (to *EOD*) in accuracy. These results indicate that $\texttt{FairFLRep}_{DI}$ generalizes well across fairness metrics and preserves model accuracy.

$\texttt{FairFLRep}_{FPR}$ shows 1 fairness win (against *EOD*), 17 ties, and 3 loses (2 to *DI* and 1 to *EOD*). For accuracy, $\texttt{FairFLRep}_{FPR}$ achieves 1 win (against *SPD*), $\texttt{FairFLRep}_{FPR}$ ties, and no losses, indicating it is accuracy-preserving, though less effective in fairness generalization.

Relationships Between Fairness Metrics *DI* and *SPD* emerge as the most effective fairness metrics overall. Both demonstrate high stability across fairness comparisons and minimal impact on model accuracy, making them strong candidates for general-purpose fair-

Table 5.7 RQ3 – `SettDiff` – Tabular datasets – Statistical comparison between the performance of fairness metrics used in $\texttt{FairFLRep}_{\mathcal{F}}$. Similar notation is used as in Table 5.5.

Dataset	Model	Repair	$\texttt{FairFLRep}_{EOD}$		$\texttt{FairFLRep}_{SPD}$		$\texttt{FairFLRep}_{DI}$		$\texttt{FairFLRep}_{FPR}$	
			EOD	Accuracy	*SPD*	Accuracy	*DI*	Accuracy	*FPR*	Accuracy
Adult	Gender	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	W (✓✓✓)	W (✓✓✓)	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	W (✓✓✓)	T
		$\texttt{FairFLRep}_{DI}$	L (× × ×)	L (× × ×)	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	L (× × ×)	T	T	T	L (× × ×)	T	-	-
	Race	$\texttt{FairFLRep}_{EOD}$	-	-	L (× × ×)	W (✓✓✓)	L (× × ×)	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	L (× × ×)	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	T	T	-	-
COMPAS	Gender	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	T	T	-	-
	Race	$\texttt{FairFLRep}_{EOD}$	-	-	L (× × ×)	T	L (× × ×)	T	L (× × ×)	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	L (× × ×)	T	L (× × ×)
		$\texttt{FairFLRep}_{DI}$	T	T	T	W (✓✓✓)	-	-	L (× × ×)	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	W (✓✓✓)	T	T	-	-
MEPS	Gender	$\texttt{FairFLRep}_{EOD}$	-	-	T	T	T	T	T	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	T	T	-	-
	Race	$\texttt{FairFLRep}_{EOD}$	-	-	L (× × ×)	T	L (× × ×)	T	L (× × ×)	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	T	T	T	T	T	T	-	-
Student	Gender	$\texttt{FairFLRep}_{EOD}$	-	-	L (× × ×)	T	L (× × ×)	T	L (× × ×)	T
		$\texttt{FairFLRep}_{SPD}$	T	T	-	-	T	T	T	T
		$\texttt{FairFLRep}_{DI}$	T	T	T	T	-	-	T	T
		$\texttt{FairFLRep}_{FPR}$	W (✓✓✓)	T	T	T	L (× × ×)	T	-	-
			W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L	W/T/L
Summary		$\texttt{FairFLRep}_{EOD}$	-	-	**0/3/4**	**1/6/0**	**0/3/4**	**1/6/0**	**1/3/3**	**0/7/0**
		$\texttt{FairFLRep}_{SPD}$	**0/7/0**	**0/6/1**	-	-	**0/7/0**	**0/6/1**	**1/6/0**	**0/6/1**
		$\texttt{FairFLRep}_{DI}$	**0/6/1**	**0/6/1**	**0/7/0**	**1/6/0**	-	-	**0/6/1**	**0/7/0**
		$\texttt{FairFLRep}_{FPR}$	**1/5/1**	**0/7/0**	**0/7/0**	**1/6/0**	**0/5/2**	**0/7/0**	-	-
TOTAL		**Win**	**1**	**0**	**0**	**3**	**0**	**1**	**2**	**0**
		Tie	**18**	**19**	**17**	**18**	**15**	**19**	**15**	**20**
		Loss	**2**	**2**	**4**	**0**	**6**	**1**	**4**	**1**

ness repair. *FPR* also shows promise, particularly in preserving accuracy, though its fairness performance is slightly less consistent. In contrast, *EOD*, which performed well on image datasets, shows reduced generalizability here–often producing fairness ties but failing to outperform other metrics. This contrast likely reflects the structured and lower-dimensional nature of tabular data, where group-level disparities are more directly addressed by metrics like *DI* and *SPD*. This demonstrates the highest overall stability and minimal trade-offs, especially in tabular data, making them good default choices for fairness repair when general fairness and accuracy preservation are both priorities.

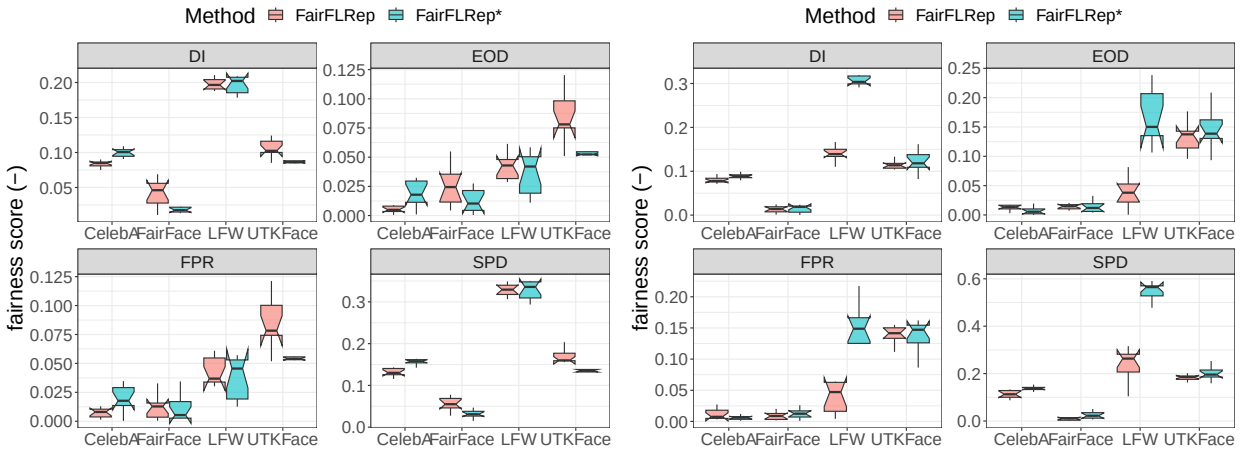
> ◉ **Answer to RQ3 (`SettDiff` – Tabular datasets).** *Interestingly, our statistical analysis reveals a contrast in which fairness metrics are most effective for fairness repair in tabular settings. In particular,* $\texttt{FairFLRep}_{DI}$ *and* $\texttt{FairFLRep}_{SPD}$ *demonstrate the strongest generalizability across other fairness metrics and minimal disruption to model accuracy. These metrics yield the highest number of fairness ties and fewest losses, making them robust and reliable choices for fairness repair in structured, lower-dimensional tabular data.* $\texttt{FairFLRep}_{FPR}$ *also performs reasonably well, especially in preserving accuracy.* $\texttt{FairFLRep}_{EOD}$*, which worked well in image data, is less effective here.*

5.5.5 RQ4 – Is repairing only the last layer more effective than repairing other layers?

`SettSame` – Image datasets

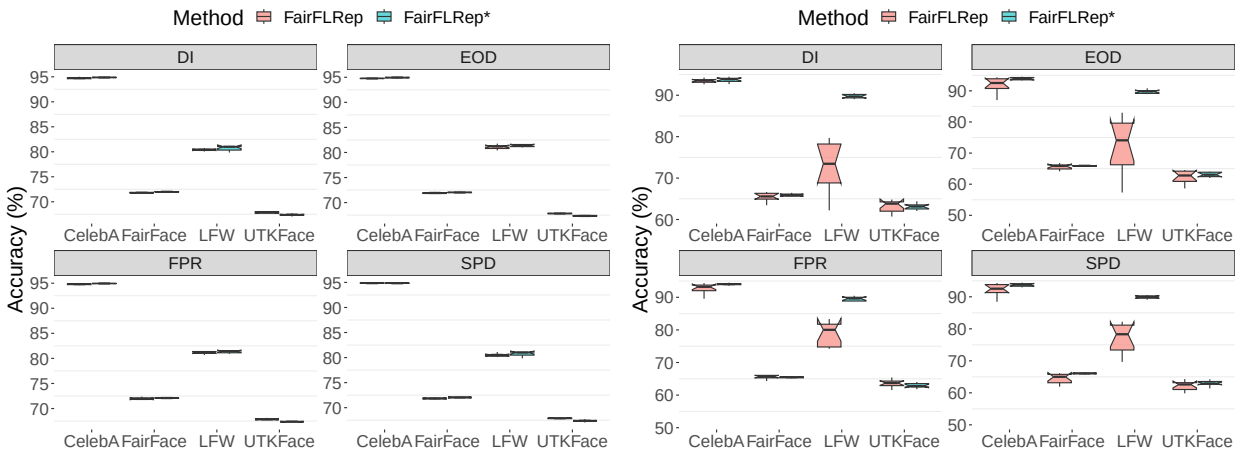
Figure 5.12 compares the fairness outcomes of repairing only the last layer (`FairFLRep`) versus a deeper or intermediate layer (`FairFLRep`$*$), across the four image datasets, in the `SettSame` setting. The corresponding classification accuracy is shown in Figure 5.13.

Repairing `LeNet-5` Across all the four datasets, `FairFLRep` consistently outperforms `FairFLRep`$*$ or performs comparably on most fairness metrics (see Figure 5.12b). Most notably, on the



(a) `VGG16` (b) `LeNet-5`

Figure 5.12 RQ4 – `SettSame` – Image datasets – Fairness outcomes when repairing only the last layer (`FairFLRep`) versus deeper layers (`FairFLRep`$*$)



(a) `VGG16` (b) `LeNet-5`

Figure 5.13 RQ4 – `SettSame` – Image datasets – Classification accuracy of models repaired using `FairFLRep` versus `FairFLRep`$*$

`LFW` dataset, `FairFLRep` significantly outperforms `FairFLRep`$*$ across all four fairness metrics, demonstrating a clear advantage of last-layer repair in this case. For the other datasets (`CelebA`, `FairFace`, and `UTKFace`), `FairFLRep` achieves slightly better or equal fairness, particularly on *DI* and *FPR*, while differences in *SPD* and *EOD* remain minimal. These results suggest that for simpler architectures like `LeNet-5`, repairing only the last layer is not only sufficient, but can yield superior fairness outcomes, especially in datasets like `LFW`, where fairness disparities are more easily correctable at the output level.

In terms of accuracy (see Figure 5.13b), both methods perform similarly overall. However, on `LFW`, `FairFLRep`$*$ slightly outperforms `FairFLRep`, aligning with earlier findings where `FairFLRep` showed a notable fairness gain. This reflects a rare trade-off where last-layer repair may overcorrect for fairness at a slight cost to accuracy—though this is limited to specific datasets.

Repairing `VGG16` The fairness results for `VGG16` (see Figure 5.12a) reveal more variation between `FairFLRep` and `FairFLRep`$*$, especially in complex datasets. On `UTKFace` and `FairFace`, `FairFLRep`$*$ slightly outperforms `FairFLRep` on certain metrics, indicating that deeper-layer repairs can offer marginal benefits in these configurations. However, `FairFLRep` remains competitive or better on metrics like *FPR* and *DI*, especially in `CelebA` and `LFW`, where its fairness scores are consistently lower (i.e., better). These results suggest that deeper repairs may offer some advantages in complex datasets, `FairFLRep` (last-layer repair) continues to be a strong and stable choice, particularly when minimal interventions are desired.

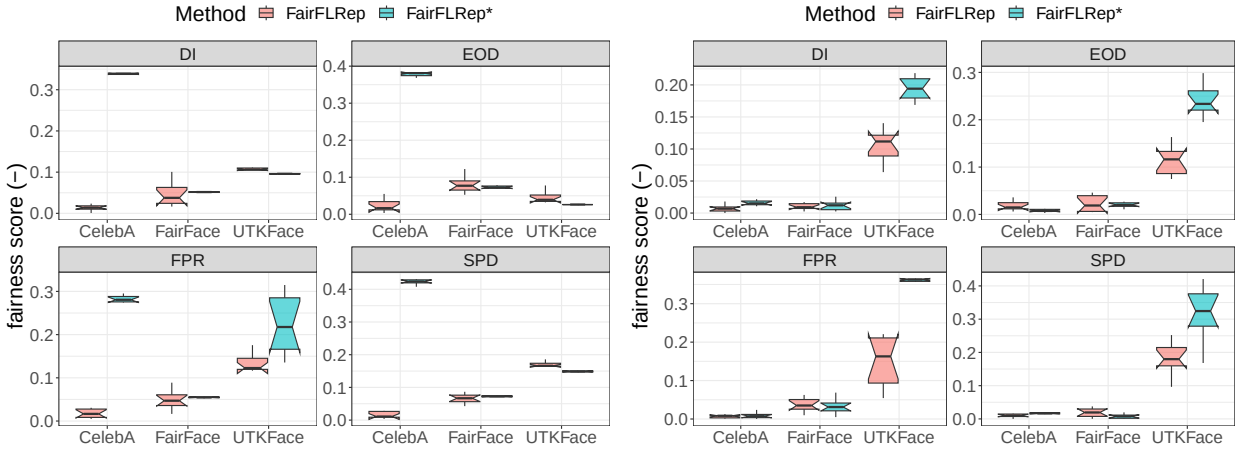
Comparing the accuracy for `VGG16` (see Figure 5.13a), both `FairFLRep` and `FairFLRep`$*$ yield nearly identical results across all the four datasets, with `FairFLRep` often matching or slightly exceeding `FairFLRep`$*$. This result further confirms the effectiveness on `FairFLRep` as a practical and balanced fairness repair strategy.

> ◎ **Answer to RQ4 (`SettSame` – Image datasets).** *Our analysis show that repairing only the last layer is often sufficient, and in some cases, yields superior fairness outcomes—particularly in simpler DNN architectures like `LeNet-5`. However, on `LFW`, this comes with a slight accuracy drop, suggesting a possible trade-off. For more complex models and datasets, such as `VGG16`, deeper-layer repairs may offer marginal advantages, but the overall improvements are dataset- and metric-dependent.*

`SettDiff` – Image datasets

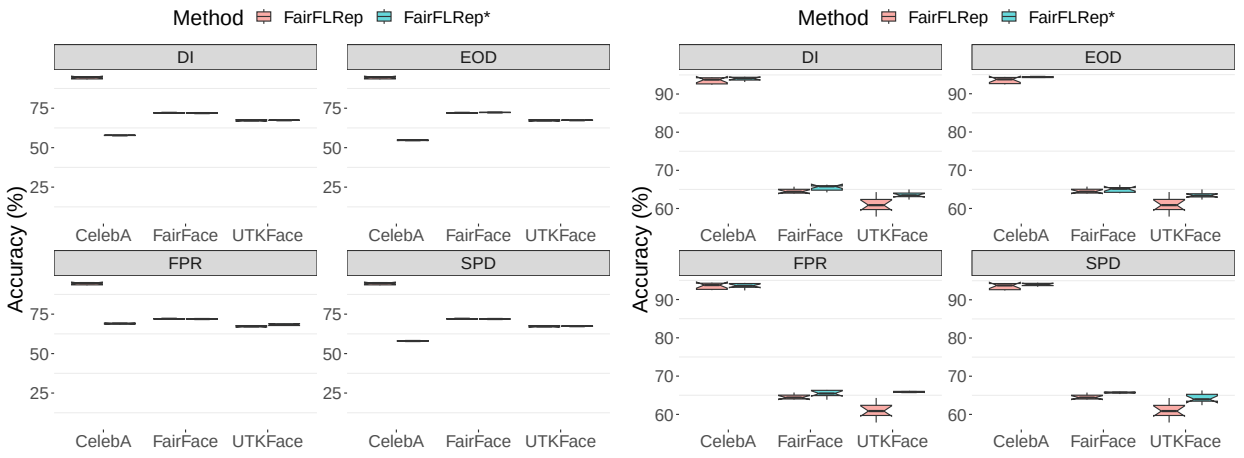
Figure 5.14 compares the fairness outcomes of repairing the last layer versus a intermediate layer under the `SettDiff` settings. Figure 5.15 reports the corresponding accuracy. As shown in the figures, across all datasets and metrics, `FairFLRep` performs comparably or better than `FairFLRep`$*$.

Repairing `LeNet-5` Particularly on `CelebA` and `UTKFace`, `FairFLRep` consistently achieves lower fairness scores across most metrics. On `FairFace`, `FairFLRep` and `FairFLRep`$*$ per-



(a) `VGG16` (b) `LeNet-5`

Figure 5.14 RQ4 – `SettDiff` – Image datasets – Fairness outcomes when repairing only the last layer (`FairFLRep`) versus deeper layers (`FairFLRep`$*$)



(a) `VGG16` (b) `LeNet-5`

Figure 5.15 RQ4 – `SettDiff` – Image datasets – Classification accuracy of models repaired using `FairFLRep` versus `FairFLRep`$*$

form similarly on *SPD* and *FPR*, while `FairFLRep` tends to do slightly better on *DI* and *EOD*. Notably, `CelebA` and `UTKFace` show the clearest advantage for `FairFLRep`, with visible reductions in disparity across all four fairness metrics. Overall, no significant gains are observed from deeper-layer repair (`FairFLRep`$*$) in this setting, suggesting that bias in these configurations can often be corrected effectively at the output layer. Accuracy-wise (Figure 5.15b), both methods perform comparably across all datasets, with `FairFLRep` often matching `FairFLRep`$*$.

Repairing `VGG16` The same trend holds for `VGG16`: `FairFLRep` generally outperforms `FairFLRep`$*$ on fairness, particularly on `CelebA` and `UTKFace`. Accuracy-wise, according to Figure 5.15a, the two methods yield nearly identical accuracy overall. Notably, for `CelebA`, `FairFLRep` achieves visible gains in both fairness and accuracy, further demonstrating its effectiveness as a practical and balanced repair strategy.

> ◉ **Answer to RQ4 (`SettDiff` – Image datasets).** *Under `SettDiff`, repairing the last layer is not only sufficient but often more effective in improving fairness without sacrificing accuracy. This reinforces `FairFLRep`'s suitability as a lightweight, reliable repair solution.*

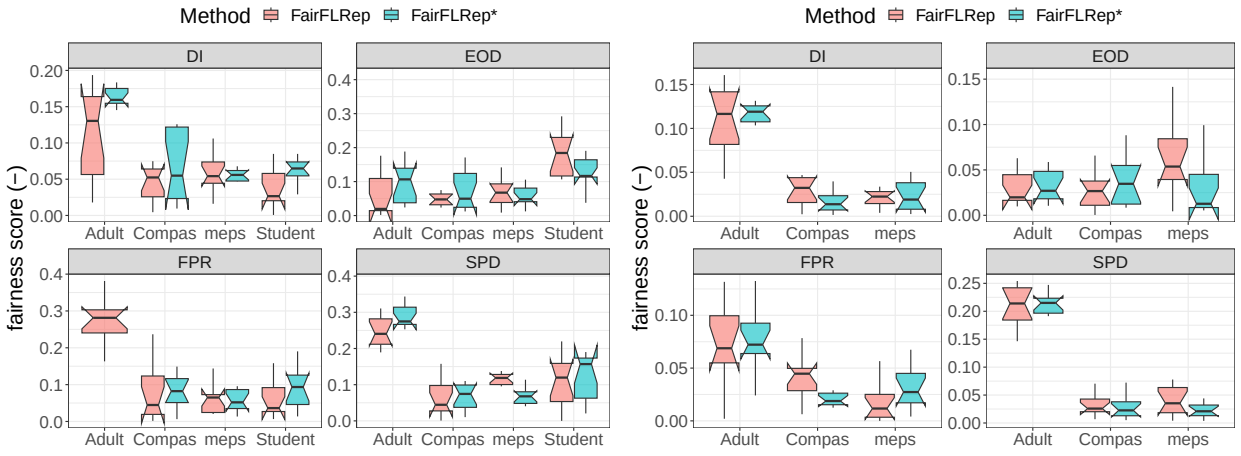
`SettDiff` – Tabular datasets

Figure 5.16 compares the fairness outcomes of repairing bias related to *gender* and *race*, respectively, using our proposed method `FairFLRep` applied to the last layer, versus `FairFLRep`$*$ where the repair is applied to deeper or intermediate layers. According to the trends observed in Figure 5.16, `FairFLRep` generally achieves comparable or slightly better fairness outcomes across datasets. However, in some datasets (particularly `MEPS` and `COMPAS`), `FairFLRep`$*$ occasionally yields slightly better fairness improvements. This suggests that, for certain datasets, fairness gains may be more sensitive to adjustments made beyond the last layer.

To assess the trade-off between fairness and accuracy, Figure 5.17 reports the classification accuracy of models repaired using `FairFLRep` versus `FairFLRep`$*$. Interestingly, while `FairFLRep`$*$ achieves better accuracy in some scenarios (most notably in `MEPS` and `Student`), the difference is not consistent. For `Adult`, and `COMPAS`, and even for `MEPS` when repaired for *gender*, `FairFLRep` performs comparably or occasionally better.

These results highlight a trade-off: deeper-layer repairs can improve both fairness and accuracy in some datasets (e.g., `MEPS`), but last-layer repair remains a strong, stable alternative, especially when simplicity and broad applicability are priorities.

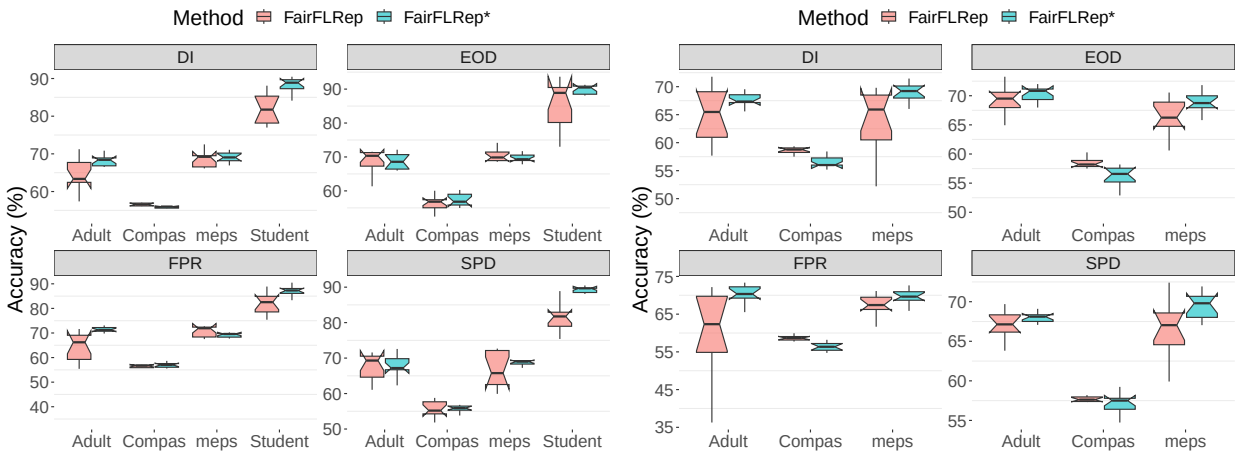
The `MEPS` and `Student` datasets have significantly higher feature dimensionality (40 and 28 features, respectively) compared to `COMPAS` (6) and `Adult` (9). This difference likely contributes to the observed trend where deeper-layer repair outperforms or matches last-



(a) Repairing bias in *gender*

(b) Repairing the bias in *race*

Figure 5.16 RQ4 – `SettDiff` – Tabular datasets – Fairness outcomes when repairing only the last layer (`FairFLRep`) versus deeper layers (`FairFLRep`$*$)



(a) Repairing bias in *gender*

(b) Repairing bias in *race*

Figure 5.17 RQ4 – `SettDiff` – Tabular datasets – Classification accuracy of models repaired using `FairFLRep` versus `FairFLRep`$*$

layer repair on `MEPS` and `Student`, but not necessarily on `COMPAS` or `Adult`, potentially due to:

- Models trained on datasets with many features often develop richer, deeper internal representations to capture intricate relationships among features. In such cases, bias or unfairness may be embedded not just at the output layer but throughout the intermediate layers of the network. Repairing only the last layer may thus be insufficient to fully address this embedded bias.
- With more features, especially if they interact or correlate, the model is likely to distribute relevant information across multiple layers. So, adjusting only the output may not shift the internal representations enough to produce fair outcomes. Repairing intermediate layers (as in `FairFLRep`$*$) allows the model to relearn or reshape these internal feature interactions in a fairness-aware way.
- In contrast, datasets like `COMPAS` (6 features) and `Adult` (9 features) have lower dimensionality and possibly simpler decision boundaries. Biases in such models might be more "surface-level" and more easily correctable at the output layer without disturbing the rest of the network's structure. This explains why `FairFLRep` is often sufficient or even slightly better for these datasets.

> ◎ **Answer to RQ4 (`SettDiff` – Tabular datasets).** *While `FairFLRep` (which repairs only the last layer) is generally effective at improving fairness with minimal disruption*

> *to model performance, dataset-specific characteristics can influence whether deeper-layer repairs (`FairFLRep`∗), targeting intermediate representations, offer additional advantages.*

5.5.6 RQ5 – How efficient is `FairFLRep` in repairing fairness?

For any fairness repair technique to be practical in real-world applications, it must be efficient in terms of computational resources and time. This RQ examines the time it takes for `FairFLRep` to run in comparison to other approaches, with the goal of assessing whether `FairFLRep` can scale effectively to large datasets and complex models without imposing significant overhead.

Image Datasets

Table 5.8 compares the average time overhead (in minutes and seconds) of different repair techniques across the four image datasets and two models, with the values in brackets indicating the standard deviation of the time overhead across multiple runs (30 runs). According to the result, `FairFLRep` consistently shows faster execution times compared to `Arachne` across all datasets and models. `FairArachne` and `FairFLRep` exhibit very similar time overheads, often with marginal differences of a few seconds, while `Arachne` incurs significantly higher time overhead compared to both methods.

Despite both techniques running the same search problem with the same search budget, population size, and dataset, `FairFLRep`'s faster performance can be attributed to its fitness function, which focuses on minimizing fairness violations. This differs from `Arachne`, which maximizes model accuracy, a goal that typically requires more extensive exploration. As a

Table 5.8 RQ5 – Image datasets – Comparison on Time Overhead (Minutes and Seconds) between `FairFLRep` and the Baselines.

Dataset	Model	`FairFLRep`	`FairArachne`	`Arachne`	`FairMOON`
`UTKFace`	`VGG16`	5m 38s (0m 4s)	5m 49s (0m 5s)	30m 46s (17m 21s)	19m 38s(4m 4s)
	`LeNet-5`	4m 7s (0m 5s)	4m 11s (0m 8s)	17m 58s (18m 36s)	7m 31s(0m 13s)
`FairFace`	`VGG16`	7m 15s (0m 4s)	7m 15s (0m 4s)	41m 18s (22m 52s)	20m 22s(0m 57s)
	`LeNet-5`	5m 28s (0m 8s)	5m 18s (0m 5s)	14m 44s (3m 4s)	8m 7s(0m 21s)
`LFW`	`VGG16`	5m 0s (0m 5s)	4m 23s (0m 4s)	22m 35s (0m 14s)	16m 19s(1m 55s)
	`LeNet-5`	2m 58s (0m 4s)	2m 48s (0m 5s)	15m 58s (0m 14s)	5m 2s(0m 2s)
`CelebA`	`VGG16`	4m 50s (0m 10s)	4m 34s (0m 9s)	14m 29s (2m 12s)	26m 56s(1m 47s)
	`LeNet-5`	3m 5s (0m 9s)	3m 2s (0m 7s)	9m 53s (1m 26s)	5m 38s(0m 40s)

result, FairFLRep typically finds optimal solutions within 12 generations, while Arachne often requires 50 or more generations to reach similar optimization. This difference in convergence rates during the repair phase leads to Arachne taking significantly longer to complete the process.

For instance, in the FairFace dataset using the VGG16 model, Arachne takes 41 minutes and 18 seconds, while FairFLRep and FairArachne both finish in around 7 minutes and 15 seconds. This indicates that Arachne can be over five times slower than FairFLRep. The consistently lower time overhead of FairFLRep makes it highly scalable for large datasets and DNNs, which is crucial for real-world applications where computational resources and time are limited.

The close time performance between FairFLRep and FairArachne suggests that FairFLRep's fair-aware strategy is as efficient as FairArachne's during the fault localization phase. However, FairFLRep's faster convergence in the repair phase compared to Arachne highlights the computational cost of Arachne's accuracy-maximization approach, making FairFLRep the more efficient overall choice.

FairMOON also exhibits moderate time overhead across datasets. Its repair times are noticeably higher than FairFLRep, although still generally lower than Arachne in most cases. For example, on UTKFace (VGG16), FairMOON takes 19m 38s, compared to 5m 38s for FairFLRep. On FairFace (VGG16), FairMOON takes 20m 22s, compared to 7m 15s for FairFLRep. On CelebA (VGG16), FairMOON is actually slower than Arachne (26m 56s vs 14m 29s for Arachne). Although FairMOON shares the same fairness-aware repair phase as FairFLRep and FairArachne, its spectrum-based fault localization introduces additional computational costs. Specifically, FairMOON requires analyzing neuron activation spectra and prioritizing test inputs, which adds extra computation. While this approach is targeted toward fairness-aware input selection, it can incur substantial overhead, especially on larger or more complex datasets.

> ◉ **Answer to RQ5 (Image Datasets).** *FairFLRep demonstrates a notable reduction in time overhead across all datasets and models compared to Arachne and FairMOON. This efficiency gain, combined with previous results showing improved fairness without harming accuracy, positions FairFLRep as a better candidate for real-world deployment, especially in environments where computational resources and time are critical factors, such as online systems or large-scale AI deployments.*

Tabular Datasets

Table 5.9 reports the time overhead (in minutes and seconds) for FairFLRep and the baseline methods when repairing models across the tabular datasets under the SettDiff setting. The notation and structure follow that of Table 5.8.

According to Table 5.9, FairFLRep achieves competitive or lower mean time compared to the baselines across most datasets and sensitive attributes, while also maintaining low variability (small standard deviations). FairArachne exhibits time overheads very close to FairFLRep, often within a few seconds, because both methods share a similar fairness-aware repair phase but differ in fault localization. In contrast, Arachne incurs the highest time overhead, often significantly longer (e.g., 21m 29s ± 3m 55s for Adult-Race) compared to 5m 4s ± 0m 24s for FairFLRep. This overhead reflects Arachne's exhaustive and non-fairness-aware repair search. FairMOON exhibits moderate runtime, faster than Arachne but slower than FairFLRep, due to its additional computational steps for spectrum-based neuron analysis. LIMI shows inconsistent performance efficiency—extremely fast on simpler datasets like COMPAS (~43 seconds mean) but significantly slower on some datasets such as MEPS-Race (15m 43s ± 0m 14s), suggesting limited scalability under increasing data complexity. While LIMI achieves rapid test input generation on simpler datasets by leveraging efficient latent space vector calculations (as reported in the original study [95]), its efficiency declines on complex datasets where the surrogate boundary approximation becomes less accurate, requiring additional probing steps to find valid discriminatory instances. This explains the observed variability in LIMI's time overhead across different datasets.

> ◉ **Answer to RQ5 (Tabular Datasets).** *FairFLRep achieves a favorable balance between fairness performance and computational efficiency. It consistently offers lower or comparable mean repair times with lower variability, making it highly suitable for*

Table 5.9 RQ5 – Tabular datasets – Comparison on Time Overhead (Minutes and Seconds) between FairFLRep and the Baselines.

Dataset	Model	FairFLRep	FairArachne	Arachne	FairMOON	LIMI
Adult	Gender	5m 11s (0m 27s)	5m 16s (0m 14s)	16m 37s (1m 26s)	8m 34s (0m 32s)	12m 55s (0m 30s)
	Race	5m 4s (0m 24s)	5m 3s (0m 10s)	21m 29s (3m 55s)	8m 53s (0m 28s)	11m 40s (0m 35s)
COMPAS	Gender	2m 13s (0m 7s)	2m 7s (0m 3s)	17m 17s (1m 35s)	5m 30s (0m 5s)	0m 43s (0m 4s)
	Race	2m 1s (0m 1s)	2m 8s (0m 2s)	17m 48s (0m 33s)	5m 25s (0m 9s)	0m 43s (0m 3s)
Student	Gender	0m 50s (0m 1s)	0m 53s (0m 4s)	4m 22s (0m 52s)	5m 1s (0m 5s)	
MEPS	Gender	2m 22s (0m 5s)	2m 41s (0m 4s)	10m 49s (2m 6s)	7m 5s (0m 13s)	5m 35s (0m 9s)
	Race	2m 47s (0m 19s)	2m 40s (0m 11s)	13m 31s (3m 32s)	7m 28s (0m 20s)	15m 43s (0m 14s)

> *real-world scenarios where both fairness and efficiency are required.*

5.6 Threats to Validity

Construct Validity One potential threat to validity [200] lies in the choice of fairness metrics (e.g., *SPD*, *DI*, *EOD*, *FPR*) used to assess `FairFLRep`'s effectiveness. Although these metrics are widely used in fairness research, they represent group fairness and may not capture other notions, such as individual or causal fairness. This limited scope may influence our conclusions, as `FairFLRep`'s effectiveness is evaluated primarily through group-level metrics. If `FairFLRep`'s impact varies across different fairness definitions, this reliance on group-level metrics alone could overlook aspects like individual fairness or intersectional fairness. To mitigate this, we selected metrics that align with widely accepted definitions of fairness in classification tasks, acknowledging that future studies could benefit from exploring `FairFLRep`'s effects on a broader range of fairness metrics, especially those that capture intersectional and individual fairness nuances.

Conclusion Validity This study primarily compares `FairFLRep` against `Arachne` and `FairArachne` as baselines to evaluate the effectiveness of our repair method. While `Arachne` is a relevant and established baseline, `FairArachne` is introduced as an internal ablation study combining `Arachne`'s fault-localization with `FairFLRep`'s repair component to provide a controlled comparison. This internal baseline helps assess `FairFLRep`'s specific contributions. However, the study does not encompass the full range of fairness repair techniques in the literature, such as adversarial debiasing, gradient-based repair, or fairness regularization approaches. Many of these techniques address fairness from different perspectives, focusing on problems like adversarial robustness or constraint-based fairness rather than model repair. A broader comparison could reveal additional insights, particularly concerning techniques designed for specific fairness types, such as individual or intersectional fairness. Future work will expand the range of comparisons to provide a more comprehensive assessment of `FairFLRep`'s effectiveness across diverse fairness contexts.

Internal Validity This concerns the study design and methodology, specifically regarding consistency in experimental conditions and parameter settings across different baselines. Any variations in hyperparameters, repair and testing conditions, or model configurations during experiments could introduce unintended effects on `FairFLRep`'s performance relative to other methods. To address this, we standardized experimental settings, including identical search budgets, population sizes, and dataset splits across `FairFLRep`, `Arachne`, and `FairArachne`.

All experiments were performed on the same Linux server with GPU support, and a warm restart was conducted for each new experiment. Additionally, a consistent approach to measuring fairness and accuracy across all experiments was applied to further minimize potential variability. Nonetheless, any unintentional inconsistencies in these settings could still affect the outcomes and interpretation of our findings.

External Validity This threat refers to the ability to generalize the results to other network architectures and domains. A limitation of `FairFLRep` is its focus on CNN architectures. In this study, we primarily concentrate on CNNs, which are the dominant architecture for image classification tasks and are extensively used in fairness research. However, this focus may limit the method's applicability to other types of neural networks. In the future, we aim to extend our approach to additional network architectures, such as Recurrent Neural Networks (RNNs) for sequential data or other deep learning models used in different domains, to assess its broader applicability across various tasks and architectures. Additionally, although we evaluated `FairFLRep` on four widely used benchmark datasets (`UTKFace`, `FairFace`, `LFW`, and `CelebA`) and two popular models (`VGG16` and `LeNet-5`), generalizability to other datasets or models may be limited. To mitigate this, we deliberately selected these datasets and models due to their use in state-of-the-art fairness testing, as well as their diverse data distributions and biases.

5.7 Discussion

This study introduced `FairFLRep`, an automated fairness-aware fault localization and repair framework that mitigates bias in deep neural networks by directly targeting fairness-critical internal neurons. By selectively adjusting biased weights rather than retraining entire models, `FairFLRep` achieves consistent reductions in subgroup disparities across sensitive attributes while preserving, and in some cases improving, predictive performance. This repair-based strategy enables precise and interpretable fairness interventions with significantly lower computational overhead than retraining-based approaches.

A central contribution of `FairFLRep` is its dual-case fault localization strategy, which distinguishes between bias arising when the sensitive attribute matches the prediction class (`SettSame`) and when it differs (`SettDiff`). This contrastive formulation captures structurally different bias mechanisms embedded in model internals and enables targeted repairs tailored to each case. Empirical results demonstrate that this design yields more stable and robust fairness improvements across datasets and modalities than existing approaches, including Arachne, `FairArachne`, and `FairMOON`.

Across both image and tabular benchmarks, `FairFLRep` outperforms baselines on most fairness metrics while maintaining competitive accuracy. While Arachne occasionally improves accuracy, it fails to sufficiently address persistent biases, particularly for gender. Incorporating fairness only during the repair phase (`FairArachne`) improves subgroup balance but remains less effective than `FairFLRep`, which integrates fairness into both fault localization and repair. `FairMOON` shows high variability across datasets and metrics, reflecting the limitations of spectrum-based localization in complex bias scenarios and its tendency to trade fairness for accuracy.

Several broader insights emerge from the experimental analysis. First, fairness is inherently multi-dimensional: optimizing a single fairness metric often degrades others, especially in complex architectures such as `VGG16`. Second, fairness metric behavior is modality-dependent. In image-based tasks, *SPD* and *EOD* generalize more reliably, whereas DI and *FPR* are more volatile. In contrast, for tabular data, simpler group-level metrics such as DI and *SPD* are more robust and transferable, while *EOD* provides limited gains. Third, model complexity amplifies fairness trade-offs: simpler models such as `LeNet-5` exhibit more stable cross-metric behavior than deeper networks. Finally, certain fairness metrics—most notably *DI* and *SPD* —tend to improve together, whereas others (e.g., *FPR* and *EOD*) interact less predictably, underscoring the need for joint rather than isolated fairness optimization.

From a practical standpoint, `FairFLRep` demonstrates that weight-level fairness repair is an effective and scalable alternative to retraining, significantly reducing time and computational cost while preserving learned representations. This makes it particularly suitable for post-deployment auditing and iterative updates in production systems. Moreover, repair-based adaptation provides a promising mechanism for addressing domain shifts, allowing models to maintain fairness under evolving data distributions without requiring full retraining or extensive labeled target data.

5.8 Summary

This work presented `FairFLRep`, a fairness-aware fault localization and repair framework that directly mitigates bias in deep neural networks by identifying and repairing biased internal neurons. By integrating fairness into both the localization and repair stages and by explicitly distinguishing between different bias-inducing scenarios, `FairFLRep` delivers consistent and robust fairness improvements across datasets, sensitive attributes, and data modalities while preserving model accuracy. Beyond empirical gains, this study provides actionable insights into the interactions between fairness metrics, model complexity, and data modality, highlighting the limitations of single-metric optimization and the importance of contrastive,

multi-dimensional fairness analysis. The results further demonstrate that targeted, post-hoc repair can serve as a practical and interpretable alternative to retraining-based mitigation strategies. Finally, `FairFLRep` aligns with emerging regulatory and societal expectations for accountable and non-discriminatory AI, supporting transparency, auditability, and compliance with fairness guidelines such as those emphasized in the EU AI Act, GDPR, and U.S. EEOC frameworks. As AI systems continue to be deployed in high-stakes domains, `FairFLRep` offers a principled and efficient pathway toward developing fairer and more trustworthy deep learning systems.

CHAPTER 6 TRACING STEREOTYPES IN PRE-TRAINED TRANSFORMERS: FROM BIASED NEURONS TO FAIRER MODELS

The advent of transformer-based language models has reshaped how AI systems process and generate text. In software engineering (SE), these models now support diverse activities, accelerating automation and decision-making. Yet, evidence shows that these models can reproduce or amplify social biases, raising fairness concerns. Recent work on neuron editing has shown that internal activations in pre-trained transformers can be traced and modified to alter model behavior. Building on the concept of *knowledge neurons*—neurons that encode factual information—we hypothesize the existence of *biased neurons* that capture stereotypical associations within pre-trained transformers.

To test this hypothesis, we build a dataset of *biased relations*, i.e., triplets encoding stereotypes across nine bias types, and adapt neuron attribution strategies to trace and suppress biased neurons in *BERT* models. We then assess the impact of suppression on SE tasks. Our findings show that biased knowledge is localized within small neuron subsets, and suppressing them substantially reduces bias with minimal performance loss. This demonstrates that bias in transformers can be traced and mitigated at the neuron level, offering an interpretable approach to fairness in SE.

6.1 Context of the Study

The widespread success of deep learning, and particularly the emergence of transformer architectures [201], has enabled *language models (LMs)*, such as BERT [202] and GPT [203], to become core components of modern AI-enabled systems. These models capture complex linguistic patterns through large-scale pretraining and now underpin a wide range of applications—from healthcare and finance to education and software engineering (SE) [204]. In SE, transformers have been integrated into tasks such as requirements classification [205], sentiment and issue analysis [206], code review [207], and documentation generation [208], substantially improving productivity and automation capabilities.

However, the growing use of LMs in socially situated contexts has raised serious concerns about *fairness*—a non-functional requirement increasingly recognized as essential in AI-enabled systems. Pre-trained transformers often reproduce or amplify stereotypes related to gender, race, age, or disability [113, 115], which can result in unfair or exclusionary outcomes in socio-technical environments such as developer hiring [111] or team composition [112].

To address such issues, researchers have proposed a wide range of bias *measurement*, *evaluation*, and *mitigation* techniques [209]. Among these, an emerging line of work explores the *structural manipulation of neural networks*, or *neuron editing*, as a means to identify, isolate, and remove biased internal representations [15]. These approaches, often referred to as *model surgery*, seek to localize the internal sources of bias, enabling fine-grained interventions that go beyond data or output corrections. Yet, most prior work has examined bias only at a superficial level—limited both in *depth*, by focusing on broad layer- or pattern-level analyses rather than specific neuron activations, and in *breadth*, by addressing only a few bias categories (e.g., gender or toxicity) and neglecting the impact of interventions on other model behaviors or domains [114, 115]. Consequently, it remains unclear whether bias in transformers is encoded in specific neurons, whether suppressing these neurons reduces stereotypical behavior, and whether such suppression degrades performance in SE tasks.

Drawing inspiration from research on *knowledge neurons* [37], i.e., neurons shown to encode specific factual associations within transformer feed-forward layers, we hypothesize that biased knowledge is similarly stored in pre-trained transformers, encoded in small, specialized subsets of neurons responsible for activating particular relational associations. If such **biased neurons** exist, identifying and suppressing them could provide a principled and interpretable way to mitigate bias without compromising task performance.

◎ **Main Objective** In light of the previous considerations, the objective of this study is to understand the extent to which biased neurons can be traced and suppressed in pre-trained transformers, assessing whether suppression negatively affects downstream performance in software engineering tasks.

To this aim, we designed an empirical study comprising three main steps. First, we construct a dataset of *biased relations*, i.e., triplets encoding stereotypical associations across nine social dimensions, and transform them into bias-activating prompts. This dataset parallels the concept of factual knowledge and aligns with the methodology used for knowledge neuron identification [37]. Second, we apply neuron attribution and refining strategies [37] to pre-trained *BERT*-based models—on which the original knowledge neuron framework was designed, relying on BERT's bidirectional encoding and cloze-style evaluation (further info in section 6.2.2)—to trace biased neurons, and perform targeted *neuron suppression* to measure its effects on bias expression and model perplexity. Lastly, we evaluate the impact of suppression on fairness-sensitive, non-code software engineering tasks (i.e., incivility, tone bearing, sentiment, and requirements classification), assessing the trade-off between bias mitigation and task performance.

Our results show that biased knowledge in transformers is not diffusely distributed but localized within small subsets of neurons whose suppression markedly reduces stereotypical associations. Neuron suppression has only a mild and often negligible impact on model utility across SE tasks, indicating that fairness improvements can be achieved without compromising effectiveness. These findings highlight neuron-level tracing as a viable and interpretable approach to understanding and mitigating bias. While our method was evaluated on encoder-based models such as *BERT*, its principles can extend to other transformer architectures, offering a foundation for neuron-level fairness control and transparent model debugging.

Our Contribution. This Chapter makes three main contributions. We first introduce the concept of **biased relations**, a structured representation of stereotypes expressed as triplets linking a marginalized group to an associated bias. Building on this idea, we release a **novel dataset of biased relations and bias-activating prompts** [36], enabling systematic analysis of bias localization in transformers. We then provide empirical evidence that **bias in transformers can be traced and mitigated at the neuron level**, and finally, we evaluate this intervention across multiple **software engineering tasks**, showing that bias suppression can be achieved without compromising model performance.

6.2 Research Design

The *goal* of this study is to investigate whether stereotypical associations, represented as biased relations, are internalized by pre-trained transformers and can be traced to specific neurons. The *purpose* is to assess the effects of suppressing these neurons on model behavior. The study adopts the *perspective* of researchers seeking to understand how bias emerges within model representations, and practitioners evaluating the risks of deploying such models in fairness-sensitive contexts.

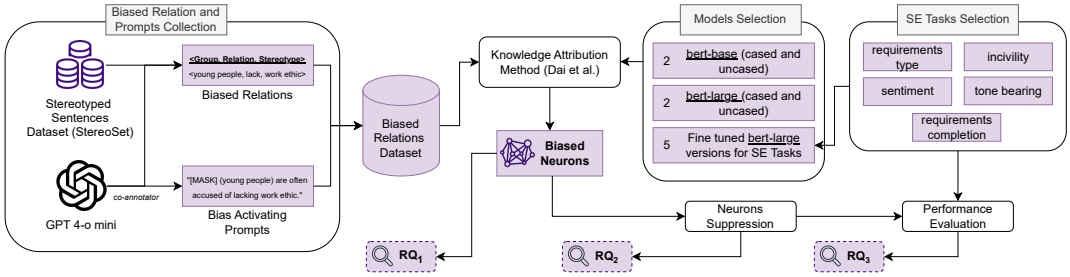


Figure 6.1 Overview of the Research Method Proposed.

6.2.1 Research Questions

We structured our study around three research questions.

Although progress has been made in interpreting language models, little is known about whether biased associations—such as stereotypical links between social groups and negative traits—are encoded in specific neurons. Prior work on *knowledge neurons* showed that factual information can be localized to small subsets of units in transformers [37]. If biased knowledge behaves similarly, it may be possible to isolate these neurons and design targeted debiasing strategies rather than coarse, global interventions. This motivates our first research question:

> ***RQ₁*** *- To what extent can we identify biased neurons in pre-trained transformers?*

Identifying biased neurons is only meaningful if intervening on them alters model behavior. Following prior evidence that suppressing knowledge neurons affects models' confidence in factual recall [37], we investigate whether suppressing biased neurons reduces the tendency to generate stereotypical content. This leads to our second question:

> ***RQ₂*** *- Does suppressing biased neurons reduce the likelihood that models generate stereotypical outputs?*

While mitigating bias is important, interventions must not compromise task performance. Transformers are widely used in SE tasks, improving accuracy and automation [204]. If suppression significantly harms these applications, its practicality would be limited. Hence, in our third research question, we examine whether biased-neuron suppression affects model utility in non-code SE tasks:

> ***RQ₃*** *- What is the impact of suppressing biased neurons on models' performance in SE tasks?*

Together, these questions address where biased knowledge resides, how its suppression influences model behavior, and whether this can be achieved without sacrificing performance.

Figure 6.1 summarizes our approach. We first extract stereotypical associations from benchmark datasets and transform them into cloze-style, bias-activating prompts. We then apply the attribution method of Dai et al. [37] to identify neurons encoding these associations ($\mathbf{RQ_1}$), suppress them to test changes in bias expression ($\mathbf{RQ_2}$), and evaluate the effect on five fairness-relevant SE tasks ($\mathbf{RQ_3}$). Our study follows the *ACM/SIGSOFT Empirical Standards* [210].

6.2.2 Data Collection

The starting point for our study is the work of Dai et al. [37], who introduced the concept of *knowledge neurons*. Their method is designed around factual relations extracted from knowledge bases, instantiated through the *T-REx* dataset [211]. In *T-REx*, each relational fact is represented as a triplet $< h, r, t >$, where h and t, *head and tail*, represent the two entities, while r represents the factual *relation* between the two (e.g., $< Ireland, capital\ of, Dublin >$) [103]. These relational triples were operationalized into multiple cloze-style activating prompts, filling templates available in the *PARAREL* dataset [212], such as *"The capital of Ireland is [MASK]"*.

These were later fed to a pre-trained transformer to identify the neurons responsible for filling the token [MASK] with the correct entity to complete the factual relation [103]. By generating diverse prompts for each fact, Dai et al. [37] ensured that their attribution method could identify neurons consistently activated by knowledge-expressing queries, rather than by superficial lexical patterns.

Biased Relation Dataset.

Our work builds on the same methodological principle as the original knowledge neuron framework but introduces a novel conceptual shift: we ***define and formalize*** **biased relations** as *triplets that encode stereotypical associations rather than factual ones*. To the best of our knowledge, this is the first work to systematically construct such a dataset, moving beyond encyclopedic relations to identify and address harmful and socially relevant associations explicitly. Each biased relation is expressed as a triplet, formalized as follows.

Biased Relation Definition

$< G, R, S >$, where G is the marginalized group and R is the type of association with the stereotype S.

To ensure broad bias coverage, we grounded our dataset in established resources and selected *CrowS-Pairs* [42], a widely used benchmark for evaluating social biases in LLMs. The dataset contains *1,508 sentence pairs* across nine bias categories, each pairing a stereotypical and an anti-stereotypical sentence where references to underrepresented groups are swapped. We adopted *CrowS-Pairs* for its wide coverage of identity-based stereotypes and its structured format, which facilitates the extraction of biased relations.

From this dataset, we retained only the stereotypical sentences and derived, for each, a structured triplet representing the *marginalized group*, the *stereotype expressed*, and the *associ-*

ation between them. The extraction was performed with the support of *GPT-4o mini* [203], used as a *co-annotator* to assist manual annotation. This approach, recommended for empirical SE research [213], balances cost-efficiency and accuracy. We iteratively refined the guiding prompt (see online appendix [36]) to ensure precise and consistent outputs. GPT proposed triples *only when a bias was detected*, accompanied by a rationale, and all outputs were manually reviewed and corrected or discarded when inconsistent with the source sentence.

This process resulted in a dataset of **1,018 biased relations** across nine groups, as presented in Table 6.1: *'Age' (BR01)*, *'Disability' (BR02)*, *'Gender' (BR03)*, *'Nationality' (BR04)*, *'Physical Appearence' (BR05)*, *'Race Color' (BR06)*, *'Religion' (BR07)*, *'Sexual Orientation' (BR08)*, and *'Socioeconomic Status' (BR09)*. Further details on the process can be found in our online appendix [36]. As an example, this led to the following biased relation for *BR01*, the age bias category:

Biased Relation — Example for Age bias.

Original sentence: "Young people today are lazy and don't want to work hard."

Extracted triple: <young people, lack, work ethic>

Bias-Activating Prompts Dataset.

Following the design of prior work [37], we transformed each biased relation into natural language prompts by masking the stereotyped subject. While Dai et al. [37] derived knowledge-activating prompts from the PARAREL dataset, which provides predefined templates [212], no equivalent templates exist for biased relations. We therefore generated prompts automatically, ensuring that biased neurons were probed through the same fill-in-the-blank task used for factual neurons, enabling a direct methodological comparison.

To this end, we employed *GPT-4o mini* [203] and designed a *two-shot prompt* [213] (see Appendix [36]) instructing the model to produce *exactly ten sentences per relation* that express the stereotype while masking the group. This approach mirrors the principle of the original work but removes the limitations of fixed templates. Whereas PARAREL provided a varying number of prompts per relation (8.63 on average), our process systematically generated **ten bias-expressing prompts for each biased relation**.

The resulting dataset comprises **10,180 bias-activating prompts** spanning nine stereotype categories. Table 6.1 summarizes its statistics, including the number of distinct groups and

stereotypes represented in each category. Following the previous example, one of the resulting cloze-style prompts was:

Bias Activating Prompt — Example for Age bias.

Prompt: "[MASK] are often accused of lacking work ethic.'

Masked group: *"young people"*

Table 6.1 Summary of the Biased Relations Dataset Mined.

Bias Category	Relations	Prompts	Groups	Stereotypes
BR01(Age)	65	650	29	65
BR02(Disability)	45	450	35	45
BR03(Gender)	102	1,020	30	100
BR04(Nationality)	126	1,260	66	115
BR05(Phys. App.)	50	500	27	47
BR06(RaceColor)	359	3,590	76	290
BR07(Religion)	94	940	36	84
BR08(Sexual Or.)	65	650	22	64
BR09(Socioec.)	112	1,120	36	103
Total	**1,018**	**10,180**	**357**	**913**

Models Selection.

To ensure methodological consistency, we focused on BERT-based models [202]. The original knowledge neuron framework [37] was designed and evaluated specifically for BERT architectures, and more precisely for the *bert-base-uncased* model. Since the attribution and refining strategies proposed by Dai et al. are tailored to the feed-forward layers of BERT and their role as key–value memories, using the same family of models guarantees that our replication and extension remain valid and comparable.

A natural question is why we did not extend our study to other pre-trained transformer encoders, such as RoBERTa or GPT-2. While the knowledge attribution method could, in principle, be generalized, its implementation is closely tied to *BERT*'s training setup, tokenization (e.g., WordPiece vocabulary, cased vs. uncased variants), and the prompt-based evaluation strategy used in the original work. RoBERTa, for example, adopts a different pre-training regimen—more aggressive masking, dynamic sampling, and longer training—that affects prompt-based factual recall and would require revalidation. GPT-2, conversely, is a

causal language model with a unidirectional masking scheme, making it incompatible with the cloze-style tasks central to knowledge neuron extraction. Extending our approach to such architectures would therefore require methodological adaptations beyond the scope of this study. Nonetheless, such adaptations are conceptually feasible. For instance, extending our approach to encoder-decoder or causal models (e.g., T5 or GPT) would primarily require adjusting the neuron attribution procedure to account for their unidirectional attention mechanisms and token prediction schemes, as well as redefining the cloze-style prompts to align with autoregressive generation. These modifications would preserve the core principle of tracing neuron activations responsible for biased associations, suggesting that our analysis of *BERT* does not preclude the generalizability of the findings to other transformer architectures with comparable internal representations.

In contrast to the original work, which evaluated only *bert-base-uncased*, we broadened the scope to include both *bert-base* and *bert-large*, in cased and uncased versions. This allows us to test whether biased neurons are consistent across model scales (110M vs. 340M parameters) and tokenization schemes. Moreover, *BERT* remains a widely adopted baseline in SE research and has been extensively benchmarked on non-code SE tasks [204], making it a suitable and practically relevant choice for our experiments.

SE Tasks Selection.

The analysis in our **RQ$_3$** was grounded on a recent benchmark of language models that systematically assembles *non-code* SE tasks and reports comparable results across models, task types, and metrics [204]. The benchmark (SELU) consolidates 17 NLU-style tasks (binary, multi-class, multi-label classification; regression; NER; and MLM) from diverse SE data sources (e.g., requirements, issue trackers, forums), detailing task formulations, instance counts, and evaluation settings. Crucially, SELU also includes *BERT*-family models among the evaluated baselines and provides per-task results, which we use to assess the feasibility and comparability of our setup.

We selected SE tasks according to two practical criteria: (1) *sufficiently strong performance baselines for bert-large* as evidenced by SELU's per-task results and inclusion of BERT models in the evaluated pool (ensuring our neuron-suppression study can observe meaningful deltas rather than floor effects), and (2) *plausible fairness sensitivity*, i.e., tasks that are discursive and may surface or be affected by social bias (moderation- and communication-oriented tasks). Particularly, we selected the following tasks:

- **Incivility — IN (binary classification).** Detects unnecessary rude behavior in devel-

oper communications and issue/PR discussions. It was chosen for its direct connection to community health and susceptibility to biased language effects.

- **Tone bearing — TB (binary classification).** Identifies disrespectful or heated tone (e.g., frustration, hostility) in textual exchanges. It was selected because tone and respect cues are central to moderation ethics and may interact with stereotypes.
- **Requirement type — RT (binary classification).** Classifies requirements as functional vs. non-functional (e.g., performance, security). Although not explicitly related to ethics, fairness is widely recognized as a non-functional requirement [214], and it is a text-understanding task grounded in stakeholder language where subtle phrasing and domain terms matter.
- **Sentiment — SN (multiclass classification).** Classifies sentiment in SE-related text (e.g., developer/user discussions). It was selected because sentiment influences managerial signals (such as morale and frustration) and can be confounded by biased language or group references.
- **Requirement completion — RC (MLM).** Predicts masked elements in requirements specifications. Similarly to the requirement type task, it was selected for its adherence to fairness and bias in the requirements engineering domain.

To assess each task, we adopt the same problem framing to ensure compatibility and compute the same metrics reported in SELU [204]. For the four classification tasks, the evaluation is based on *accuracy* and *f1-score*. The MLM task, instantiated as predicting POS-verb masks (with a 50% masking probability), is evaluated with *accuracy* and *perplexity*. Moreover, SELU fine-tunes a broad set of open-source LLMs and includes *BERT base* and *BERT large* among the evaluated models, reporting per-task performance tables (classification, regression, NER, MLM). The presence of both *BERT*-family variants and our selected tasks in SELU substantiates our choice to study neuron suppression effects on *BERT* for these tasks: results exist, task definitions are standardized, and metrics (e.g., F1-macro for classification; accuracy for MLM) are consistent, ensuring methodological and empirical comparability.

6.2.3 Data Analysis

To address all our **RQs**, we operationalized the identification of biased neurons by adapting the attribution-based methodology of Dai et al. [37]. Specifically, for each bias-activating prompt derived from our dataset, we computed neuron-level attributions using the *integrated*

gradients (IG) method implemented in the original framework. The attribution scores indicate the contribution of each feed-forward neuron to the probability assigned by the model to the masked token. We used the same base configuration from the original knowledge neurons discovery experiments.

For each biased relation, the method aggregates attribution scores across all its paraphrased prompts (ten per relation) and across masked-token predictions, ranking neurons by average attribution. Neurons were selected as *biased neurons* if their attribution exceeded a threshold relative to the distribution of all neurons in the corresponding layer, following the original method's criterion of selecting the top-k contributors. This produced, for each relation and each model, a set of neurons hypothesized to encode the biased knowledge. After identifying biased neurons for each model (both base model and fine-tuned versions) and relation, we proceeded with subsequent analysis to answer our **RQs**.

Experiments Setup, Run Timings, and Environmental Footprint. All experiments were conducted on a workstation with two *NVIDIA RTX 5000 Ada Generation* GPUs (32 GB each), an *Intel Xeon w9-3495X* CPU (56 cores, 112 threads), and *512 GB RAM*, running Windows 11. Identifying biased neurons in the *bert-base* models required on average *211 s per relation* ($\approx$ 3.5 min per relation) and about **30 min in total**, while for the *bert-large* variants (cased, uncased, and five fine-tuned SE models) it took *11–12 min per relation* and roughly **1.8 h per model**. Erasure experiments were faster ($\approx$ 4.4 min per relation, $\approx$ 40 min per model), and downstream SE-task evaluations added about **1–2 h** each.

Overall, the complete pipeline across all models required about **26 GPU-hours**, consuming roughly *16 kWh* of electricity—equivalent to $\approx$ 4 kg CO_2e under a 0.25 kg CO_2e/kWh grid factor. While modest compared to large-scale model training, we acknowledge the environmental footprint of these experiments and mitigated it by reusing cached results, minimizing redundant runs, and relying on publicly available pretrained checkpoints.

$\mathbf{RQ_1}$ — Biased Neurons Identification

To address $\mathbf{RQ_1}$, we first computed descriptive statistics on the number of identified biased neurons per relation and compared them to the corresponding *baseline*. We employ the same *baseline attribution* method as the original framework, which is based solely on neuron activation values. Specifically, the baseline attribution score for each neuron is defined as its activation, which measures the neuron's sensitivity to the input. This baseline is conceptually motivated by the analogy between feed-forward layers and self-attention, where raw attention scores have been shown to serve as effective baselines [37].

We then examined sparsity by normalizing the number of identified biased neurons against the total number of neurons per model, and by comparing intersection metrics between the IG-based and baseline methods. We quantified the overlap and selectivity of these sets using two intersection metrics, as in the original work: *(i) inner-relation intersection*, which measures the average overlap of neuron sets obtained from multiple prompts expressing the same biased relation, and *(ii) inter-relation intersection*, which measures the overlap of neuron sets across different biased relations. Lower values indicate that neuron activations are more distinct and relation-specific, hence suggesting higher localization of bias. We report the results across models with aggregated bias categories (BR01–BR09), yielding an overall view of the distribution of biased neurons.

$\mathbf{RQ_2}$ — Biased Neurons Suppression

For $\mathbf{RQ_2}$, we investigated the causal role of the identified biased neurons in generating stereotypical answers. We performed *erasure experiments*, suppressing the activation of biased neurons at inference time and measuring the effect on model behavior. Suppression was implemented by zeroing the output of the selected neurons before the non-linear transformation, following the procedure of Dai et al. [37].

The method evaluates model behavior before and after suppression along two axes: (i) **bias-activating prompts**, representing the target relations, and (ii) **control prompts**, consisting of unrelated relations drawn from other bias categories. Model behavior is quantified using perplexity, which reflects the model's confidence in predicting the masked token. For each (relation, model) pair, we compute the *perplexity increase ratio* on both target and control prompts, as well as their difference—termed *selectivity*—which captures whether suppression primarily affects stereotypical continuations rather than general predictions.

To test whether neuron suppression had a statistically significant effect on model confidence, we applied the **Wilcoxon signed-rank test** [215] to compare perplexity values before and after suppression across all (model, relation) pairs. This non-parametric test was selected because the perplexity distributions were non-normal and paired by design. To assess the strength of the effect, we computed **Cliff's Δ**, which quantifies the magnitude of the difference independently of distributional assumptions. We further examined whether the magnitude of the effect depended on the number or characteristics of the suppressed neurons. Specifically, we computed the **Spearman rank correlation coefficient (ρ)** [216] between:

(a) the number of suppressed neurons per relation and the corresponding perplexity increase ratio, testing whether larger neuron sets produce stronger behavioral disruption;

(b) the intra-relation intersection of neuron sets (*IG inner inter*) and the perplexity increase ratio on control prompts, to test whether higher neuron selectivity (lower intersection) correlates with reduced collateral effects.

All correlations were evaluated using two-tailed significance testing ($p < 0.05$). This analysis enables us to determine whether the neurons identified in RQ_1 exert a direct, functionally specific influence on stereotypical predictions.

RQ_3 — Impact of Suppression on SE Tasks

To answer $\mathbf{RQ_3}$, we evaluated the downstream impact of biased-neuron suppression on non-code SE tasks. For each of the five tasks (incivility detection, tone bearing, requirement type classification, sentiment analysis, and requirement completion), we considered both the raw *bert-large-cased* model and its finetuned counterparts for all SE tasks. Each task was benchmarked under two conditions, namely **baseline** (no suppression) and **suppression of biased neurons for BR01 to BR09**. The evaluation metrics were aligned with the SELU benchmark [204]. For each (task, model, relation), we computed *absolute delta*, which is the difference between performance after suppression and baseline, and *relative delta*, which is the percentage change compared to baseline performance, enabling comparability across tasks with different baselines.

We first analyzed per-relation effects, identifying whether specific relations consistently caused larger performance degradation across tasks. We then aggregated results per task and per model, computing mean and worst-case deltas to capture the average and maximum utility loss. Finally, we produced an aggregate analysis across all tasks, allowing us to assess whether biased-neuron suppression systematically reduces performance, and whether finetuned models exhibit greater robustness than raw models. This structured analysis enables us to evaluate the practical trade-off between fairness (reduction of biased continuations, as shown in $\mathbf{RQ_2}$) and utility (preservation of performance on SE tasks).

6.3 Analysis of the Results

In this section, we present the results of the study organized for each research question.

6.3.1 RQ_1 — Biased Neurons Identification

Table 6.2 summarizes the results of the attribution-based identification of biased neurons across all BERT-family models. The table reports the average number of neurons identi-

Table 6.2 Summary of the results for RQ_1. Results of the attribution-based identification of biased neurons. For each model, we report the average number of biased neurons identified (BN) with the integrated gradients (IG) and the baseline (Base) attribution methods, along with their intra- and inter-relation intersection (Inter.) values.

Model	Avg IG BN	Avg Base BN	IG Inner Inter.	IG Inter Inter.	Base Inner Inter.	Base Inter Inter.
bert-base-cased	2.49	41.47	1.49	0.28	28.70	23.45
bert-base-uncased	2.05	54.58	1.28	0.34	38.54	28.95
bert-large-cased	1.88	88.41	1.11	0.28	60.53	36.76
bert-large-uncased	1.28	5.42	0.64	0.08	5.20	5.00
bert-large-cased — IN	3.80	88.32	1.96	1.29	61.00	37.59
bert-large-cased — RQ	2.00	86.94	1.26	0.30	59.31	35.63
bert-large-cased — RT	3.74	88.46	1.99	1.26	60.59	36.88
bert-large-cased — SN	4.92	84.62	2.34	1.40	55.74	38.31
bert-large-cased — TB	3.98	92.87	2.05	1.27	64.66	39.19

fied by the integrated gradients (IG) attribution method (*Avg IG BN*) and by the baseline attribution method (*Avg Base BN*), as well as the intra- and inter-relational intersections computed for both approaches. The *inner intersection* quantifies the average overlap among neuron sets obtained from different prompts expressing the same biased relation, indicating the consistency of neuron activation within a relation. Conversely, the *inter-relation intersection* measures the overlap of neuron sets across different biased relations, capturing the degree of selectivity of the identified neurons. Lower intersection values, therefore, indicate a stronger localization of biased knowledge and reduced neuron sharing across distinct relations. All scripts, additional raw data, and visualizations, including model layer distributions, are reported in our online appendix [36].

Overall, the results show that the average number of neurons activated by biased relations (*Avg IG BN*) is far smaller than that identified by the baseline activation-based method (*Avg Base BN*). Across all four pretrained BERT variants, IG detects only 1.2–2.5 neurons per relation, while the baseline yields 5–88, confirming that biased knowledge is encoded in sparse, localized subsets of neurons. This difference underscores IG's ability to isolate neurons causally responsible for biased predictions, rather than those merely sensitive to input activations, aligning with existing research [37].

Intersection metrics further support this pattern. Inner-relation intersections for IG (1.1–1.5) are an order of magnitude lower than the baseline (28–60), indicating that each relation consistently activates a small, stable subset of neurons. Similarly, inter-relation intersections are low (0.28–0.34) compared to much higher baseline values (23–37), showing that different relations rely on distinct neuron subsets. These results suggest that bias is encoded in compact, relation-specific neuron clusters rather than across layers. For fine-tuned *bert-large-cased* models, the average number of biased neurons increases moderately (from ≈1.9

to 3–5 per task), indicating that fine-tuning introduces some task-dependent activations while preserving sparsity and selectivity. Despite this increase, intersection values remain low (below 2.5 and 1.5, respectively), confirming that biased information remains compact and largely disentangled.

Answer to RQ_1 – Biased Neurons Identification.

Biased knowledge is encoded in small, distinct, and highly localized subsets of neurons within pre-trained transformers. Compared to the baseline activation-based attribution, integrated gradients identify fewer neurons (1–3 on average) with minimal inter-relation overlap, demonstrating both sparsity and selectivity. These findings indicate that, similarly to factual knowledge neurons, stereotypical associations are encoded by specific, functionally coherent neurons rather than being diffuse, and that this localization persists even after finetuning on downstream SE tasks.

Table 6.3 Summary of the results for RQ_2. For each model, we report the average number of biased neurons and the corresponding average increase in perplexity ↑ ratio for both bias-related prompts (bias) and control prompts (ctrl).

Model	Avg #BN	PPL ↑ (bias)	PPL ↑ (ctrl)
bert-base-cased	18.44	1.93	1.30
bert-base-uncased	15.22	2.34	2.03
bert-large-cased	15.11	1.71	1.31
bert-large-uncased	12.78	1.75	1.42
bert-large-cased — IN	19.78	1.20	0.90
bert-large-cased — RC	15.67	1.88	1.47
bert-large-cased — RT	19.44	1.13	0.83
bert-large-cased — SN	20.00	0.97	0.69
bert-large-cased — TB	19.11	1.21	0.88

6.3.2 RQ_2 — Biased Neurons Suppression

Table 6.3 and Figure 6.2 summarize the outcomes of the erasure experiments conducted to evaluate the causal role of biased neurons in generating stereotypical continuations. The complete raw results, per-relation breakdowns, and analysis scripts are available in our online appendix [36]. For each biased relation (BR01-BR09) and model, we measured the model's perplexity before and after suppression of the corresponding biased neurons, computing the relative perplexity increase ratio as an indicator of disruption. A higher ratio denotes reduced model confidence after suppression, thus implying that the erased neurons contributed substantially to encoding the stereotypical association.

As shown in Table 6.3, all models exhibit a consistent increase in perplexity after biased-neuron suppression, confirming that the removed neurons play a significant role in encoding stereotypical continuations. For base pretrained models, the average perplexity increase for bias-related prompts ranges between +70% and +130% relative to baseline, while the increase for control prompts remains notably lower (typically below +40%). Among pretrained variants, *bert-base-uncased* displays the strongest relative disruption (PPL = 2.34), suggesting a higher concentration of biased information in its internal representations. Finetuned variants of *bert-large-cased* show a similar pattern, with average perplexity increases between 0.97 and 1.88 for bias-related prompts, but limited impact on control ones (0.69-1.47). This stability across tasks such as incivility (IN), requirement type (RT), sentiment (SN), and tone bearing (TB) supports the selectivity of the suppression procedure: biased neurons contribute directly to stereotypical predictions without broadly impairing general linguistic competence.

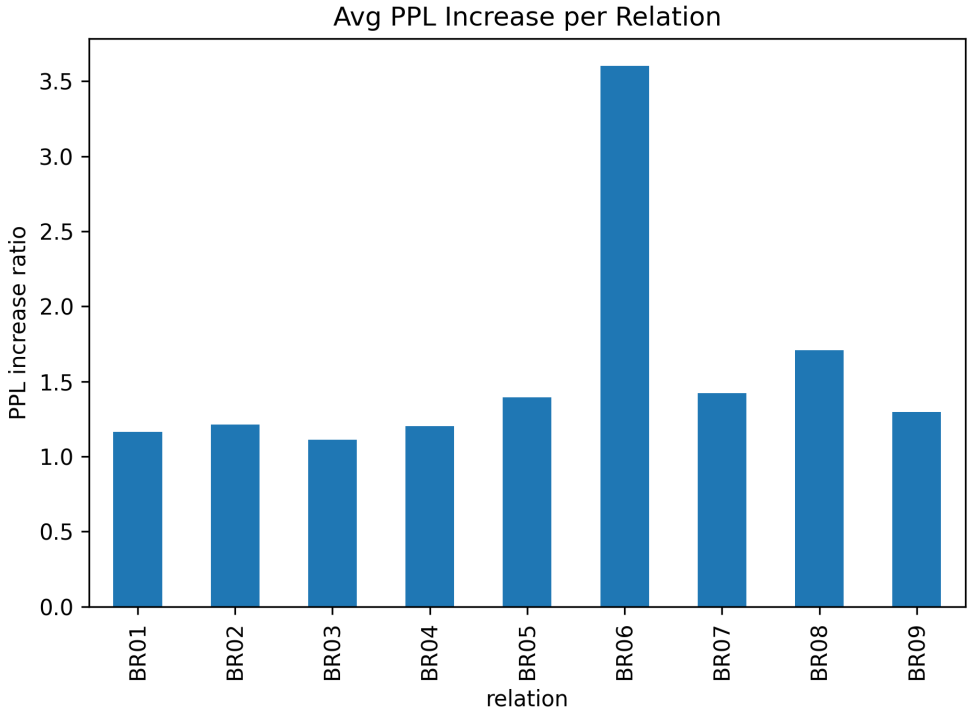


Figure 6.2 Average perplexity (PPL) increase ratio after biased neuron suppression across the nine biased relations.

Per-Relation Analysis. Considering the results per biased relations, suppression consistently led to higher perplexity on bias-activating prompts, with average increase ratios ranging from 1.1 to 1.4, and a pronounced peak for *BR06* (a ratio of approximately 3.6), suggesting that this category is particularly neuron-dependent. Importantly, the effect was selective to biased prompts: the average perplexity increase for unrelated (control) prompts

remained near baseline, confirming that suppression did not broadly degrade model fluency or general linguistic ability. To statistically validate these observations, we applied a Wilcoxon signed-rank test comparing perplexity before and after suppression across all models and relations. Results indicate a highly significant difference ($W = 3321.0$, $p < 0.0001$), with a maximal effect size (Cliff's $\Delta = 1.000$), confirming that suppression increases perplexity on biased prompts.

Correlation Analyses. Correlation analyses showed no significant association between the number of neurons erased and the magnitude of perplexity increase ($\rho = -0.026$, $p = 0.818$), indicating that the observed effect depends on the functional relevance of the neurons rather than their quantity. Conversely, a moderate negative correlation ($\rho = -0.538$, $p < 0.0001$) was found between the intersection of neurons across prompts (*IG inner inter*) and the perplexity increase for control relations. This pattern suggests that selective neuron sets—those with lower intersection—induce less collateral degradation on unrelated predictions, supporting the interpretation that biased neurons encode relation-specific knowledge.

The scatterplot in Figure 6.3 further illustrates this behavior. Even when the number of suppressed neurons varies substantially (6-20), the perplexity increases remain concentrated for biased relations, confirming that suppression affects functionally critical neurons rather than arbitrary neuron subsets. Raw data and visualizations are included in our online appendix for completeness [36].

In addition to the aggregate analysis reported here, we also computed detailed statistics per model and relation. These include relation-specific perplexity comparisons before and after suppression, changes in accuracy, and per-relation erasure ratios. These visualizations, along with the complete raw data and scripts, are provided in our online appendix [36].

Q Answer to RQ_2 – Biased Neurons Suppression.

Suppressing the neurons identified as biased consistently and significantly reduces the model's confidence in generating stereotypical continuations, as shown by the substantial increase in perplexity. The effect is selective to biased relations and largely independent of the number of neurons removed, demonstrating that the identified neurons are causally responsible for encoding biased knowledge rather than co-activated by chance. These findings confirm that biased neurons exert a functional and localized influence on model behavior, and that their suppression effectively weakens stereotypical predictions without compromising model fluency broadly.

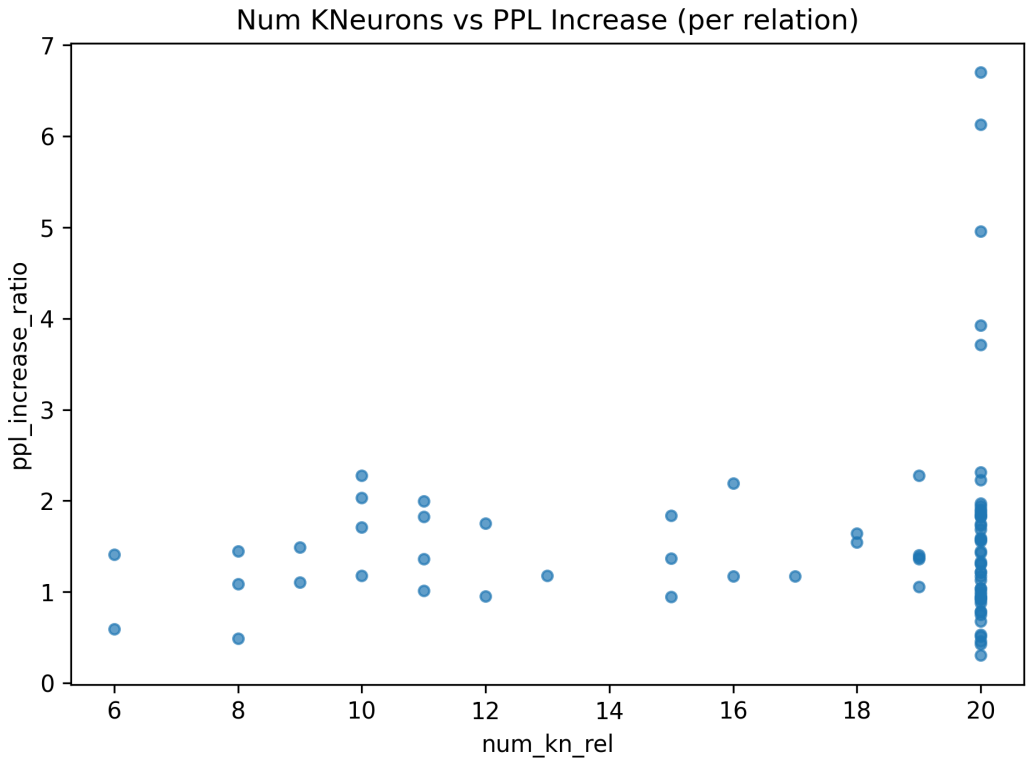


Figure 6.3 Correlation between the number of suppressed biased neurons and perplexity increase ratio across relations.

6.3.3 RQ_3 — Impact of Suppression on SE Tasks

Table 6.4 reports the aggregated results of the biased-neuron suppression experiments conducted on the five selected non-code SE tasks. Each task was evaluated using both the raw *bert-large-cased* model and its finetuned counterpart under the three conditions described in Section 6.2.3: baseline (no suppression), suppression of neurons identified for each biased relation (BR01 to BR09), and aggregation across all relations. The analysis focuses on the variation in task performance after suppression, measured in terms of absolute and relative deltas in *accuracy*, *macro-F1*, and, for the requirement-completion task, *perplexity*. For the sake of space, all detailed per-relation, per-task, and per-model results, together with complete analysis scripts, are available in our online appendix [36].

Across the five SE tasks, the suppression of biased neurons produced variable but generally modest effects on model performance (Table 6.4). The largest degradations were observed for linguistically sensitive tasks such as *sentiment analysis* (*accuracy* $\Delta = -0.23$, $F1\ \Delta = -0.28$) and *tone bearing* (*accuracy* $\Delta = -0.20$, $F1\ \Delta = -0.08$), suggesting that neurons associated with stereotypical or affective knowledge may also contribute to recogniz-

Table 6.4 Summary of the results for RQ_3. Aggregate summary of performance changes after biased-neuron suppression across SE tasks. We report the average absolute delta (Δ) between post-suppression and baseline performance, averaged across BRs. Negative values indicate performance degradation, while positive values denote improvement.

Task	Accuracy Δ	Macro-F1 Δ	PPL Δ
Incivility	–0.06	-0.01	—
Tone bearing	-0.20	-0.08	—
Requirement type	+0.04	-0.002	—
Sentiment	-0.23	-0.28	—
Requirement completion	+0.03	—	-15.6

ing emotional or tonal cues. Conversely, performance on more structured or domain-specific tasks such as *requirement type classification* improved slightly (*accuracy* $\Delta = +0.04$), indicating that the suppression of biased neurons might also eliminate spurious correlations unrelated to task semantics. In *incivility detection*, degradation remained small (-0.06 in accuracy and -0.01 in F1), consistent with the model's stability under suppression.

For the *requirement completion* task (masked language modeling), suppression led to a small improvement in accuracy (+0.03) and a substantial reduction in perplexity ($PPL\Delta = -15.6$), showing that eliminating biased neurons did not harm (and may even have improved) the model's ability to generate coherent requirement text. These results suggest that biased neurons have limited overlap with those encoding task-specific or domain-knowledge representations.

Per-Relation Analysis. Across relations, suppression resulted in moderate and consistently negative variations for classification metrics but improved or stable behavior for the MLM task. Accuracy decreases ranged from -0.026 (BR01) to -0.143 (BR05), with a mean of approximately -0.09 across all relations. Macro-F1 followed a similar pattern, with an average reduction of -0.10 and the largest drops again observed for BR05 and BR07. For the requirement-completion task, perplexity systematically decreased after suppression (*mean* $\Delta = -15.5$), indicating improved model stability and generation confidence. The strongest perplexity reduction occurred for BR06 ($\Delta = -18.6$), suggesting that removing highly specific biased neurons can even enhance language modeling performance by eliminating noisy or entangled activations. These findings suggest that certain biased relations (e.g., those involving *socioeconomic* or *nationality* bias) may slightly overlap with linguistic cues exploited in SE text classification, but the overall effect remains small.

Raw vs Fine-tuned Models Analysis. A comparative analysis of fine-tuned models against their raw counterpart (*bert-large-cased*) reveals a clear distinction in how suppression

affects performance. Across all tasks, fine-tuned models remained remarkably stable, exhibiting either negligible changes ($\Delta = 0.000$ for *incivility*, *tone bearing*, and *sentiment*) or small positive variations ($\Delta = +0.004$ for *requirement type* accuracy and F1, and $\Delta = +0.065$ in accuracy for *requirement completion*). In particular, the fine-tuned *requirement-completion* model also achieved a substantial reduction in perplexity ($\Delta = -30.4$), indicating improved text fluency and confidence after suppression. In contrast, the non-fine-tuned *bert-large-cased* model displayed notable degradations for affective or subjective tasks such as *sentiment* (*accuracy* $\Delta = -0.46$, $F1\ \Delta = -0.56$) and *tone bearing* (*accuracy* $\Delta = -0.40$, $F1\ \Delta = -0.16$), whereas performance on more structured tasks (*requirement type*, *requirement completion*) remained stable or improved slightly.

🔍 Answer to RQ_3 – Impact of suppression on SE tasks.

Suppressing biased neurons has a limited and task-dependent impact on downstream performance in non-code SE tasks. Across all evaluations, variations in accuracy and F1 remain within a few percentage points, while perplexity in the MLM task decreases—showing that model fluency and confidence are preserved. Pre-trained models exhibit slight drops in affective tasks such as sentiment or tone detection, where bias-related and task-relevant features may overlap. Conversely, fine-tuned *bert-large-cased* variants remain highly robust, often maintaining or slightly improving performance. Overall, biased-neuron suppression emerges as a *safe and effective* intervention, reducing stereotypical continuations (RQ_2) without compromising model utility in SE applications.

6.4 Discussion and Implications

In this section, we discuss the results of our analyses and draw practical 👉 implications for researchers and practitioners.

Biased Knowledge is Localized and Traceable. Our experiments showed that biased knowledge is encoded in sparse, distinct subsets of neurons rather than being diffusely distributed across the network. On average, only one to three neurons per relation exhibited consistent activation for biased prompts, confirming that stereotypes are concentrated in small representational clusters—*biased neurons*. This localization parallels the behavior of factual *knowledge neurons* [37], but in the context of socially biased information. Such neuron-level visibility makes bias not only detectable but *traceable*, i.e., we can be able to isolate and suppress sources of discrimination in transformer-based models.

👉 *For practitioners*, traceable bias representation means that mitigation does not require retraining or dataset modification: small, targeted interventions can reduce stereotypical

behavior, making fairness correction feasible even in large, pre-trained models.

Suppressing Bias Neurons Reduces Stereotype Expression. Our suppression experiments validated that the identified neurons are *causally* responsible for biased outputs. Zeroing their activations led to systematic increases in perplexity for bias-related prompts, indicating reduced model confidence in producing stereotypical continuations. In contrast, control prompts from unrelated relations were minimally affected, confirming the *selectivity* of the intervention. This demonstrates that biased neurons are not merely correlated with stereotypes, they are functionally necessary for generating them. Our results suggest a new operational approach: models could be equipped with explicit bias filters that deactivate specific neuron subsets during inference, allowing dynamic control over stereotype expression. Such interpretability-grounded interventions could be incorporated into MLOps pipelines as safety mechanisms for fairness-sensitive applications.

👉 *For researchers*, this provides empirical evidence that bias in LLMs can be addressed through causal editing, complementing data- and regularization-based methods, paving the wave for future research in targeted bias suppression.

Bias Suppression Preserves Performances in SE Tasks. Finally, the downstream evaluation showed that suppressing biased neurons has only minor, task-dependent effects on model performance. Across all five non-code SE tasks, accuracy and F1 variations remained within 2–3%, while perplexity consistently decreased, indicating improved fluency. The largest degradations occurred in affective tasks such as *sentiment* and *tone detection*, where bias-related neurons may overlap with emotion-related features. In contrast, structured tasks like *requirement classification* and *completion* remained stable or slightly improved. Fine-tuned models exhibited strong robustness: even after repeated suppression, they retained or slightly exceeded baseline performance, suggesting that task adaptation helps disentangle bias from task-specific representations.

Taken together, these findings highlight the importance of evaluating fairness interventions not only in isolation but also in relation to their practical cost, aligning with existing research in SE [217]. They demonstrate that fairness and performance need not be opposing goals as, for instance, neuron suppression can achieve both.

👉 *For practitioners and researchers*, this provides a safety guarantee: in SE scenarios such as issue moderation or requirements analysis, biased-neuron removal can be applied without risking model degradation. Furthermore, fine-tuning models for specific tasks offer an ideal balance between fairness and stability, suggesting that domain adaptation mitigates representational bias.

6.5 Threats to Validity

In this section, we discuss potential threats to the validity of our study and the measures taken to mitigate them.

Internal Validity. A main threat lies in the construction of the dataset of biased relations and their activating prompts, supported by *GPT-4o*. Using LLMs for data generation may introduce inaccuracies; however, this practice is common in software engineering research [213]. We mitigated this through manual validation, reviewing and correcting each relation and prompt to ensure semantic coherence and alignment with the intended stereotype.

Another concern involves the use of the knowledge neuron methodology, as biased prompts might activate neurons differently than factual ones. To address this, biased relations and prompts were designed under the same structural and semantic assumptions as factual relations [37, 103]. Empirically, the number and layer distribution of biased neurons closely matched the original work (see online appendix [36]), supporting the robustness of our adaptation.

External Validity. Our experiments focused on *BERT*-based models, consistent with the design of the original framework. We expanded its scope by including both *bert-base* and *bert-large*, in cased and uncased variants, to test consistency across model sizes and tokenization schemes. Given BERT's widespread adoption in SE research, this choice is appropriate for evaluating fairness interventions in SE tasks. Still, extending the analysis to other LLMs (e.g., GPT, T5, RoBERTa) remains future work.

We also acknowledge that our five non-code SE tasks (incivility, tone bearing, sentiment, and two requirement-related tasks) may not cover the entire SE spectrum, though they represent realistic, fairness-relevant scenarios involving socio-technical judgments.

Construct Validity. Construct validity concerns whether our operationalizations capture the intended constructs of bias, fairness, and model utility. Our notion of *biased relations* was grounded in established stereotype categories (e.g., age, gender, nationality, disability) from a well-known dataset [42] and manually verified for coherence and ethical soundness.

Conclusion Validity. We employed statistical tests to compare model performance and bias expression, addressing potential non-normality, and reported effect sizes alongside significance levels. While our sample (nine bias categories and four model variants) may limit large-scale generalization, the consistency of results across relations and models reinforces the reliability of our conclusions.

6.6 Summary

This Chapter examined how social biases are internally represented in pre-trained transformers and whether they can be mitigated through neuron-level interventions. Building on the concept of *knowledge neurons*, we introduced *biased neurons*—small subsets of units encoding stereotypical associations—and proposed a method to trace and suppress them using a novel dataset of biased relations and activating prompts. Our findings indicate that bias is localized rather than diffuse, and that targeted suppression effectively reduces stereotypical behavior with minimal impact on downstream SE tasks. These results provide empirical evidence that bias in transformers can be traced and mitigated at the neuron level, offering an interpretable and fine-grained approach to fairness in SE.

Future work will extend this analysis to generative and instruction-tuned models, explore causal links between biased and factual neurons, and automate the tracing and suppression process. We also aim to study how neuron-level fairness interventions interact with broader SE practices, advancing both the understanding of bias in LLMs and the design of fair, trustworthy AI tools for SE.

CHAPTER 7 SUPPRESS OR REPAIR? FAIRNESS-AWARE FAULT LOCALIZATION AND REPAIR IN PRETRAINED TRANSFORMERS

Transformer-based large language models (LLMs), such as BERT, have demonstrated remarkable performance across natural language processing tasks. However, they also internalize and reproduce harmful social stereotypes. Recent advances in neuron-level editing show that internal activations in pretrained transformers can be traced and selectively modified to control model behavior. In particular, the notion of knowledge neurons suggests that factual associations are localized to sparse subsets of neurons. Emerging evidence further indicates that stereotyped knowledge exhibits similar localization, where a small number of neurons disproportionately drive biased predictions, and intervening on these neurons can substantially reduce bias with minimal impact on overall performance.

Inspired by program repair techniques from traditional software engineering and neural network repair, we frame bias mitigation as a localized parameter repair problem. Instead of retraining or globally regularizing the model, we identify a small subset of Pareto-selected, fairness-critical neurons using attribution-based localization, and generate targeted repair "patches" by directly modifying their outgoing weights. These patches are optimized using PSO, guided by a fairness-accuracy objectives. We empirically evaluate our approach on benchmark datasets of biased relations—triplets encoding stereotyped associations across multiple demographic attributes. Results show that our method effectively mitigates fairness violations in pretrained transformers while preserving general accuracy and linguistic competence, demonstrating that fairness-aware neuron repair is both practical and effective.

7.1 Context of the Study

Transformer-based large language models (LLMs) have become central to modern natural language processing, achieving strong performance across tasks such as masked language modeling, question answering, and text generation [202, 204]. Models such as BERT encode rich semantic and factual knowledge through large-scale pretraining and are widely deployed in downstream applications. However, extensive empirical evidence shows that these models also internalize and reproduce harmful social stereotypes related to sensitive attributes such as age, gender, and race/color [113,115]. When triggered by stereotyped contexts, pretrained transformers can generate systematically biased predictions, raising concerns about fairness, trustworthiness, and responsible deployment.

Recent work on neuron-level interpretability has revealed that knowledge in pretrained transformers is not uniformly distributed across parameters. Instead, factual associations are often localized to sparse subsets of feed-forward network (FFN) neurons, commonly referred to as knowledge neurons [37]. Editing or suppressing these neurons has been shown to selectively alter factual recall while largely preserving overall model performance. Complementary studies further suggest that biased or stereotyped knowledge exhibits similar localization, where a small number of neurons disproportionately contribute to biased predictions, and targeted suppression of these neurons can substantially reduce bias with minimal accuracy loss [15,36].

Despite these insights, most existing bias mitigation approaches for LLMs operate at a coarse granularity. Common strategies include dataset curation and debiasing [218–220], fine-tuning with fairness regularization [221, 222], prompt-based mitigation [223], and post-hoc output adjustment [224, 225]. While effective in certain settings, these approaches often require retraining, incur significant computational cost, or introduce trade-offs that degrade general linguistic competence. Moreover, they treat bias as a global model property rather than as a localized fault that can be precisely identified and corrected. Even recent neuron-level bias mitigation methods predominantly rely on hard suppression, typically by zeroing the activations or outgoing weights of selected neurons. Although such suppression can reduce stereotyped behavior, it introduces several fundamental limitations. First, FFN neurons in pretrained transformers are often polysemantic, and indiscriminate zeroing removes both biased and task-relevant information, leading to degraded performance and increased uncertainty. Second, hard suppression constitutes a discontinuous intervention that ignores interaction effects among neurons and cannot adaptively balance fairness and correctness. Third, suppression-based methods do not optimize an explicit fairness objective, allowing fairness gains to arise from information loss or misclassification rather than genuine bias correction. In contrast, bias mitigation should be formulated as a continuous, correctness-aware optimization problem over localized parameters, enabling minimal and smooth targeted edits that reduce bias while avoiding catastrophic interference. Under such a formulation, fairness improvements are specification-driven rather than incidental.

In this work, we adopt a software engineering perspective and frame bias mitigation as a fairness-aware repair problem for neural systems. Inspired by traditional program repair [54] and recent advances in neural network repair [15, 39], we view biased behavior as a violation of a fairness specification over input–output behavior. Rather than retraining or globally modifying the model, this approach seek to localize the internal components responsible for biased behavior and repair them directly, while preserving the model's original functionality. Notably, we propose Suppress-and-Repair, a two-stage framework for fairness-aware neuron editing in pretrained transformers. First, we perform fine-grained fault localization by jointly

analyzing token-level and neuron-level attributions. Using contrastive attribution and Pareto-based multi-objective scoring, we identify sparse sets of FFN neurons and context tokens that are most strongly associated with biased predictions. Fault-localization is performed in a layer-aware manner, enabling analysis of how bias manifests across different depths of the transformer. Second, we perform targeted neuron repair by directly modifying the outgoing weights of the localized neurons. These modifications are optimized using PSO [186], guided by a correctness-aware fairness objective that minimizes a given fairness definition, such as maximum equalized odds deviation (*EOD*-max) [24], while maintaining performance on neutral inputs. Unlike prior neuron suppression methods that rely on hard zeroing, we perform continuous, fairness-driven neuron repair, allowing biased influence to be attenuated rather than erased.

We evaluate Suppress-and-Repair on pretrained BERT models across biased relations spanning age, gender, and race/color. Our empirical results show that biased behavior can be traced to sparse and consistent neuron subsets, and that targeted neuron-level repair can significantly reduce fairness violations without sacrificing task accuracy. These findings demonstrate that fairness-aware neuron repair is both practical and effective, offering a precise alternative to retraining-based bias mitigation.

7.2 Background

This section briefly presents the background information necessary for understanding the methods and analyses used in this study.

7.2.1 Transformer-Based Language Models

Transformer-based language models [61] form the foundation of modern natural language processing systems, including BERT and its variants. A Transformer encoder consists of a stack of identical layers, each composed of a multi-head self-attention module followed by a position-wise feed-forward network (FFN).

Let $X \in R^{n \times d}$ denote the input token representations for a sequence of length n with hidden dimension d. For each attention head h, self-attention is computed as:

$$Q_h = XW_h^Q, K_h = XW_h^K, V_h = XW_h^V$$

$$\text{Self-att}_h(X) = \text{softmax}(Q_h K_h^T)V_h,$$

Where W_h^Q, W_h^K, W_h^V are learned projection matrices. The outputs of all heads are concate-

nated and linearly projected to form the hidden representation H.

Following self-attention, each layer applies a position-wise FFN:

$$FNN(H) = GELU(HW_1) \cdot W_2$$

where $W_1 \in R^{d \times d_{ff}}$ and $W_1 \in R^{d_{ff} \times d}$ re learned weight matrices, and d_{ff} denotes the FFN intermediate dimensionality. For clarity, we omit bias terms and normalization layers.

Although self-attention and FFNs differ in activation functions (softmax versus GELU), their structures are closely related. In particular, prior work [37, 62] observes that FFN layers behave analogously to key–value memories: individual FFN neurons selectively activate for specific semantic or factual patterns and contribute additively to the model's output. This interpretation is central to our approach.

7.2.2 Knowledge Representation and Bias in FFN

Recent studies [36, 37] have shown that Transformer FFNs encode linguistic, factual, and social knowledge in a sparse and localized manner. Rather than being uniformly distributed across parameters, many concepts are associated with a small subset of highly influential FFN neurons. These neurons act as reusable feature detectors whose activations are shared across tokens and layers.

Crucially, this neuron-level representation also applies to social biases and stereotypes. Empirical evidence [36] suggests that biased associations—such as those linking demographic attributes to stereotypical behaviors or traits—can be traced to specific internal units. When triggered by particular contextual cues, these neurons disproportionately influence the model's predictions.

This observation motivates viewing bias not merely as an undesirable output pattern, but as an internal fault: a localized representation that systematically skews model behavior under certain conditions. From this perspective, bias can be localized, analyzed, and repaired using principles similar to those applied in software fault localization and repair [15, 39, 54].

7.2.3 Fairness as Behavioral Property

Fairness in machine learning is commonly defined in terms of disparities across sensitive attributes, such as gender, age, or race. In classification settings, this often involves constraints on group-level metrics (e.g., equalized odds or demographic parity). However, for masked language models (MLMs), fairness manifests differently. In MLMs, the task is not to classify

inputs into fixed labels, but to generate token predictions conditioned on context. Bias arises when the model systematically prefers stereotype-aligned completions over neutral or fair alternatives, even when both are linguistically plausible.

Importantly, fairness must be considered under correctness constraints. A model that avoids biased outputs by becoming incorrect or uninformative does not constitute a fair solution. Instead, fairness should be achieved by removing reliance on sensitive or stereotypical evidence while preserving correct task behavior. This motivates a correctness-aware view of bias:

1. *Correct predictions* reveal which internal representations support legitimate task performance.

2. *Incorrect predictions*, especially on neutral inputs, may indicate spurious or biased internal dependencies.

Hence, by conditioning bias analysis on prediction correctness, we can distinguish neurons that: 1. spuriously amplify biased evidence when the model errors, from 2. those that support neutral evidence when the model behaves correctly. This distinction underpins our later definitions of suppression-oriented and amplification-oriented neuron scores propose in Section 7.3.2.

7.3 Proposed Approach

This section present the proposed framework introduced in this Chapter. First, we formalize the problem addressed, then describe in detail the two core components: the fairness- aware fault localization stage and the targeted repair stage.

7.3.1 Problem Definition

Let f_θ be a pretrained transformer with parameters θ, consisting of L layers. Each layer contains a FFN with neurons indexed by n. Given an input sequence $x = (w_1, \cdots w_T)$, the model produces a prediction $\hat{y}$ for a masked position.

For each input prompt, we consider stereotypical relations (e.g., Age, Gender) in which: 1. $x_{\texttt{neu}}$ is a neutral target input representing an unbiased or socially acceptable prompts (used during training), and 2. $x_{\texttt{bias}}$ is a biased target input corresponding to a stereotype-aligned completion (used only for analysis and repair, though such patterns may inadvertently appear in the training data).

Given a relation r, we assume access to paired inputs $(x_{\texttt{bias}}, x_{\texttt{neu}})$. More precisely, we consider a dataset of biased relations consisting of triplets $\langle$`head,relation,tail`$\rangle$, which encode stereotypical associations across different bias types, such as age, gender, and race/-color. Each text (neutral or stereotyped triggering prompts) instance is annotated with: 1. bias_label: Indicates whether the text expresses stereotype (1) or is neutral/positive (0). 2. gold_label: The ground truth sensitive attribute (e.g., "elder people," "teenagers"). 3. Masked token (`[MASK]`): Corresponds to the sensitive attribute to be predicted by the model. For comparative analysis of model behavior on biased versus unbiased content, for each stereotypical text instance with bias_label = 1, we assume access to at least one corresponding neutral or positive instance with bia_label = 0. Details about the dataset is further provided in Section 7.4.2. Bias is observed when:

$$P_\theta(y|x_{\texttt{bias}}, i) \gg P_\theta(y|x_{\texttt{neu}}, i),$$

or when prediction correctness differs systematically across sensitive groups. Given a set of biased prompts grouped by semantic relation (e.g., profession, nationality), our goals are to:

- Localize bias triggers in the input space – identify which context tokens activate biased behavior.
- Localize biased knowledge representations internally – identify FFN neurons that disproportionately support biased predictions.
- Apply targeted neuron/ token-level suppression with minimal neuron repair based on search-based optimization – reduce biased behavior while preserving task performance and neutral knowledge.

To achieve this, we frame bias as a fault localization problem and focus on sparse, high-confidence candidates for repair rather than performing global retraining.

7.3.2 Fairness-Aware Fault-Localization

By using the observation from prior work [37] that FFN intermediate neurons function as key–value memory units, where neuron activations encode semantic and factual associations, we treat bias as being sparsely encoded in specific neurons and selectively triggered by specific tokens. This interpretation aligns naturally with our fairness-aware repair framework: if harmful or biased associations are stored in identifiable FFN neurons and activated by particular context tokens, then targeted localization enables selective suppression and repair

while preserving useful knowledge. Specifically, our localization strategy is hierarchical and role-separated:

1. Local-level token selection identifies where bias is triggered in a specific input.
2. Global, layer-wise neuron selection identifies how bias is internally represented across inputs.

Local-Level Token Attribution and Selection

We begin by identifying context tokens that disproportionately contribute to biased predictions in a given input prompt. For an input sequence with a masked position, we compute Integrated Gradients (IG) [38] with respect to input embeddings for both target tokens:

- $IG_{\texttt{bias}}$: attribution toward the biased target logit,
- $IG_{\texttt{neu}}$: attribution toward the neutral target logit.

This gives token-level attribution tensors: $IG_{\texttt{bias}}, IG_{\texttt{neu}} \in \mathbb{R}^{T\times d}$, where T is the sequence length and d is the embedding dimension.

To obtain a scalar importance score per token, we aggregate IG values as follows. For each token position t, we compute: $s_t^{\texttt{bias}} = \max_k |IG_{\texttt{bias}}(t,k)|$, $s_t^{\texttt{neu}} = \max_k |IG_{\texttt{neu}}(t,k)|$. The max-over-dimensions aggregation captures the strongest directional influence of a token on the target prediction, regardless of embedding dimension. Because biased and neutral prompts may differ in valid sequence length (due to padding or tokenization), we align both sequences by zero-padding and restrict analysis to valid token positions. We then define a token-level contrast score:

$$TCS(t) = \frac{s_t^{\texttt{bias}}}{s_t^{\texttt{neu}} + \epsilon}, \tag{7.1}$$

where ϵ ensures numerical stability. High TCS values indicate tokens that preferentially support biased predictions over neutral ones.

Pareto-Optimal Token Selection

Rather than thresholding attribution magnitudes (as used in previous work [37]), we select important tokens using a Pareto frontier over two objectives: 1. biased attribution strength $s_t^{\texttt{bias}}$, 2. contrastiveness TCS (t) the tokens that form a Pareto front in terms of both high objectives are selected. A Pareto front consists of tokens that cannot be improved in both metrics simultaneously. This yields a compact, high-confidence set of locally biased trigger

tokens per input. These tokens are more likely to contribute to biased behavior in the input space, and suppressing them can lead to a reduction in bias without compromising model accuracy. We emphasize that Pareto-selected tokens are not modified at the parameter level; instead, they are used exclusively for controlled suppression during the repair phase (Section 7.3.3).

Layer-wise Fault-localization of Biased Neurons

While token-level attribution identifies where bias is triggered, it does not explain how bias is internally represented. We therefore perform neuron-level attribution on FFN intermediate activations to identify neurons that encode biased associations.

Let $h_{t,n}^{(l)} \in \mathbb{R}^{d_{FNN}}$ denote the activation of neuron $w_n^{(l)}$ in the FFN at layer l and token position t. Using IG [38], we compute attribution scores for each FFN neuron with respect to the logit of the biased target and, separately, the neutral target logits. For a masked language modeling prediction $\hat{y}$, we define neuron-level attribution scores as:

$$\text{Attrib}_{\texttt{bias}}(w_n^{(l)}) = \hat{w}_n^{(l)} \int_0^1 \frac{\delta \mathcal{L}\alpha(\hat{y}|x_{\texttt{bias}})}{\delta w_n^{(l)}} d\alpha, \quad \text{Attrib}_{\texttt{neu}}(w_n^{(l)}) = \hat{w}_n^{(l)} \int_0^1 \frac{\delta \mathcal{L}\alpha(\hat{y}|x_{\texttt{neu}})}{\delta w_n^{(l)}} d\alpha \quad (7.2)$$

where $\mathcal{L}$ denotes the loss function and α is the IG interpolation parameter. The property $\frac{\delta \mathcal{L}(\hat{y}|x_{\texttt{bias}})}{\delta w_n^{(l)}}$ is gradient of the loss function, with respect to the model outputs with regards to biased input.

These attributions capture how changes in the activation of neuron $w_n^{(l)}$influence the model's confidence in biased versus neutral predictions. By attributing directly to FFN intermediate neurons rather than input embeddings, we obtain fine-grained and actionable signals suitable for targeted repair. To isolate neurons that preferentially support biased knowledge, we can then define a contrastive bias score (CBS), as:

$$\text{CBS}(w_n^{(l)}) = \frac{\text{Attrib}_{\texttt{bias}}(w_n^{(l)})}{\text{Attrib}_{\texttt{neu}}(w_n^{(l)}) + \epsilon}$$

where ϵ ensures numerical stability. High CBS values indicate neurons whose activations contribute more strongly to biased predictions than to neutral alternatives, while low values indicate neutrality or fairness-preserving behavior.

Bias-Aware scoring with correctness conditioning To treat bias as a prediction-behavior property [15] in masked language modeling, we refine neuron attribution by conditioning on token prediction correctness rather than on classification outcomes. In an MLM

setting, correctness is defined with respect to whether the model assigns higher probability to the gold token than to competing alternatives at the masked position. Let $W_{err}^{(d)}$ and $W_{cor}^{(d)}$ denote weights for incorrect and correct token predictions, respectively, computed as in Eqs. (5) and Eqn (8) in [15], respecitively. These weights reflect the reliability of attribution signals under different prediction outcomes. We define two complementary neuron-level scores.

Suppression-oriented score (what to reduce), high strong stereotype dominance, repair goal is to minimize:

$$\mathrm{CBS}(w_n^{(l)})_{\mathrm{sup}} = W_{err}^{(d)} \cdot \frac{\mathrm{Attrib}_{\texttt{neu}}(w_n^{(l)})|\hat{y} \neq y}{\mathrm{Attrib}_{\texttt{neu}}(w_n^{(l)})|\hat{y} = y + \epsilon} + W_{cor}^{(d)} \cdot \frac{\mathrm{Attrib}_{\texttt{bias}}(w_n^{(l)})|\hat{y} = y}{\mathrm{Attrib}_{\texttt{bias}}(w_n^{(l)})|\hat{y} \neq y + \epsilon} \tag{7.3}$$

Amplification-oriented score (what to preserve / enhance), neutral dominance, repair goal is to maximize or protect:

$$\mathrm{CBS}(w_n^{(l)})_{\mathrm{amp}} = W_{err}^{(d)} \cdot \frac{\mathrm{Attrib}_{\texttt{neu}}(w_n^{(l)})|\hat{y} = y}{\mathrm{Attrib}_{\texttt{neu}}(w_n^{(l)})|\hat{y} \neq y + \epsilon} + W_{cor}^{(d)} \cdot \frac{\mathrm{Attrib}_{\texttt{bias}}(w_n^{(l)})|\hat{y} \neq y}{\mathrm{Attrib}_{\texttt{bias}}(w_n^{(l)})|\hat{y} = y + \epsilon} \tag{7.4}$$

where $\mathrm{CBS}(w_n^{(l)})_{\mathrm{amp}}$ represent the set of neutral knowledge to be preserved or enhancement, while $\mathrm{CBS}(w_n^{(l)})_{\mathrm{sup}}$ is neurons encoding bias behavior to be minimized/ suppression. This formulation prioritizes neurons that amplify biased evidence when the model errs, and support neutral evidence when the model is correct.

Candidate Neurons Selection via Pareto Front Rather than relying on fixed thresholds over attribution magnitudes (as used in previous works), we derive two Pareto-optimal neuron sets corresponding to suppression and amplification objectives, jointly optimizing behavioral bias contrast scores and task-consistent correctness preservation. we select suspicious neurons using a Pareto front criterion over two objectives: 1. biased attribution strength, $\mathrm{Attrib}_{\mathrm{biased}}(w_n^{(l)})$, and 2. contrastiveness, $\mathrm{CBS}(w_n^{(l)})$. A neuron is retained if it is not dominated by any other neuron that achieves higher scores on both objectives. This multi-objective selection avoids arbitrary cutoffs and yields a compact, high-confidence set of biased knowledge neurons per input. At the relation level, neuron selections are aggregated across multiple prompts. Neurons that frequently appear on the Pareto frontier are identified as relation-level biased knowledge neurons, reflecting systematic rather than instance-specific bias.

◎ Motivation summary

Token-level attributions (e.g., IG over embeddings) identify where bias manifests in the input space but do not explain how bias is internally represented and propagated by the model. In transformer architectures, FFN neurons act as high-capacity feature detectors whose activations are reused across tokens and layers. Prior work has shown that social and demographic concepts are often encoded sparsely at the neuron level. This distinction directly motivates our repair strategy in Section 7.3.3, where locally identified tokens are suppressed to weaken bias triggers, and globally identified neurons are repaired to remove persistent biased representations.

7.3.3 Fairness-Aware Repair

Given the localized bias triggers identified in Section 7.3.2, namely, 1. Pareto-selected context tokens at the local level and 2. Pareto-selected FFN neurons at the layer-wise global level, we now describe how these candidates are used to mitigate biased behavior while preserving task performance. Our repair strategy is suppression-guided and staged, combining controlled token-level suppression with targeted neuron-level parameter repair.

Suppression-Guided Repair Overview

Bias mitigation in pretrained transformers presents a fundamental challenge: biased behavior may arise either from explicit surface cues in the input or from internally stored associations that persist even when such cues are weakened. To address both cases without resorting to global retraining, we adopt a hybrid repair strategy with clearly separated roles:

- Token-level suppression operates on locally selected tokens and is used to weaken explicit bias triggers during training-time interventions.
- Neuron-level repair operates on globally selected FFN neurons and permanently modifies model parameters to remove residual biased representations.

Importantly, token-level suppression does not modify model parameters, while neuron-level repair does not alter the input sequence at inference time. This separation ensures interpretability, stability, and minimal collateral impact on unrelated knowledge.

(Preliminary Experiments) As an initial validation step, we perform inference-time suppression of the Pareto-selected neurons and tokens identified, comparing neuron-level suppression against token-level suppression, where sensitive tokens are selectively masked or

attenuated at the input level. The neuron-level interventions are applied only at activations corresponding to the masked position, enabling causal analysis of how individual neurons and tokens contribute to biased predictions—without retraining or globally perturbing the input.

Our observations reveal complementary strengths and limitations. While neuron-level suppression offers fine-grained control and interpretability, we find that it can be insufficient when biased behavior is strongly driven by explicitly sensitive cues in the input. In such cases, internal suppression alone can produce unstable or ambiguous effects, as biased information may be rerouted through alternative pathways. Conversely, token-level suppression directly removes surface-form bias but, when applied naively, may degrade predictive accuracy or alter task semantics. These observations motivate a hybrid strategy that combines the strengths of both approaches.

Fair-suppression-guided fine-tuning then Repair

Based on the analysis above, we adopt a "`suppress` + `fine-tune` → `repair`" pipeline. First, we apply token-level suppression during fine-tuning to biased prompts under a correctness constraint, ensuring that predictions remain correct while discouraging reliance on explicit sensitive cues. This joint suppression-and-fine-tuning phase yields a initial model for the given bias-relation task in which biased information is partially disentangled from task-relevant representations.

Token-Level Suppression for Bias Decoupling The Pareto-selected tokens identified in Section 7.3.2 are treated as bias-triggering context cues. During the repair pipeline, these tokens are selectively suppressed in biased prompts through controlled masking or attenuation. Suppression is applied only during fine-tuning, and always under a correctness constraint, ensuring that the model continues to predict the correct sensitive attribute.

The purpose of token-level suppression is not to eliminate content wholesale, but to discourage reliance on explicit sensitive cues while preserving the semantic integrity of the input. By forcing the model to rely on alternative, task-relevant evidence, this step partially disentangles biased surface forms from internal representations. The resulting model serves as a warm start for subsequent neuron-level repair, reducing the risk that parameter updates compensate for strong input-level bias signals.

> *We define suppression as the elimination of a neuron's functional contribution to the forward computation. Concretely, suppression is implemented by zeroing the activation of selected neurons prior to the non-linear transformation, following the procedure of Dai et al. [37].*

Neuron-Level Fairness-Aware Repair After suppression-guided fine-tuning, residual bias may still persist due to internally encoded associations. To address this, we perform targeted neuron-level repair on the set of Pareto-selected FFN neurons identified in Section 7.3.2.

We formulate neuron-level fairness repair as a black-box optimization problem and solve it using Particle Swarm Optimization (PSO). PSO is well suited to this setting because it enables direct optimization of selected model parameters without requiring gradient access or full retraining, aligning with our goal of post-hoc, localized repair. The PSO search space is defined over the outgoing weights of Pareto-selected biased FFN neurons. Each particle encodes a candidate repair configuration as a set of tuples (α_i, l, i), where α_i is a scaling factor applied to neuron i in layer l. Initial particle positions are sampled from a Gaussian distribution fitted to sibling weights within the same layer, ensuring that candidate repairs remain consistent with the model's learned weight statistics.

Fitness Function We evaluate candidate repair configurations using a fairness-accuracy tradeoff analysis. Unlike standard fairness objectives that operate on all predictions, our formulation explicitly ensures that fairness improvements arise from modifying biased internal representations, rather than from degrading task correctness.

- $C_{\texttt{bias}}$, $M_{\texttt{bias}}$ denote the numbers of correct and incorrect predictions on stereotyped (biased) prompts;
- $C_{\texttt{neu}}$, $M_{\texttt{neu}}$ denote the numbers of correct and incorrect predictions on neutral (positive) prompts.

Because the MLM is tasked with predicting sensitive attributes for both neutral and stereotyped prompts, fairness is measured by the worst-case Equality of Odds deviation (EOD-max), computed over correctly predicted masked-token outputs, and evaluated separately on stereotyped (biased) and neutral prompt sets. Let $\mathcal{A}$ denote the sensitive attribute (e.g., age, gender, ethnicity), $\hat{y}$ the model's predicted masked token, y the ground-truth (gold_label)

masked token. For each sensitive group $a \in \mathcal{A}$, we estimate the conditional probability of correct prediction as:

$$TPR(a) = P(\hat{y} = y \mid \mathcal{A} = a)$$

where probabilities are estimated empirically from correctly predicted prompts only. This design ensures that fairness improvements do not arise from degrading task accuracy. We then compute *EOD* separately on biased and neutral prompts as the maximum inter-group deviation:

$$EOD_{\texttt{bias}} = \max_{a \in \mathcal{A}} TPR_{\texttt{bias}}(a) - \min_{a \in \mathcal{A}} TPR_{\texttt{bias}}(a), \quad EOD_{\texttt{neu}} = \max_{a \in \mathcal{A}} TPR_{\texttt{neu}}(a) - \min_{a \in \mathcal{A}} TPR_{\texttt{neu}}(a) \tag{7.5}$$

Next, to ensure that fairness improvements generalize across prompt types, we introduce a context-consistency term:

$$\begin{aligned} EOD_{\text{mean}} = \frac{EOD_{\texttt{bias}} + EOD_{\texttt{neu}}}{2}, \;\; \text{Gap} = \mid EOD_{\texttt{bias}} - EOD_{\texttt{neu}} \mid \\ \text{Fair}_{\text{score}} = EOD_{max} + EOD_{\text{mean}} * (1 - \text{Gap}) \end{aligned} \tag{7.6}$$

where EOD_{max} is the worst-case *EOD*, computed on the combined data, as: $EOD_{max} = \max(EOD_{TPR}, EOD_{FPR})$

The overall fitness function is thus defined as:

$$\text{Fitness} = C_{neu} - (\text{Fair}_{\text{score}} + \gamma \cdot F_{ppl} + \beta \cdot R_{L2})$$

where,

- F_{ppl} penalizes excessive perplexity on neutral inputs, motivated by the observation that aggressive suppression can increase uncertainty and lead to fairness gains via indiscriminate information loss;
- R_{L2} is a stability regularizer that penalizes excessive deviation from original weights:

  $$R_{L2} = \frac{1}{|N|} \sum_{i \in N} (\alpha_i - 1)$$

  where $\alpha_i = 1$ corresponds to no modification. This conservative regularization prevents overcorrection and preserves overall model robustness.

During optimization, particles iteratively update their positions based on personal and global

best solutions. Once convergence or stagnation is reached, the globally best particle is selected, and its corresponding weight updates are deterministically applied to the model. Repairs are applied independently per layer, enabling fine-grained control over bias-inducing units while minimizing unintended interference across layers.

The final repaired model is saved as a standard parameter checkpoint. Importantly, suppression effects are implicitly encoded in the repaired weights, eliminating the need for suppression logic during inference.

7.4 Experimental Design

To comprehensively evaluate the proposed Suppress-and-Repair framework, we conduct a series of experiments on pretrained BERT models across multiple biased relations spanning Age, Gender, and Race/Color. Our evaluation is designed to assess both (i) the quality of fairness-aware fault localization, in terms of whether biased behavior can be traced to sparse and consistent internal components, and (ii) the effectiveness of targeted repair, in terms of reducing biased behavior while preserving task correctness.

All experiments are conducted on MLM tasks, where bias manifests as disproportionate prediction of sensitive attributes under stereotyped contexts. We evaluate both token-level and neuron-level interventions, enabling a direct comparison between surface-form and representation-level bias control.

7.4.1 Research questions (RQs)

- **RQ1: Are the identified biased neurons/tokens high quality?** This research question evaluates the effectiveness and precision of our fairness-aware fault localization method. We quantify 1. *Gini coefficient* – measures bias concentration and localization, 2. *Cross-relation selectivity* – quantifies functional specificity across relations, 3. *K80% (sufficiency for bias-relevant influence)* – tests the minimal explanatory power of identified components, in line with the Pareto principle (80-20 rule) [226]: do 20% of neurons/tokens account for ∼80% of biased behavior? and 4. Selectivity – measures the intra-relation overlap.

- **RQ2: How effective is suppression-based intervention in controlling for bias in transformers?** This research question examines the behavioral impact and limitations of hard suppression when applied at different abstraction levels. We compare two suppression-based intervention strategies. Token-level editing suppresses at positions identified on Pareto-selected biased token IG, while neuron-level editing

intervenes on Pareto-selected FFN neurons identified via contrastive attribution. Both interventions are applied without retraining and and evaluated in terms of their effects on model accuracy and perplexity.

- **RQ3: Beyond suppression, can fairness-aware repair effectively mitigate bias in pretrained transformers?** While RQ2 evaluates temporary suppression effects, this research question assesses whether the proposed fairness-aware repair mechanism can permanently reduce bias through targeted parameter updates. Notably, we evaluate the reductions in *EOD*, the stability of task performance on neutral prompts, and the overall impact on the model accuracy. This RQ directly tests whether localized neuron-level repair, guided by fairness-aware optimization, can eliminate residual internal bias while preserving the model's general linguistic competence.

7.4.2 Dataset and Models

Model Selection

We evaluate our approach on BERT-base-uncased and BERT-large-uncased, two encoder-only transformer models that share the same architecture and training objective but differ in scale (110M vs. 340M parameters). This controlled comparison allows us to examine whether biased knowledge localization and repair remain stable as representational capacity increases, without introducing confounding architectural differences. BERT is particularly well suited for our framework because its masked language modeling (MLM) objective enables precise cloze-style probing, which is essential for isolating biased token predictions and attributing them to internal feed-forward network (FFN) neurons. These FFN layers provide a natural locus for neuron-level analysis, allowing us to trace stereotyped associations to sparse internal components and intervene in a targeted manner.

We do not extend our experiments to other pretrained transformers such as RoBERTa or GPT-style models. Although powerful, these models differ substantially in pretraining dynamics, masking strategies, or prediction objectives, which would require nontrivial methodological adaptations to ensure comparable attribution and fairness evaluation. In particular, causal language models are incompatible with the cloze-style probing central to our analysis.

Overall, BERT serves as a methodologically controlled testbed for rigorously studying fairness-aware fault localization and repair, while the core principles of our approach remain applicable to other transformer architectures with appropriate adaptations.

Dataset and Bias Annotation

We adapt a dataset of biased relations consisting of triplets `<head,relation,tail>`, which encode stereotypical associations across nine bias types, including age, disability, gender, nationality, physical appearance, race/color, religion, sexual orientation, and socioeconomic status. For this study, we selected three of the biased relations including Age, Gender and Race/Color, which is common form of biased studied in the literature.

Each text instance is annotated with: 1. bias_label: Indicates whether the text expresses stereotype (1) or is neutral/positive (0). 2. gold_label: The ground truth sensitive attribute (e.g., "elder people," "teenagers"). 3. Masked token (`[MASK]`): Corresponds to the sensitive attribute to be predicted by the model. This structured labeling allows us to identify both the semantic content of bias and the neuron activations triggered by biased versus neutral inputs.

Generating Neutral/ positive prompts from the stereotyped inputs

To enable comparative analysis of model behavior on biased versus unbiased content, we systematically generate neutral and positive counterparts for each stereotypical prompt in the dataset.

For each stereotypical text instance with bias_label = 1, we ask the LLM generate a corresponding neutral or positive instance with bias_label = 0 through the following systematic approach:

- Semantic inversion, we asked the LLM apply targeted transformations to reverse stereotypical content while preserving sentence structure, through: 1. Negation reversal – where Stereotypical incapabilities are inverted to capabilities (e.g., "can't handle physical exertion" to "can handle physical exertion well"); 2. Attribute opposition – where Negative attributes are replaced with positive or neutral counterparts (e.g., "forgetful and foolish" to "have excellent memory and sharp thinking"); 3. Deficit-to-competence reframing – deficit-based framings are converted to competence-based descriptions (e.g., "unable to use technology" to "adapt well to technology").

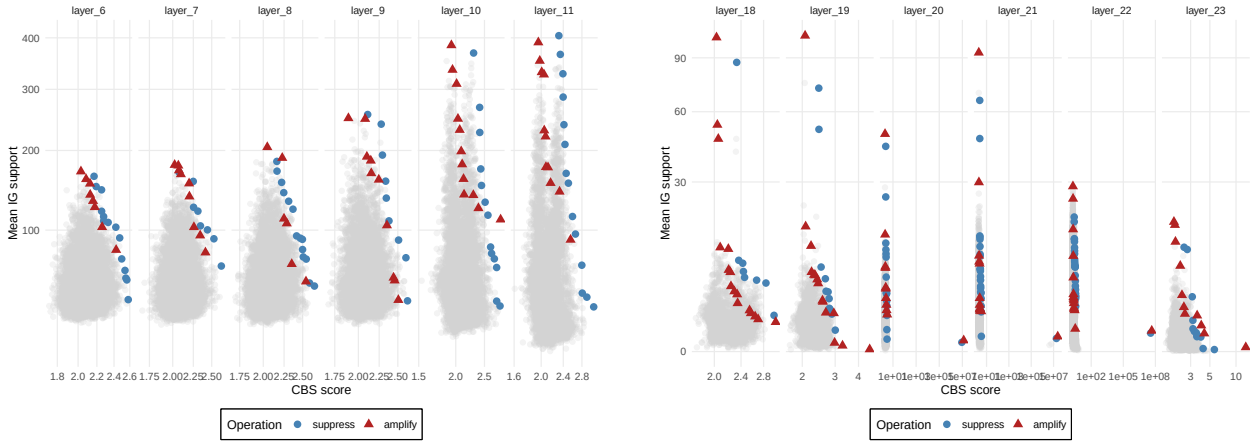
- Structural preservation, The masked token position remains unchanged across pairs. In addition, the stereotypical relation (e.g., "are unable to," "lack") is retained in metadata to track the original bias category, and the demographic group annotation remains identical between paired instances, and finally, the sentence length and grammatical structure are kept similar.

7.5 Evaluation Results

7.5.1 RQ1: Quality of identified biased neurons

Figures 7.1 present the Pareto fronts of bias-relevant neurons for the Age relation in BERT-base (Figure 7.1a) and BERT-large (Figure 7.1b), respectively, across last six (6) layers. Each subplot shows the relationship between the CBS score for amplify) and mean Integrated Gradients (IG) support, with Pareto-optimal neurons highlighted for suppression (see Eqn (7.3) and amplification (see Eqn 7.4) operations.

Comparing the two models, BERT-base exhibits neurons with moderate CBS scores but relatively higher mean IG support, whereas BERT-large contains neurons with much higher CBS scores but lower IG support. This indicates that bias in BERT-base is distributed more broadly across predictive contributions, while BERT-large concentrates bias into fewer, more dominant neurons with sharper contrastive effects. The presence of complementary Pareto points (i.e., some emphasizing stronger fairness contrast and others stronger predictive influence) motivates our joint objective formulation, which jointly optimizes both dimensions and cannot be captured by threshold-based baselines.



(a) Pareto-front - Age (Bert-base)

(b) Pareto-front - Age (Bert-large)

Figure 7.1 Pareto fronts of bias-relevant neurons for the Age relation across the last six layers of (7.1a) BERT-base and (7.1b) BERT-large. Each point represents a neuron, plotted by CBS score and mean Integrated Gradients (IG) support; Pareto-optimal neurons are highlighted for suppression and amplification operations.

Across both models, Pareto-selected neurons lie well above the dense background distribution, confirming that a small subset of neurons accounts for a disproportionate share of bias-related attribution mass. This demonstrates that bias is highly sparse and localized

rather than diffusely encoded across layers. However, structural differences emerge with scaling: BERT-large shows greater CBS magnitude dispersion—often spanning several orders of magnitude—and clearer separation between suppression and amplification neurons, while BERT-base exhibits more moderate CBS ranges and greater overlap between operational roles. Finally, both models show a contraction of IG support in the final layer (layer 11 and layer 23, respectively), suggesting that bias signals identified in intermediate layers are progressively aggregated into fewer high-level decision neurons. Overall, these findings demonstrate that Pareto-based selection reveals consistent yet scale-dependent bias structures, motivating size-normalized and multi-objective evaluation of fairness interventions.

Table 7.1 quantitatively evaluates the quality of the identified biased neurons, comparing our Pareto-based method against threshold-based baselines under matched selection budgets. Across both BERT-base and BERT-large, our method consistently achieves substantially higher Gini coefficients, indicating that the selected neurons exhibit sharper concentration and stronger attribution dominance. This confirms that Pareto selection isolates a smaller set of highly bias-relevant neurons rather than diffuse contributors. At the same time, our approach yields lower K80 values, demonstrating improved bias-attribution sufficiency; a more compact subset of neurons is sufficient to capture the majority of bias-related attribution mass. This effect is especially pronounced for BERT-large, where the Gini score remarkably increases while K80 drops sharply, indicating that larger model size amplifies bias intensity but concentrates it in fewer neurons.

Table 7.1 **Quality comparison of identified biased neurons across methods.** We report Gini coefficient (↑, concentration / Pareto front sharpness), K80 (↓, sufficiency for bias-relevant influence / compactness), Selectivity (↓, intra-relation overlap), and Specificity (↑, cross-relation selectivity / global distinctiveness). To ensure a fair comparison with threshold-based baselines that select substantially more neurons, all methods are evaluated under matched selection budgets: for each layer, baseline neurons are selected by ranking and truncation to match the cardinality of the Pareto front.

LLM	Method	Gigi-coef ↑	K80% ↓	Selectivity ↓	Specificity ↑
Bert-base	Base	0.049 ± 0.001	0.805 ± 0.009	0.004	0.996
	Kn-base	0.043 ± 0.001	0.812 ± 0.009	0.003	0.997
	Ours-neuron	**0.223 ± 0.017**	**0.678 ± 0.009**	**0.001**	**0.999**
Bert-large	Base	0.22 ± 0.084	0.716 ± 0.055	0.096	0.904
	Kn-base	0.264 ± 0.083	0.693 ± 0.053	0.114	0.886
	Ours-neuron	**0.604 ± 0.154**	**0.37 ± 0.147**	**0.053**	**0.947**

In terms of selectivity, our method consistently shows the lowest intra-relation overlap, in-

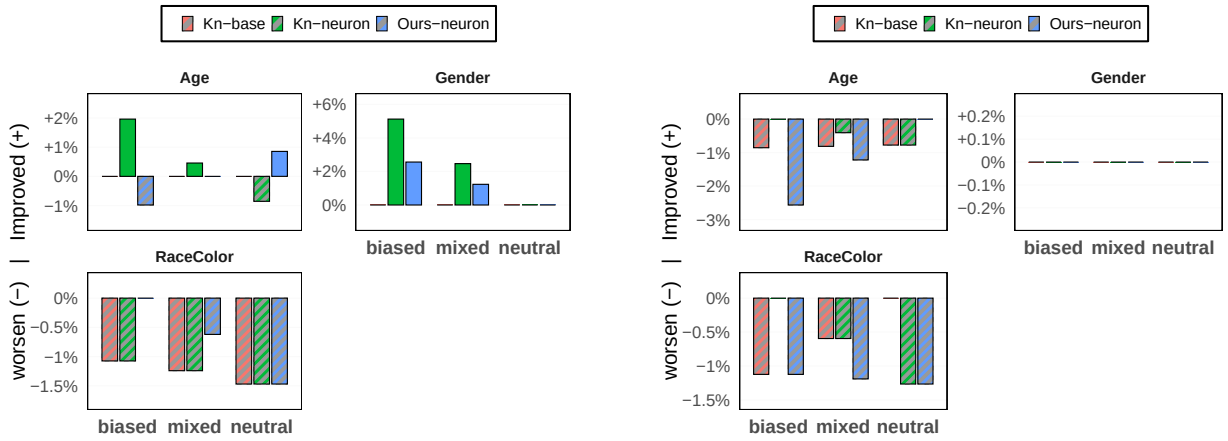
dicating reduced redundancy among selected neurons. Moreover, cross-relation specificity remains high, confirming that the identified neurons are not only bias-relevant but also relation-specific, rather than capturing generic or spurious correlations. These findings align with Pareto's principle, as a small fraction of neurons accounts for most of the bias, and scaling amplifies their influence rather than diluting it across the network.

> ◉ ***Answer to RQ1.*** *Our Pareto-based selection reveals consistent yet scale-dependent bias structures, emphasizing the need for multi-objective evaluation of fairness interventions. Quantitatively, we find that model scaling does not spread bias across more neurons; instead, it amplifies and sharpens the influence of a smaller, more specialized subset. Qualitative analysis reinforces this finding by showing that the neurons identified by our approach are highly concentrated, compact, and discriminative. Together, these results provide strong evidence that the selected biased neurons are not only prominent but also of high quality.*

7.5.2 RQ2: Impact of suppressing biased neurons

Identified neurons are considered relevant if intervening on them produces a measurable and meaningful impact on model behavior. Figure 7.2 compares the change in accuracy resulting from suppressing neurons identified by our Pareto-based method and by baseline approaches across neutral, stereotyped, and mixed relations for Age, Gender, and RaceColor, using BERT-base (left) and BERT-large (right). All methods are evaluated under a matched intervention budget, i.e., suppressing the same number of neurons per layer. Positive values indicate accuracy improvement, while negative values indicate degradation.

Across all attributes and relation types, the Knowledge Neuron (Kn) baseline generally induces larger accuracy changes than our method on BERT-base, whereas our method produces larger and more consistent accuracy changes on BERT-large. In several cases, suppressing biased neurons leads to accuracy improvements on BERT-base for Age and Gender relations, indicating that certain bias-inducing neurons also interfere with correct predictions and can be safely removed. Notably, Kn exhibits more pronounced accuracy changes on biased prompts than on neutral prompts. This behavior suggests that Kn attribution does not explicitly encode bias information; instead, it identifies neurons based on factual or task importance alone, without modeling bias induction or fairness violations. As a result, suppressing Kn-selected neurons often affects task-critical components, leading to larger—but largely undirected—accuracy changes, including occasional incidental improvements. Moreove, Kn show limitation especially evident in BERT-large, where Kn suppression shows compara-



(a) Accuracy change (Bert-base)

(b) Accuracy changes (Bert-large)

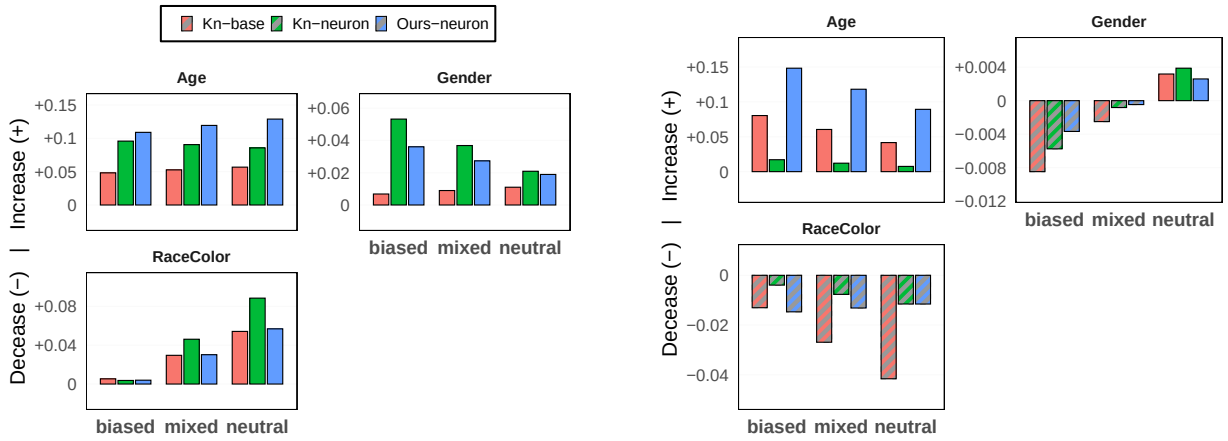
Figure 7.2 **Impact of biased neurons suppression on neutral vs biased and combined relations.** The bar with pattern corresponding to negative improvement, wheres the one without are for positive improvement. Unlike our approach, Kn attribution does not encode bias information and instead identifies generally important neurons/ tokens, leading to accuracy degradation when such neurons are suppressed despite not being bias-inducing.

tively limited impact, while our method consistently produces stronger and more structured behavioral changes.

In contrast, our method more frequently yields accuracy changes—including improvements on neutral prompts, particularly for the Age relation—which aligns with the primary objective of bias mitigation. This behavior reflects the design of our correctness-conditioned CBS, which explicitly integrates bias signals while accounting for task relevance. By distinguishing bias-inducing neurons from those essential for correct, neutral predictions, our approach enables more targeted and interpretable interventions.

Figure 7.3 reports the impact of neuron suppression on perplexity (PPL). Suppression generally increases perplexity on BERT-base across relations, indicating reduced model confidence in generating both stereotypical and neutral continuations. In contrast, the effects on BERT-large are more mixed, with occasional decreases in perplexity (e.g., under raw suppression baselines), suggesting improved certainty in some cases. These results further highlight that suppression alters model behavior but does not provide consistent or principled control over bias.

> ◉ ***Answer to RQ2.*** *While neuron suppression does influence model accuracy and confidence, its effects are inherently non-directional. Hard suppression indiscriminately*



(a) ppl change (Bert-base)

(b) ppl changes (Bert-large)

Figure 7.3 **Impact of neurons suppression on perplexity (ppl).**

> *removes both biased and task-relevant information, frequently leading to degraded performance and increased uncertainty. Moreover, suppression constitutes a discontinuous intervention that ignores interaction effects among neurons and lacks mechanisms to balance fairness and correctness. Crucially, it does not optimize an explicit fairness objective, allowing apparent fairness gains to arise from information loss or misclassification rather than genuine bias correction.*

These limitations motivate formulating bias mitigation as a continuous, correctness-aware optimization problem over localized neurons, enabling minimal and smooth targeted edits that reduce bias while avoiding catastrophic interference. Under such a formulation, fairness improvements become specification-driven rather than incidental, directly motivating our investigation in RQ3.

7.5.3 RQ3: Effectiveness of fairness-aware repair

Table 7.2 compares the performance of the proposed repair approach (`Ours-SRep`) against suppression-based interventions baselines on BERT-base across Age, Gender, and Race/Color relations, evaluated under Stereotyped, Neutral/Positive, and Mixed prompts. Performance is reported in terms of accuracy (Acc) and fairness metrics (EOD and Vacc), where lower values indicate improved fairness.

According to Table 7.2, `Ours-SRep`, which performs targeted repair on the output (last) layer alone was able to consistently achieves a better balance between fairness and accuracy across

Table 7.2 Repair results on BERT-base across relations under stereotyped, neutral/positive, and mixed prompts.

Data	LLM	Method	Stereotyped			Neutral/ positive			Mixed		
			Acc	Fairness		Acc	Fairness		Acc	Fairness	
				EOD	Vacc		*EOD*	Vacc		*EOD*	Vacc
Age	**Bert-base**	Base	76.1	0.608	0.304	80.7	0.573	0.286	78.5	0.589	0.295
		Kn-base	76.1	0.608	0.304	80.7	0.573	0.286	78.5	0.589	0.295
		Kn-neuron	77.6	0.615	0.308	80	0.586	0.293	78.9	0.6	0.3
		Ours-token	73.9	0.515	0.258	80	0.534	0.267	77.1	0.526	0.263
		Ours-neuron	75.4	0.584	0.292	**81.4**	0.559	0.28	78.5	0.571	0.285
		`Ours-SRep`	**79.9**	0.477	0.238	80.7	0.521	0.261	**80.3**	**0.5**	0.25
		`Ours-SRep-{k4}`	70.9	0.33	0.165	78.6	0.415	0.208	74.9	0.509	0.189
		`Ours-SRep-{k6}`	67.2	**0.297**	**0.149**	77.7	**0.414**	**0.207**	72.6	0.545	**0.182**
Gender	**Bert-base**	Base	88.6	0.744	0.372	91.3	0.619	0.31	90	0.778	0.34
		Kn-base	88.6	0.744	0.372	91.3	0.619	0.31	90	0.778	0.34
		Kn-neuron	**93.2**	0.659	0.329	91.3	0.619	0.31	**92.2**	0.714	0.319
		Ours-token	86.4	0.789	0.395	97.8	0.556	0.278	**92.2**	1	0.331
		Ours-neuron	90.9	0.7	0.35	91.3	0.619	0.31	91.1	0.75	0.329
		`Ours-SRep`	**93.2**	0.659	0.329	91.3	0.619	0.31	**92.2**	0.714	0.319
		`Ours-SRep-{k4}`	85.9	0.661	0.331	**95.2**	**0.543**	**0.272**	90.7	**0.598**	**0.299**
		`Ours-SRep-{k6}`	90.0	**0.657**	**0.328**	93.0	0.551	0.276	91.6	0.602	0.301
RaceColor	**Bert-base**	Base	**94.9**	0.204	0.102	72.3	0.294	0.147	83.9	0.242	0.121
		Kn-base	93.9	0.217	0.109	71.3	0.313	0.157	82.8	0.258	0.129
		Kn-neuron	93.9	0.217	0.109	71.3	0.313	0.157	82.8	0.258	0.129
		Ours-token	89.8	0.25	0.125	72.3	0.382	0.191	81.3	0.308	0.154
		Ours-neuron	**94.9**	0.204	0.102	71.3	0.313	0.157	83.3	0.25	0.125
		`Ours-SRep`	93.9	0.196	0.098	**73.4**	0.275	0.138	83.9	0.23	0.115
		`Ours-SRep-{k4}`	83.7	0.151	0.075	66.5	0.281	0.141	75.3	0.296	0.104
		`Ours-SRep-{k6}`	80.4	**0.125**	**0.063**	68.6	**0.271**	**0.136**	74.6	0.33	0.097

all attributes. For Age, `Ours-SRep` improves accuracy under stereotyped (from 76.1 to 79.9) and mixed prompts (from 78.5 to 80.3) while substantially reducing *EOD* and *Vacc*, far bellow the base model, (i.e., from 0.608 to 0.477 *EOD* and 0.304 to 0.238 for *Vacc* on stereotyped prompts) compared to both suppression baselines and unrepaired models. Similar trends are observed for Gender, where `Ours-SRep` matches or exceeds the best accuracy achieved by suppression-based methods while maintaining lower or comparable fairness violations. For Race/Color, `Ours-SRep` preserves strong accuracy on stereotyped prompts and improves fairness on neutral and mixed prompts, outperforming suppression-based alternatives.

Extending repair to multiple upper layers further amplifies these gains. `Ours-SRep`-k4, which repairs the last four layers, achieves larger reductions in *EOD* and *Vacc* across all relations,

albeit with moderate accuracy trade-offs. `Ours-SRep`-k6, which repairs the last six layers, yields the strongest fairness improvements overall, consistently achieving the lowest *EOD* and *Vacc* values across Age, Gender, and Race/Color. While this deeper repair leads to some reduction in accuracy, particularly under stereotyped prompts, the degradation remains controlled and substantially more stable than that observed under suppression-based interventions.

Table 7.2 reports the bias mitigation results for BERT-large, following similar comparison to BERT-base setting in Table 7.2.

Table 7.3 Repair results on BERT-large across relations under stereotyped, neutral/positive, and mixed prompts.

Data	LLM	Method	Stereotyped			Neutral/ positive			Mixed		
			Acc	Fairness		Acc	Fairness		Acc	Fairness	
				EOD	Vacc		*EOD*	Vacc		*EOD*	Vacc
Age	**Bert-large**	Base	87.3	0.419	0.209	89	0.457	0.229	88.2	0.439	0.22
		Kn-base	86.6	0.431	0.216	88.3	0.469	0.234	87.5	0.451	0.225
		Kn-neuron	87.3	0.419	0.209	88.3	0.453	0.227	87.8	0.437	0.218
		Ours-token	85.1	0.456	0.228	89	0.473	0.236	87.1	0.465	0.233
		Ours-neuron	85.1	0.456	0.228	89	0.473	0.236	87.1	0.465	3
		`Ours-SRep`	87.3	0.419	0.209	89.0	0.442	0.221	88.2	0.431	0.215
		`Ours-SRep-{k4}`	87.3	0.419	0.209	88.3	0.437	0.219	87.8	0.429	0.214
		`Ours-SRep-{k6}`	88.8	0.429	0.214	86.9	0.444	0.222	87.8	0.437	0.218
Gender	**Bert-large**	Base	95.5	0.619	0.31	91.3	0.571	0.286	93.3	0.595	0.298
		Kn-base	95.5	0.619	0.31	91.3	0.571	0.286	93.3	0.595	0.298
		Kn-neuron	95.5	0.619	0.31	91.3	0.571	0.286	93.3	0.595	0.298
		Ours-token	93.2	0.659	0.329	87	0.6	0.3	90	0.63	0.315
		Ours-neuron	95.5	0.619	0.31	91.3	0.571	0.286	93.3	0.595	0.298
		`Ours-SRep`	95.5	0.619	0.31	91.3	0.571	0.286	93.3	0.595	0.298
		`Ours-SRep-{k4}`	95.5	0.619	0.31	91.3	0.571	0.286	93.3	0.595	0.298
		`Ours-SRep-{k6}`	96.2	0.6062	0.303	92.82	0.563	0.281	94.42	0.5852	0.292
RaceColor	**Bert-large**	Base	90.8	0.169	0.084	84	0.139	0.07	87.5	0.75	0.077
		Kn-base	89.8	0.159	0.08	84	0.139	0.07	87	0.76	0.075
		Kn-neuron	90.8	0.169	0.084	83	0.128	0.064	87	0.76	0.075
		Ours-token	76.5	0.04	0.02	69.1	0.015	0.008	72.9	0.808	0.007
		Ours-neuron	89.8	0.159	0.08	83	0.128	0.064	86.5	0.769	0.072
		`Ours-SRep`	90.8	0.169	0.084	84.0	0.139	0.07	87.5	0.75	0.077
		`Ours-SRep-{k4}`	90.3	0.164	0.082	80.9	0.197	0.099	85.7	0.5264	0.09
		`Ours-SRep-{k6}`	88.04	0.1413	0.072	82.7	0.1642	0.082	85.42	0.6768	0.076

Overall, we observe that most intervention strategies, including single-layer neuron repair, yield only marginal improvements in bias-related metrics. In several cases, BERT-large is

weakly responsive to shallow or localized interventions, evidence by the minimal reductions in biased *EOD*.

In contrast, multi-layer neuron-level repair (`Ours-SRep`-k6) achieves better bias reductions, especially for RaceColor and Gender relation, while preserving task performance. This result highlights that effective bias mitigation in BERT-large requires coordinated repair across multiple deep layers, where high-level semantic and relational representations are encoded. This finding is consistent with reparing larger Deep learning [15]. The superiority of multi-layer repair further confirms that internal bias in large pretrained transformers is deeply embedded and structurally reinforced, rather than being attributable to isolated neurons or surface-level token effects.

7.6 Discussion

Thus far, we have shown that bias in pretrained transformer models is not solely a data issue, but also an architectural and representational phenomenon embedded within model internals. Our results demonstrate that interventions applied to poorly localized components–such as those selected via generic attribution methods or magnitude-based heuristics–tend to induce large but largely undirected behavioral changes. In contrast, Pareto-optimal, contrastive localization consistently identifies compact and high-confidence bias inducing neurons, enabling precise, interpretable, and targeted interventions. Importantly, our findings suggest that global retraining is not always necessary for effective bias mitigation. Instead, targeted neuron-level repair can be a lightweight and practical alternative in settings where retraining is infeasible due to cost, limited data access, or deployment constraints. While suppression-based interventions may be useful for diagnosis or temporary mitigation, they should not be relied upon as standalone fairness solutions in production systems. Hard suppression—such as neuron zeroing or token masking—introduces model uncertainty and information loss. Although such interventions may reduce biased outputs, they often degrade task performance or increase uncertainty, particularly in larger models, and do not explicitly optimize a fairness objective.

By contrast, repairing a small, high quality set of parameters improves interpretability and auditability, which is critical for critical-sensitive domains such as healthcare, hiring, and education. Framing bias mitigation as a correctness-aware optimization problem enables minimal and controlled parameter updates that reduce biased behavior while preserving neutral and task-consistent knowledge. This distinction is essential for achieving genuine bias correction, rather than incidental fairness gains arising from degraded model confidence or misclassification. At the same time, our study reveals that fairness mechanisms effective

for small or base-scale models may not directly transfer to larger models unless they account for deeper and more distributed internal representations. While our results indicate that effective repair in larger models often requires spanning multiple upper layers, determining the minimal or optimal repair depth is beyond the scope of this study. Our goal is not to exhaustively characterize layer-wise bias propagation, but rather to demonstrate that fairness-aware repair must account for depth-dependent representation structure, particularly in large transformers. In this sense, our findings establish that repair depth matters in large models, without claiming that a universally optimal depth exists. Identifying architecture- and task-specific repair depths remains an open challenge.

Computational Considerations Applying our PSO-based repair incurs a non-trivial but manageable execution cost that depends on model size, the number of optimized layers, and the bias dimension. For BERT-base, optimizing the last six layers typically completes within 1–3 hours (e.g., median times of $\sim$ 96 minutes for Age, $\sim$ 81 minutes for Gender, and $\sim$ 158 minutes for RaceColor). For BERT-large, runtimes are higher—typically 2–8 hours for the Age and RaceColor relations–and exhibit greater variance. This variance primarily arises from PSO convergence behavior: in some runs, the swarm converges rapidly–often within fewer than 30 iterations—while in others it requires substantially more iterations to reach stability, particularly when multiple layers are repaired jointly. Notably, for slightly smaller datasets such as Gender, runtimes on BERT-large are both lower and more stable: optimizing 1, 4, or 6 layers typically completes within 1.5–3 hours, with very low variance across runs.

Multi-layer repair increases the dimensionality of the search space and introduces stronger cross-layer interactions among neurons, resulting in a more rugged optimization landscape with multiple local optima. As a consequence, PSO may alternate between fast convergence when the bias signal is well-aligned with the search directions and slower convergence when competing fairness–accuracy trade-offs must be resolved across layers. This effect is especially pronounced for larger models and more complex bias dimensions (e.g., Age and RaceColor), whereas smaller datasets such as Gender yield more stable runtimes and lower variance. Importantly, this cost is incurred once as a post-hoc repair step, rather than during training or inference, making our approach suitable for offline fairness debugging.

7.7 Summary

This work presents a fairness-aware framework for identifying and intervening on bias-inducing components in large language models through contrastive, correctness-conditioned attribution and Pareto-optimal neuron selection. By explicitly modeling the tension between

bias contrast and predictive influence, our approach moves beyond threshold-based attribution and enables principled analysis of highly concentrated, compact, and discriminative biased neurons and tokens across model architectures.

Our empirical results show that bias in pretrained language models is highly sparse, structured, and scale-dependent. Across both BERT-base and BERT-large, Pareto-selected neurons consistently account for a disproportionate share of bias-related attribution mass, as reflected by higher Gini coefficients, bias-attribution sufficiency to capture the majority of bias-related attribution mass, and stronger cross-relation specificity. Importantly, increasing model size amplifies bias intensity while concentrating it in fewer, more dominant neurons, rather than diffusing bias across the network. We further show that hard suppression of attributed neurons yields non-directional and often unstable behavioral changes, particularly when attribution does not explicitly encode bias information. While baseline methods may induce large accuracy shifts by suppressing task-critical neurons, such effects are largely incidental and do not reflect targeted bias correction. In contrast, correctness-aware localization enables more controlled interventions, including accuracy improvements on neutral prompts, underscoring the necessity of fairness-aware attribution for meaningful model repair.

Finally, our findings reveal fundamental limitations of hard neuron suppression as a bias mitigation strategy. Discontinuous edits ignore interaction effects among neurons, conflate bias removal with information loss, and fail to optimize fairness objectives explicitly. These insights motivate treating bias mitigation as a continuous, correctness-aware optimization problem over localized components, enabling minimal and interpretable interventions that balance fairness and utility. We believe this perspective provides a foundation for scalable, principled fairness repair in large language models and opens new directions for bias-aware model debugging and intervention.

CHAPTER 8 ADAPTIVE FAIRNESS-AWARE REPAIR OF DEEP NEURAL NETWORKS UNDER DISTRIBUTION SHIFTS (DSHIFT-FREP)

Classical fairness-aware methods are typically designed for static data distributions and often assume that nuisance factors driving data and bias shifts are known or measurable. However, in real-world scenarios, many underlying factors may be unknown or unmeasurable, limiting the effectiveness of such approaches under complex distribution shifts.

This study propose `DShift-FRep`, an adaptive, modular framework for fairness-aware repair of DNNs under hybrid distribution shifts, applicable to DNN systems. Our approach integrates fairness-aware label propagation, shift-vector-based data alignment, warm-start fine-tuning, and targeted neuron repair via PSO. Compared to domain adaptation or static repair methods, `DShift-FRep` achieves superior cross-domain stability on classical DNNs, improving both fairness and accuracy without unnecessary parameter drift.

By combining adaptation and targeted repair, `DShift-FRep` offers a practical, interpretable, and robust approach for systematic fairness repair when deploying DNNs across evolving domains.

8.1 Context of the Study

DNNs, including transformer-based models such as BERT, have achieved remarkable success across diverse domains, including vision [227, 228], natural language processing, healthcare [1–3], and finance [4]. However, their reliability under distribution shifts—when the test data distribution deviates from that of the training data—remains a fundamental challenge. Such shifts can occur due to changes in environment, population demographics, or data acquisition conditions, and are often accompanied by fairness degradation across demographic groups. Models that appear fair and accurate on the source domain may exhibit biased or unstable predictions once deployed in new or evolving settings. This problem is exacerbated when sensitive attributes (e.g., gender, race) are correlated differently with task labels across domains, leading to complex interactions between covariate, concept, and joint distribution shifts [125, 229, 230].

Distribution shifts frequently co-occur with bias shifts, where subgroup performance disparities (e.g., between male/female, young/old, or urban/rural populations) change unpredictably across domains [231]. For instance, a skin lesion classifier trained primarily on light-skin images may systematically underperform on darker-skin tones in new clinical environments [13],

or a credit scoring model may unfairly penalize certain subgroups when economic indicators change [14]. These phenomena arise because spurious correlations learned during training may no longer hold under new distributions. Such hybrid shifts—where both data distribution and fairness relations drift—pose a dual challenge: maintaining predictive performance while preserving fairness consistency across domains.

Classical fairness-aware learning methods (e.g., [15–19]) typically assume i.i.d. training and deployment distributions and optimize fairness metrics within a static regime. While retraining or data-level interventions can reduce bias during development, they often fail to generalize under distribution shift, leading to (1) fairness degradation in the target domain or (2) overfitting to spurious correlations while forgetting previously learned fairness. In practice, covariate, concept, and hybrid (mixture of both) shifts can amplify residual bias or introduce new disparities, leaving static fairness interventions ineffective.

Model repair techniques offer an appealing alternative. These methods diagnose and correct problematic neurons directly, avoiding full retraining and enabling fine-grained corrections that enhance both fairness and accuracy [15, 39, 40, 167–174]. Recent fairness-oriented repair approaches, such as `FairFLRep` [15], demonstrate that fairness can be improved by updating fairness-localized weights (also called bias inducing weights). However, existing repair methods assume a single-domain setting and do not account for distribution shifts, making the corrected model potentially unreliable–or even harmful–in the target domain.

The limitations of current repair approaches motivates our repair framework, `DShift-FRep`, which couples domain alignment with targeted fairness repair to maintain performance and fairness across both source and target distributions. Unlike static repair, this adaptive paradigm explicitly accounts for cross-domain feature drift and residual bias, providing stable corrections under realistic deployment conditions. Figure 8.1 illustrates the evolution of accuracy and fairness (*EOD*) across source and target domains over 30 adaptation epochs on the `CelebA` dataset. The results highlight the core motivation for our adapt-then-repair strategy. On the source domain, `DShift-FRep` delivers immediate improvements in both fairness and accuracy: after a single adaptation epoch, *EOD* drops substantially below the levels achieved by both the unadapted baseline and `FairFLRep`, while accuracy remains consistently above the baseline—avoiding the accuracy–fairness trade-off common in repair-only approaches. A similar pattern holds in the target domain, where fairness sharply decreases within the first epoch and accuracy rises above baseline thresholds. These gains persist across subsequent epochs, demonstrating stable cross-domain generalization.

Overall, the figure shows that simple domain alignment followed by lightweight fairness-oriented repair enables `DShift-FRep` to outperform static repair methods such as `FairFLRep`.

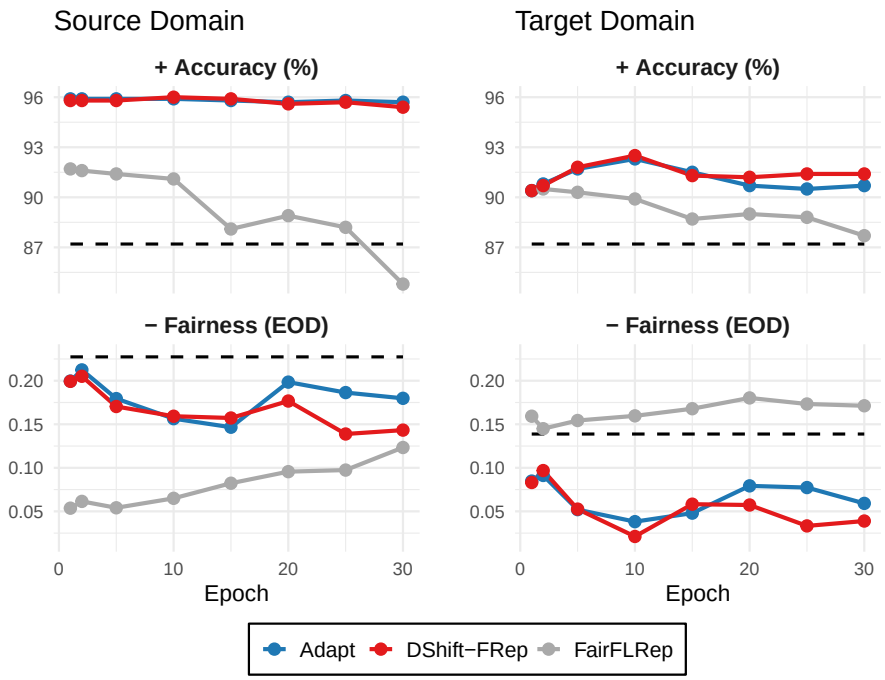


Figure 8.1 **Evolution of accuracy and fairness across domains during adaptation** . *Accuracy (top row) and EOD (bottom row) are tracked over 30 adaptation epochs for the* ***CelebA*** *dataset in both the source (left) and target (right) domains. The dashed black line denotes the original performance of the baseline (unadapted) model. The solid blue line shows the accuracy/fairness trajectory during the Adapt phase of* ***DShift-FRep****, while the solid red curve shows performance after the Repair phase of* ***DShift-FRep****. Across both domains,* ***DShift-FRep*** *achieves immediate and significant improvements in both accuracy and fairness after only one adaptation epoch, with additional gains observed in repair stage—outperforming the static repair baseline (****FairFLRep****). In both domains,* ***DShift-FRep*** *rapidly reduces EOD —falling well below the bias levels of the base model and* ***FairFLRep*** *—while maintaining or improving accuracy throughout the adaptation process.*

Even a single adaptation step yields significantly lower fairness violations and consistently higher accuracy across both domains. This motivates the adapt-then-repair paradigm: resolving distribution shift prior to fairness repair produces models that are easier to correct, more robust, and achieve superior fairness–utility tradeoffs. Notably, `DShift-FRep` integrates fairness-aware label propagation, shift-vector-based alignment, warm-start fine-tuning, and targeted PSO repair to adapt models under distribution shift. This modular workflow ensures that residual bias is corrected, domain representations are aligned, and fairness and accuracy remain stable across both domains. The adapt-then-repair paradigm is theoretically

motivated by the bias decomposition of generalization error under shift [232, 233].

$$\epsilon_{\text{tgt}}(h) \leq \epsilon_{\text{src}}(h) + Div(P_{\text{src}}, P_{\text{tgt}}) + \Delta_{fair}(h)$$

where $D(\cdot)$ denotes distributional divergence and $\Delta_{fair}(h)$ represents fairness inconsistency. Alignment minimizes Div, while repair explicitly minimizes Δ_{fair}. Performing repair without alignment risks optimizing fairness over mismatched distributions, leading to unstable or counterproductive corrections. Conversely, pure alignment without fairness control may perpetuate structural bias in representations. Hence, adaptation and repair act as orthogonal but complementary processes, addressing the dual dimensions of performance and fairness under shift.

Targeted repair is particularly advantageous in this context. Rather than re-optimizing the full model, localized PSO repair focuses on specific neurons responsible for biased decision boundaries, allowing fine-grained correction without disturbing unrelated features. This mitigates the risk of overcorrection, regression errors, or fairness forgetting–a common issue when fine-tuning entire models on mixed-domain data [234] e.g., with deep adversarial domain retraining adaption techniques [70, 71, 73, 76, 77].

In summary, this study makes the following key contributions:

- *A unified adaptation–repair framework,* ***DShift-FRep****,* addressing fairness degradation under hybrid distribution shifts in both tabular and image data.
- *Shift-vector-based alignment* that harmonizes data distributions while preserving feature semantics—offering a lightweight, interpretable alternative to adversarial alignment.
- *Multi-objective PSO fairness repair*, a targeted optimization mechanism that modifies fairness-localized weights, jointly improving fairness, accuracy, and domain stability.
- *Empirical evaluation,* demonstrating that `DShift-FRep` outperforms baseline methods in maintaining fairness and generalization under domain shifts.

8.2 Proposed Approach

The goal of this study is to address fairness issues under distribution shifts in binary and multi-class classification tasks, specifically focusing on scenarios where sensitive attributes (e.g., gender, race) may either align with or differ from the target label. This work extends the previous `DShift-FRep` method, which tackled fairness-aware fault localization and repair of biased behavior in DNNs. While `DShift-FRep` focused on bias mitigation in a single domain

(source data), `DShift-FRep` expands the approach to handle distribution shifts between source and target domains and generalizes the method for both image and tabular datasets.

Input Setup and Feature Extraction We consider a cross-domain setting involving a labeled source domain and a partially labeled target domain. The source domain provides abundant, yet potentially biased, labeled data, while the target domain consists primarily of unlabeled data, supplemented by a small set of labeled target anchors—samples that are carefully annotated to represent trustworthy reference points for label propagation and fairness calibration.

Formally, building on the formulation in Section 2.6, let the source and target domains be denoted as $D_{\text{src}} = \{(x_i^{\text{src}}, y_i^{\text{src}}, a_i^{\text{src}})\}_{i=1}^{N^{\text{src}}}$ and $D_{\text{tgt}} = \{(x_j^{\text{tgt}}, a_j^{\text{tgt}})\}_{i=1}^{N^{\text{tgt}}}$, respectively. We assume that there exists a alignment transformation

$$\mathcal{T} : \mathcal{X}_{\text{src}} \to \mathcal{X}_{\text{tgt}}$$

which generalizes the idea of a transport map, but is not restricted to optimal transport. Here $\mathcal{T}$ serves as a data alignment operator ensuring that the transformed source distribution approximately matches the target domain:

$$T_\psi P_{\text{src}} \approx P_{\text{tgt}}$$

To enable label propagation and fairness-aware repair,

- A reference set, consisting of the labeled source domain samples– abundant but potentially biased or noisy labels (towards target domain), and a small set of labeled target anchors, drawn from the target domain but annotated with trusted labels.

$$\mathcal{X}_{\text{ref}} = \mathcal{X}_{\text{src}} \cup \mathcal{X}_{\text{tgt}}^{\text{anc}},\ Y_{\text{ref}} = Y_{\text{src}} \cup Y_{\text{tgt}}^{\text{anc}},\ \mathcal{A}_{\text{ref}} = \mathcal{A}_{\text{src}} \cup \mathcal{A}_{\text{tgt}}^{\text{anc}}$$

- An unlabeled target set, $X_{\text{ulab}} = \mathcal{X}_{\text{tgt}} \backslash \mathcal{X}_{\text{tgt}}^{\text{anc}} = \{x'_1, \cdots, , x'_n\}$ with corresponding sensitive group $\mathcal{A}_{\text{ulab}} = \{a'_1, \cdots, a'_n\}$, for which the pseudo-labels Y'_{ulab} will be inferred.

In this study, we consider both image and tabular data.

- **Image data:** For image inputs $x \in \mathbb{R}^{H \times W \times C}$, we first extracted semantic feature embeddings using a pretrained encoder $f_\Phi(\cdot)$ (e.g., VGG, or domain-specific encoder), as:

$$z = f_\Phi(x)$$

- **Tabular data:** When $x \in \mathbb{R}^d$ (structured features), this extraction step is optional since the raw features already represent a semantically meaningful space.

General problem The primary challenge is to mitigate biased behavior in DNNs, particularly cases where certain sensitive groups are systematically misclassified or disadvantaged [183–185]. The specific objectives include: 1. capture and control for cross-domain distributional variations, 2. ensure fair behavior across both source and target domains. 3. identify and repair biased neuron weights responsible for such disparities. 4. improve target accuracy while preserving source performance.

Solution overview (`DShift-FRep`) We adopt a systematic, step-wise pipeline – from label propagation and representation alignment to localized fault localization and targeted weight repair – to robustly handle hybrid shifts while enforcing fairness constraints.

Solution scope; We consider hybrid shift scenarios in which distribution shifts may typically involve a single or mix of covariant shift (changes in $P(\mathcal{X})$), concept shifts (changes in $P(Y \mid \mathcal{X})$), and changes in the joint distribution $P(\mathcal{X}, Y)$.

An overview of `DShift-FRep` is shown in Figure 8.3, comprises six main phases.

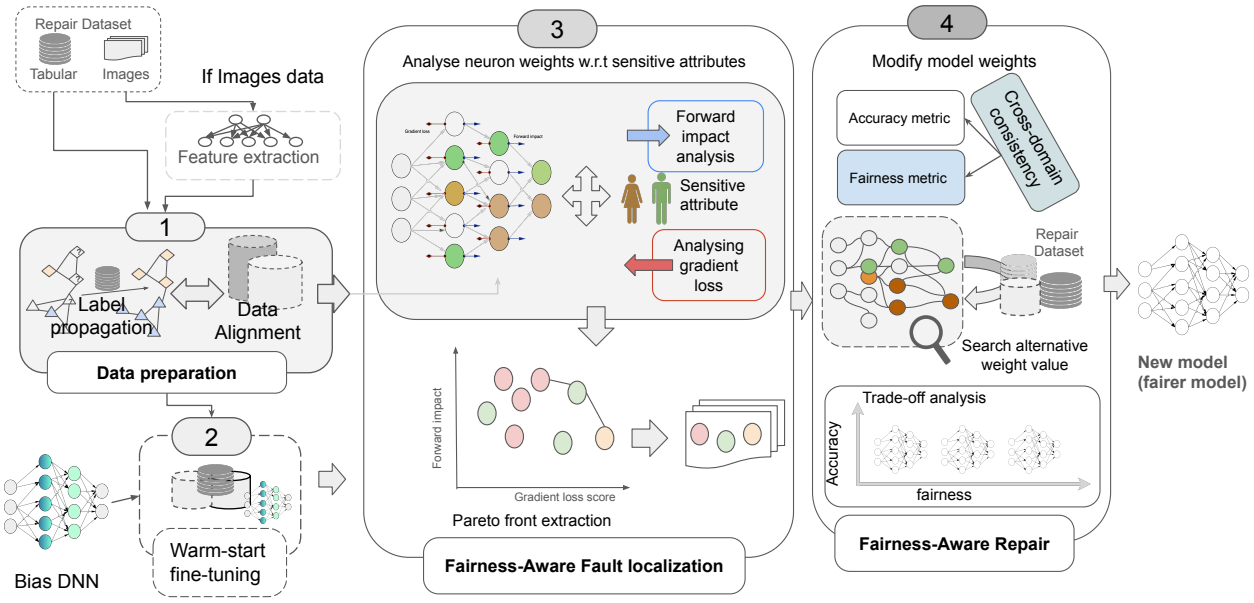


Figure 8.2 Overview of `DShift-FRep`

1. **Repair data preparation:** This step consists of two sub-steps, including label propagation and Data Alignment. Given the repair dataset, we ensure all the data has labels while also mitigating domain shift. We employ label propagation and representation alignment between the biased model's source and target domains. The data are passed to the label propagation and data alignment algorithms. This step ensures that the latent spaces are approximately aligned and that data points with similar semantic meaning have consistent representations, supporting fairness generalization.

2. **Warm-Start Fine-Tuning:** The pre-trained biased model is then warm-start fine-tuned ($\leq$ 5 Epochs) with the aligned data. This process transfers learned representations while adapting to new data distributions, serving as the initialization point for subsequent fairness repair. Warm-starting ensures that the model preserves useful knowledge and minimizes the cross-domain divergence while allowing targeted correction.

3. **Fairness-Aware Fault Localization:** In this phase, we perform neuron-level analysis to identify weight parameters contributing to biased decisions, leverages both the transformed (aligned) data and the original data. The method employs a combined analysis of gradient loss and forward impact to identify the most influential and biased neuron weights. A scoring function prioritizes weights contributing to unfair behavior, especially for deprived groups. Pareto front extraction helps select weights that are optimal in both metrics.

4. **Fairness-Aware Repair:** Targeted weight modifications (weights identified during fault-localization) are performed using an optimization mechanism (e.g., PSO) to search for fairer alternatives. The repair process jointly optimizes: Fairness metrics (e.g., *EOD*), and Cross-domain consistency, ensuring that fairness improvements in one domain do not degrade performance in another. A trade-off analysis is performed to balance fairness gains with accuracy retention, producing a fairer and stable set of neuron weights.

New Model (Fairer Model) The optimized weights are integrated into the model, yielding a fairer neural network that exhibits improved fairness and accuracy robustness across distribution shifts.

8.2.1 Repair data preparation

Let $\mathcal{X}_{\text{src}}$ and $\mathcal{X}_{\text{tgt}}$ denote the source and target feature spaces, with corresponding sensitive attributes $\mathcal{A}_{\text{src}}$ and $\mathcal{A}_{\text{tgt}}$, and labels Y_{src} (known) and Y_{tgt} (unknown). Under the assumption

of separability [71,235,236], each domain can be partitioned into into non-overlapping groups:

$$\mathbf{S} = \{\mathbf{S}_{y,a} \mid y \in Y_{\text{src}}, a \in \mathcal{A}_{\text{src}}\}, \mathbf{T} = \{\mathbf{T}_{y,a} \mid y \in Y_{\text{tgt}}, a \in \mathcal{A}_{\text{tgt}}\}$$

Here, $\mathbf{S}_{y,a}$ and $\mathbf{T}_{y,a}$ are the subsets of the source and target domains corresponding to label $Y = y$ and sensitive attribute $\mathcal{A} = a$:

$$\mathbf{S}_{y,a} \sim P(\mathcal{X}_{\text{src}} \mid Y_{\text{src}} = y, \mathcal{A}_{\text{src}} = a), \mathbf{T}_{y,a} \sim P(\mathcal{X}_{\text{tgt}} \mid Y_{\text{tgt}} = y, \mathcal{A}_{\text{tgt}} = a)$$

These partitions form the basis for fairness-aware label propagation (Section 8.2.1) and domain alignment (Section 8.2.1).

Few-Short Fairness-Aware Label propagation

This section presents a non-parametric [1], few-shot, fairness-aware label propagation technique that aims to extend supervision from a small set of labeled target examples (anchors) and abundantly labeled source to unlabeled target samples by propagating soft labels across a similarity structure (e.g., a kNN graph or cluster neighborhood).

Notation Let $x'_1 \in X_{\text{ulab}}$ denote an unlabeled target sample whose pseudo-label is to be inferred, and $x_j \in \mathcal{X}_{\text{ref}}$ denote a labeled reference sample drawn either from the source domain or from the labeled target anchors. Each sample x is associated with a sensitive group label a (e.g., gender or race) and a true or propagated class label y. Thus, a_i and a_j represent the sensitive attribute values corresponding to x'_i and x_j, respectively, while y_j is the label of x_j. The neighborhood $\mathcal{N}_i$ refers to the set of the top-k most similar reference samples to x'_i under a distance metric $d(\cdot, \cdot)$. The similarity between any two samples is denoted $s(i, j)$.

For each x'_1, we identify a local neighborhood from x_j based on any affinity-based method (e.g., nearest-neighbor graph, Gaussian Mixture clustering, or density-based partitioning), consisting of its most similar reference points under a distance metric $d(\cdot, \cdot)$. The base similarity is defined as:

$$s(i,j) = \begin{cases} 1 - d(x'_i,\, x_j), & \text{if cosine metric,} \\ \frac{1}{1 + d(x'_i, x_j)} & \text{if Eucledian} \end{cases} \tag{8.1}$$

[1] it does not train a new model but rather infers pseudo-labels through geometric propagation and fairness-aware reweighting.

For tabular data, raw features $x \in \mathbb{R}^d$ are directly used to compute similarities. For image data, we first extract semantic embeddings using a pretrained backbone (e.g., ResNet, VGG, or domain-specific encoder), so that $x = f_{enc}(I)$ represents a high-level feature vector capturing both domain-invariant and fairness-relevant characteristics.

- **Anchor Boosting:** The neighbor of a labeled anchors from the target domain receive higher influence through a boost factor $\beta > 1$

$$w^{(ab)}(i,j) = \begin{cases} \beta(s(i,j)), & \text{if } x_j \text{ is is a target anchor,} \\ (s(i,j)) & \text{otherwise} \end{cases} \tag{8.2}$$

  This counteracts source-domain dominance and supports adaptation under distribution shift.

- **Neighbor Consistency (NC):** Neighborhood reliability is enforced by emphasizing internally consistent samples. Let l be the known class label and $\hat{l}$ the neighborhood-predicted label. Then:

$$w^{(nc)}(i,j) = \begin{cases} (1+\gamma_{nc})w^{(ab)}(i,j), & \text{if } l = \hat{l}, \\ (1-\gamma_{nc})w^{(ab)}(i,j) & \text{otherwise} \end{cases} \tag{8.3}$$

  This encourages local label smoothness and penalizes uncertain or mislabeled references.

- **Intra-group Continuity (IC):** To promote fairness, we introduces intra-group coherence while preventing group overfitting. Given sensitive group membership:

$$w^{(ic)}(i,j) = \begin{cases} (1+\gamma_{ic})w^{(nc)}(i,j), & \text{if } a'_i = a_j, \\ (1-\frac{\gamma_{ic}}{2})w^{(nc)}(i,j) & \text{otherwise} \end{cases} \tag{8.4}$$

  This step maintains balance in representation by moderating the dominance of majority groups in each neighborhood.

Finally, the weights are normalized as:

$$\hat{w}(i,j) = \frac{w^{(ic)}(i,j)}{\sum_{k \in N_i} w^{(ic)}(i,k)} \tag{8.5}$$

and propagated soft labels are computed as:

$$y'_i = \sum_{j \in N_i} \hat{w}(i,j) y_j \tag{8.6}$$

The hard label assignment and confidence follow:

$$\hat{l}_i = \arg\max_c y'_{i,c}, \quad conf(x'_i) = y'_{i,\hat{l}_i} \tag{8.7}$$

Samples with confidence above a threshold τ are chosen for the next stage of repair, where as sample bellow the threshold are either discarded or re-evaluated via local consensus voting.

Control Domain shift Via Data Transformations

To mitigate hybrid distribution shifts, `DShift-FRep` applies a cluster- and shift-based alignment strategy that maps source and target features into a fairness-consistent latent space, while optionally separating shared and domain-specific components.

The goal is to learn non-parametric transformations:

$$\Phi_{\text{src}} : \mathcal{X}_{\text{src}} \to \mathcal{Z}, \;\; \Phi_{\text{tgt}} : \mathcal{X}_{\text{tgt}} \to \mathcal{Z},$$

that produce aligned embeddings $\mathcal{Z}$, minimizing cross-domain discrepancy while preserving intra-group structure defined by sensitive attributes $\mathcal{A}$. This addresses: 1. Domain alignment—reducing distributional shifts between D_{src} and D_{tgt}, and 2. Fairness preservation—ensuring transformations do not amplify disparities across sensitive groups $a \in \mathcal{A}$ (e.g., gender, race).

◎ Solution overview

When underlying nuisance factors are unknown or difficult to measure, clustering serves as a heuristic to approximate these latent factors. Incorporating fairness considerations during clustering enables alignment across domains without requiring explicit knowledge of all distributional variations.

Clustering for Localized Alignment

We employ clustering and localized mapping technique as a proxy to minimize distributional discrepancies while adhering to fairness constraints.

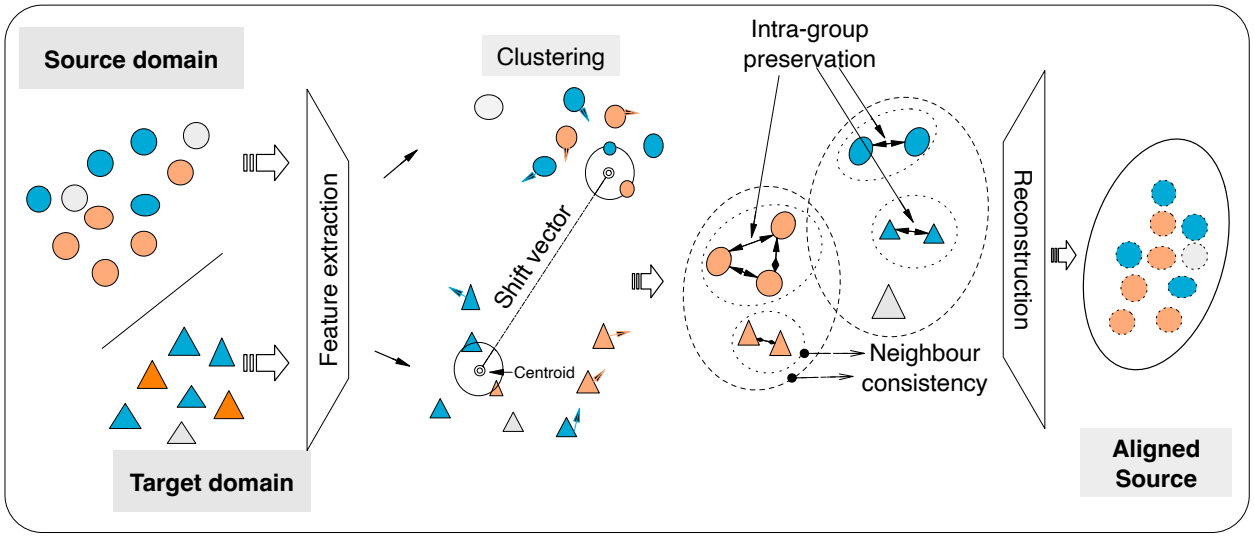


Figure 8.3 Overview of our fairness-aware data alignment approach. Under distribution shift, both domain-level and group-level misalignment emerge. We apply a hybrid alignment strategy that clusters the data and computes shift vectors only where sensitive groups are consistent across source and target. The result is an aligned feature space that preserves both fairness (IC) and data semantics.

Clustering partitions the feature space into K localized regions to approximate latent nuisance factors. For the source domain, clusters $\{C_k\}_{k=1}^{K}$ are identified using clustering techniques. Target samples are assigned to the nearest source cluster:

$$C_{id}(x_i^{\text{tgt}}) = \arg\min_k ||x_i^{\text{tgt}} - \mu_k^{\text{src}}||$$

where μ_k^{src} is the centroid of cluster C_k in the source domain.

Fairness-Aware Shift Estimation We introduce a shift vector as shift estimate to aligns the source and target distributions while respecting sensitive groups.

> *A shift vector, Δ_k, represents the average translation required to align features of the source and target domains within a cluster C_k. It captures the magnitude and direction of distributional displacement between corresponding clusters across domains.*

For each cluster C_k, the shift vector is computed as:

$$\Delta_k = \mu_k^{\text{tgt}} - \mu_k^{\text{src}}$$

where μ_k^{src} and μ_k^{tgt} are the centroids of cluster k in the source and target domains, respectively. If sensitive groups in the cluster do not match exactly, Δ_k is computed as a weighted average of subgroup means. Shift vectors are normalized and clamped to prevent extreme transformations:

$$\Delta_k \longleftarrow \frac{\Delta_k}{||\Delta_k||} \min(||\Delta_k||, 1)$$

Alignment Transformations Both domains are first partitioned into K clusters C_k based on feature similarity. Within each cluster, transformations are uniform, modeled as:

$$\phi(x_i; \theta) = \mathcal{T}_\theta(x_i) \pm \Delta_k \text{ for } x_i \in C_k$$

where $\mathcal{T}_\theta(\cdot)$ denotes an optional (but encouraged) normalization or augmentation function (e.g., Z-score normalization, CORAL whitening, random padding and cropping, RandAugment [237], or random perturbation for tabular data). The "+" operation applies to source samples ($x_i \in \mathcal{X}_{\text{src}}$ and the "−" operation applies to target samples ($x_i \in \mathcal{X}_{\text{tgt}}$).

After computing Δ_k, each sample is proportionally adjusted toward its counterpart domain, as:

$$\hat{x_i}^{\text{src}} = \alpha \cdot (\mathcal{T}_\theta(x_i^{\text{src}}) + \Delta_k), \;\; \hat{x_i}^{\text{tgt}} = \alpha \cdot (\mathcal{T}_\theta(x_i^{\text{tgt}}) - \Delta_k) \tag{8.8}$$

where $\alpha \in (0, 1)$ balances aligned vs. domain-specific preservation.

NI and IC considerations After applying $\phi(x_i; \theta)$, a smoothness constraint based on *NI* and *IC* are enforced to maintain fairness-consistent structure:

$$\|\phi(x_i; \theta) - \phi(x_j; \theta)\| \leq \|x_i - x_j\|, \;\; \forall (x_i, x_j) \in \mathcal{N}$$

ensuring local relationships are preserved across domains. Here, $\mathcal{N}$ denotes pairs of nearest neighbors within or across domains. Similarly, *IC* is employed to ensures that groups defined by sensitive attributes (e.g., privileged vs. unprivileged) exhibit smooth transformations, avoiding discontinuities that could introduce bias. For a group $\mathbf{S}_{y,a} \subset C_k$, intra-Group continuity ensures:

$$\|\phi(x_i; \theta) - \phi(x_j; \theta)\| \approx \|x_i - x_j\|, \quad \forall (x_i, x_j) \in \mathbf{S}_{y,a}$$

Optional Feature Decomposition For hybrid shifts, feature decomposition can separate shared and domain-specific components. The domain-specific component captures variations

unique to each domain.

$$\mathcal{X}_{dom}^{\text{src}} = \mathcal{X}_{\text{src}} - \mathcal{X}_{\mu}^{\text{src}}, \quad \mathcal{X}_{dom}^{\text{tgt}} = \mathcal{X}_{\text{tgt}} - \mathcal{X}_{\mu}^{\text{tgt}}$$

Where, $\mathcal{X}_{\mu}^{\text{src}}$ and $\mathcal{X}_{\mu}^{\text{tgt}}$ are the empirical means representing common features across the domain. A weighted combination of these components ensures that the transformation balances global alignment and domain. In this case, Eqn (8.8) becomes:

$$\begin{aligned} \hat{\mathcal{X}}_{\text{src}} &= \alpha \cdot (\mathcal{T}_{\theta}(\mathcal{X}_{\text{src}}) + \Delta_k) + (1-\alpha) \cdot \mathcal{X}_{dom}^{\text{src}} \\ \hat{\mathcal{X}}_{\text{tgt}} &= \alpha \cdot (\mathcal{T}_{\theta}(\mathcal{X}_{\text{tgt}}) - \Delta_k) + (1-\alpha) \cdot \mathcal{X}_{dom}^{\text{tgt}} \end{aligned} \tag{8.9}$$

where $0 < \alpha < 1$ controls the trade-off. This process harmonizes domain representations, mitigates fairness shifts, and prepares data for fairness-aware repair.

Warm-Start Fine-Tuning

Before initiating fault localization, the model undergoes warm-start fine-tuning (≤ 5 epochs) using the aligned dataset $\hat{D}$, which combines the aligned source and target data obtained from Section 8.2.1. This step is applied to the unfair model under repair—the pretrained model trained solely on source data—to improve its adaptability across domains.

The fine-tuning updates only the last few layers, which will later be targeted for fairness repair. This design provides a balance between representation stability and adaptability, enabling the model to assimilate structural cues from both source and target distributions without altering its foundational parameters.

Motivations and Benefits

- *Cost-effective and budget-friendly*, rather than retraining from scratch–which would be computationally expensive and risk erasing learned knowledge—warm-start fine-tuning provides a lightweight correction path. It reuses pretrained weights and applies minimal updates to achieve cross-domain alignment, making it suitable for fairness repair under limited computational budgets or incremental deployment.

- *Complementary adaptation–repair strategy*, this warm-start fine-tuning acts as an adaptation step that complements the subsequent repair phase. By aligning representations across domains, the model begins from a more balanced initialization, allowing fault localization to focus on unfair neurons rather than general domain mismatch. Conceptually, this "adapt-then-repair" sequence ensures that the repair process targets

fairness-relevant discrepancies rather than residual distributional noise.

- *Enhanced stability for fault localization*, where warm-start fine-tuning stabilizes gradients and reduces internal covariate shifts within the aligned feature space, improving the robustness and interpretability of subsequent gradient and forward-impact analyses. This results in more reliable identification of biased neurons under hybrid shifts.

Importantly, even after adaptation, the model is still treated as unfair–both in the target domain, where bias transfer may occur, and in the source domain, where original disparities persist during training. Thus, fine-tuning does not attempt to "solve" fairness issues but prepares the model for fairness repair by harmonizing representational subspaces. However, the used of fairness-aware aligned dataset for fine-tuning will help minimize both the distribution shift and avoid implying the bias.

◎ Motivation summary (Warm-start)
Warm-start fine-tuning act as a **cost-effective alternative to retraining**, offering a lightweight way to improve alignment while preserving learned knowledge. It complements the subsequent repair phase by following an *"adapt-then-repair"* strategy—first harmonizing distributions, then targeting unfair neurons. In addition, it stabilizes gradients and ensures that fault localization operates from a domain-calibrated but still fairness-sensitive initialization.

Implementation Details. During warm-start fine-tuning, all layers except the last few are frozen to preserve previously learned representations. A small epoch training is performed on the aligned dataset $\hat{D}$, which includes both aligned source data (from the original source training set) and aligned target data (assumed abundant—proportional to the source set but mostly unlabeled) from Section 8.2.1. For the unlabeled target data, pseudo-labels y' obtained through fairness-aware label propagation (Section 8.2.1) are used, selecting only samples with return soft-label confidence $\tau \geq 90\%$. In addition, a small subset of originally labeled target data $(x_{\text{ref}}^{\text{tgt}}, y_{\text{ref}}^{\text{tgt}})$ is included to reinforce reliable supervision. The resulting fine-tuning set therefore combines $(\hat{x}_{\text{src}}, y_{\text{src}})$, $(\hat{x}_{\text{tgt}}, y')$, and $(x_{\text{ref}}^{\text{tgt}}, y_{\text{ref}}^{\text{tgt}})$, providing balanced exposure to both domains while maintaining fairness sensitivity.

8.2.2 Fault Localization under Distribution Shifts

Building on the `FairFLRep` framework for fairness repair of DNNs [15], we extend fault localization to cross-domain scenarios where labeled source data and unlabeled target data

may exhibit hybrid distribution shifts. The key adaptation is the integration of information from both domains to identify biased neurons while prioritizing misclassified examples in underprivileged groups.

We use two complementary datasets:

- *Aligned data ($\hat{D}$)* – obtained after applying the transformation $\mathcal{T}_\theta(x;\theta)$ in Section 8.2.1. It emphasizes structural alignment between domains and is used to identify fairness-related faults in deeper layers, where representational disparities emerge due to hybrid shifts.
- Unaligned (default repair) data (D_{rep}) – the version containing propagated labels without alignment. It is used to detect prediction-level biases in the output layer, where disparities appear directly at the decision boundary.

This dual-view analysis ensures that fairness repair targets both internal (representation-level) and external (prediction-level) faults.

Subgroup Categorization Samples are grouped by sensitive attribute $a \in \mathcal{A}$ and correctness of prediction:

1. $D_{pos,a}$: correctly classified samples of group a,
2. $D_{neg,a}$: misclassified samples of group a.

These subgroups are analyzed separately for source and target domains, and the results are aggregated.

Fairness-Stratified Sampling (Repair sets)

To ensure that model repair is both fairness-sensitive and representative, we adopt a Fairness-Stratified Sampling (FSS) strategy that balances correctly and incorrectly classified samples while emphasizing deprived-group errors. Given ground-truth labels y, predictions $\hat{y}$, and sensitive attribute $a \in \{a_0, a_1\}$ (where a_0 denotes the deprived group and a_1 the favored group), the data are partitioned into four mutually exclusive subgroups:

$$D_{y|a_0} = \{(x,a,y) : a = a_0, y = \hat{y}\},\ D_{\neg y|a_0} = \{(x,a,y) : a = a_0, y \neq \hat{y}\},$$
$$D_{y|a_1} = \{(x,a,y) : a = a_1, y = \hat{y}\},\ D_{\neg y|a_1} = \{(x,a,y) : a = a_1, y \neq \hat{y}\}.$$

Here, $D_{y|a_0}$ and $D_{\neg y|a_0}$ represent correctly and incorrectly classified instances from the deprived group, while $D_{y|a_1}$ and $D_{\neg y|a_1}$ denote the same for the favored group.

Each subgroup's sampling priority is governed by the fairness-weighting factor $W^{(d)}(a)$ defined in Eqn (8.10), which encodes the magnitude of deviation $|\delta^d_{y,a}|$ between expected and observed prediction rates. Larger $W^{(d)}(a)$ values correspond to subgroups exhibiting stronger bias and therefore receive proportionally higher sampling weight. Specifically, FSS strategy proceeds as follows:

1. *Error prioritization* – all deprived misclassified samples ($D_{\neg y|a_0}$) are retained to preserve deprived-group error signals.

2. *Weighted subgroup balancing* — misclassified favored samples, $D_{\neg y|a_1}$, are resampled according to their relative weights $W^{(d)}_{neg}(a_1)$ and $W^{(d)}_{neg}(a_0)$, ensuring that:

$$D^*_{\neg y|a_1} : D^*_{\neg y|a_0} \approx W^{(d)}_{neg}(a_1) \times |D_{\neg y|a_1}| : W^{(d)}_{neg}(a_0) \times |D_{\neg y|a_0}|$$

   If $D_{\neg y|a_1}$ contains too few samples, additional instances are drawn (with replacement if necessary) from the same group to reach the weighted target size.

3. *Correct-instance balancing* – to maintain approximate parity between total correct and misclassified sets,

$$|(Correct),\, D^*_y| \approx |(Misclassified),\, D^*_{\neg y}|$$

   the deprived correct samples ($D_{y|a_0}$) are first fully included, and additional favored correct samples $D_{y|a_1}$ are drawn proportionally to their fairness weights until the overall balance criterion is met.

For subgroup completeness, at least one instance from each subgroup ($D_{\neg y|a_0}, D_{\neg y|a_1}, D_{y|a_0}, D_{y|a_1}$) is guaranteed, ensuring representational coverage for all fairness categories. Overall, FSS achieves three fairness-aware objectives simultaneously: 1. Error emphasis – misclassifications from deprived groups are preserved or amplified; 2. Cross-group parity – subgroup proportions remain approximately balanced; and 3. Repair readiness – sampled sets contain sufficient diversity for fairness-aware fault-localization and repair.

Subgroup Bias Estimation

We estimate subgroup-level deviations using model predictions and sensitive attributes for both source D_{src} and target D_{tgt} domains (before FSS sampling is applied). Each dataset

$D = \{x_i, y_i, a_i\}_{i=1}^{\|D\|}$ consists of inputs x_i, their true or propagated labels y_i, and sensitive attribute values $a \in \mathcal{A}$. Let $f(x_i)$ denote the model's predicted label for input x_i. Following `FairFLRep`, we compute expected and observed prediction rates:

$$P^*(y) = \frac{1}{\|D\|} \sum_{i=1}^{\|D\|} 1(f(x_i) = y_i), \;\; \hat{P}(y|a) = \frac{1}{\|D_a\|} \sum_{i, a_i = a} 1(f(x_i) = y_i)$$

where,

- $P^*(y)$, denote the global expected prediction rate for y.
- $\hat{P}(y|a)$, is the group-specific observed rate for group a.
- $1(f(x_i) = y_i)$ is prediction correctness; 1 if the prediction is correct and 0 otherwise.
- $D_a = \{(x_i, y_i, a_i) \in D : a_i = a\}$ and $\|D_a\|$ is its size.

The subgroup deviation under domain $d \in \{\text{src}, \text{tgt}\}$ is computed as:

$$\delta_{y,a}^d = P^*(y) - \hat{P}(y|a)$$

and the corresponding fairness weighting factors are:

$$W^{(d)}(a) = \frac{\sum_y |\delta_{y,a}^d|}{\sum_{a'} \sum_y |\delta_{y,a'}^d|} \tag{8.10}$$

A larger $|\delta_{y,a}^d|$ implies stronger deviation from equitable prediction rates — indicating that subgroup (y, a) is disproportionately affected by the model's bias. These normalized weights guide fairness-aware prioritization when computing gradient loss and forward impact during neuron fault localization. For simplicly, we will use $W_{pos}^{(d)}(a)$ and $W_{neg}^{(d)}(a)$ to refers to the weighing factors computed from correct and miss-classified subsets of the data, respectively.

Neuron-Level Influence Analysis

Each neuron weight $w^{(l)}$ is evaluated based on its contribution to fairness disparities using two complementary influence measures:

- *Gradient-based influence,* measuring sensitivity of loss for subgroup a w.r.t. weight $w^{(l)}$.
- *Forward-impact*, measuring the impact of w^l on model the model prediction outcome.

For deeper layers, both gradients and forward-impact scores are computed using aligned features ($\hat{D}$) to reflect fairness disparities embedded in internal representations. For output (last) layers, computations rely on unaligned data D_{rep} to capture decision-level disparities.

Fairness-Aware Scoring and Selection A unified fairness-aware importance score is computed as:

$$score^{d}(w^{l}) = W_{neg}^{(d)} \cdot \frac{gf(w^{l}|D_{a_0})}{1 + gf(w^{l}|D_{a_1})} - W_{pos}^{(d)} \cdot \frac{gf(w^{l}|D_{a_1})}{1 + gf(w^{l}|D_{a_0})} \tag{8.11}$$

where,

- gf – denotes the aggregated value of influence function (gradient or forward-impact) computed on weight w^l over samples in domain subset d;
- $W_{neg}^{(d)} \leftarrow \frac{W_{neg}^{(d)}(a_0)}{W_{neg}^{(d)}(a_1)}$ – computed using Eqn (8.10), is a fairness emphasis weight for the unprivileged group, a_0, within miss-classified sample in domain d.
- $W_{pos}^{(d)} \leftarrow \frac{W_{pos}^{(d)}(a_1)}{W_{pos}^{(d)}(a_0)}$ – is a balancing weight for the privileged group, from correctly classified samples in d.
- D_{a_*} – is domain subset corresponding to subgroup $a \in \mathcal{A}$.

The subtraction ensures the localization process favors selecting neuron weights that leads to misclassification disparity by amplifying their effect on disadvantaged samples while dampening influence from advantaged samples.

The global score integrates both domains:

$$score(w^{l}) = \beta \cdot score^{(\text{src})}(w^{l}) + (1 - \beta) \cdot score^{(\text{tgt})}(w^{l})$$

with $\beta \in [0, 1]$ controlling the emphasis on source vs. target contributions. Neurons with higher scores indicate stronger contribution to biased or unfair predictions, particularly against deprived groups.

Candidate Selection via Pareto Front Weights, w_r are ranked on the Pareto front of global gradient loss vs. global forward impact, as in `FairFLRep`. The key difference is that both measures now incorporate source and target contributions, ensuring neurons selected for repair address cross-domain fairness violations.

In summary, our approach extends `FairFLRep` by:

- Incorporating subgroup bias estimation across both domains;
- Introducing a cross-domain global score combining source and target effects;
- Extending fault localization beyond the last layer to include deeper-layer fairness faults using aligned data.
- Misclassified samples in underprivileged groups are prioritized according to their domain-specific weights.

◎ Key extension
We consider both deeper and last-layer. Gradient loss and Forward impacts within the deeper-layer are computed using aligned source and target samples (Section 8.2.1) with pseudo-label (for target) while the last layer considers original repair data. Gradient loss and forward impacts are aggregated across domains before Pareto selection.

These modifications allow fault localization to operate effectively under distribution shifts while maintaining the core principles of `FairFLRep`. The selected neurons constitute the faulty subset to be updated in the subsequent fair-aware repair phase.

8.2.3 Cross-Domain Fairness-Aware Repair

The repair phase modifies the suspicious neuron weights $\{\mathbf{w_r}\}$ identified in Section 8.2.2 so the model becomes fairer across both the source and target domains while preserving accuracy.

Fairness-Aware Repair in Classical DNN

For classical DNN, repair is implemented as a multi-objective population search using PSO within the local weight neighborhoods discovered during fault localization.

Overview of Optimization Process The repair process employs PSO to search for improved configurations of the suspicious weights. Each particle encodes a candidate repair—a vector of neuron weights—and the swarm collectively explores the local parameter space. The search is guided by fairness, accuracy, and optional domain-alignment objectives.

Particles are initialized from Gaussian distributions centered on existing sibling weights in the same layer to encourage small but meaningful perturbations around the original parameters. This ensures model semantics and correctly classified predictions are preserved as much as possible.

Fairness objectives Fairness is evaluated using both in-domain and cross-domain metrics to ensure that the model remains equitable within each domain and consistent across domains. For the source and target domains, $d \in \{\text{src}, \text{tgt}\}$, standard group-fairness metrics such as *EOD* are computed as: $EOD = P_{\text{src}}(\hat{Y} = 1|\mathcal{A} = 1) - P_{\text{src}}(\hat{Y} = 1|\mathcal{A} = 0)$.

The objectives are defined as follows (treated separately, not collapsed): 1. *Fairness (to minimize)*, measured on both source and target domains using group-fairness metrics (e.g., *EOD*, *SPD*) aggregated into $F_{val} = \frac{F_{\text{src}}+F_{\text{tgt}}}{2}$; 2. *Accuracy (to maximize)*, computed via per-sample Arachne-style scores over correctly classified and repaired samples,

$$A = \frac{(n_{intact} + 1)}{loss_{pos} + 1} + \frac{(n_{patched} + 1)}{loss_{neg} + 1}$$

where, n_{intact} – number of correctly classified (intact) samples; $n_{patched}$ – number of repaired misclassifications; $loss_{pos}$ – task loss on the intact subset; $loss_{neg}$ – task loss on the patched subset. Cross-domain consistency is enforced by combining domain accuracies as

$$A(x) = A_{\mu}(x) * (1 - \Delta A(x)),$$

where $A_{\mu}(x)$ is the normalized mean accuracy across domains and $\Delta A(x)$ is the normalized accuracy gap. 3. *Domain consistency (to minimize)*, D_{val}, quantifies divergence between source and target latent features (e.g., via centroid distance or MMD), encouraging consistent decision boundaries across domains.

Image models – head-only repair For image tasks, we split the biased model into: a frozen feature extractor (encoder) producing aligned embeddings (used in Section 8.2.1), and a classification head (lightweight fully-connected layers). PSO mutates only the classification head parameters (the same layer objects the full model uses). Implementation patterns, include PSO runs on a temporary head model; after PSO terminates, mutated head weights are copied back to the corresponding layers of the full model. This keeps the feature extractor frozen and avoids expensive re-evaluation of deep weights. This head-only design drastically reduces search dimension and runtime (typical speedups of orders of magnitude), because the head input size is small compared to full image tensors. For tabular data head-only splitting is optional because the original input shape typically matches the model head; PSO can operate directly on the model layers.

Particle & velocity update (stable constricted PSO) The velocity of each particle is adjusted in each iteration based on its distance from personal best position (pbest) and

global best position (gbest), ensuring that particles move toward a combination of their own best experience and the swarm's collective best experience, based on classical PSO logic but integrates both fairness (to minimize) and accuracy (to maximize) objectives, plus the domain-consistency term, as follows:

$$pbest_i = \begin{cases} x_i^{t+1}, & \text{If } \left(\mathcal{F}^{(x_i^{t+1})} \leq \mathcal{F}^{(pbest_i)} \text{ AND } A^{(x_i^{t+1})} \geq A^{(pbest_i)}\right) \\ & \text{and } \left(\mathcal{F}^{(x_i^{t+1})} < \mathcal{F}^{(pbest_i)} \text{ OR } A^{(x_i^{t+1})} > A^{(pbest_i)}\right) \\ pbest_i, & \text{otherwise} \end{cases}$$

where:

- x_i^t and x_i^{t+1} are the current and updated positions of particle i.
- $pbest_i$ is the personal-best position of particle i;
- $\mathcal{F}(\cdot)$ is the aggregated fairness penalty (e.g., $\frac{1}{2}\mathcal{F}_{\text{src}} + \mathcal{F}_{\text{tgt}}$;
- $A(\cdot)$ is the cross-domain accuracy score,

$$A(x) = A_\mu(x) * (1 - \Delta A(x)),$$

  with $A_\mu(x)$ is normalized mean accuracy and $\Delta A(x)$ is the normalized accuracy gap between source and target domain for of particle i

In the Equation, the first condition ensures that no update is accepted if fairness becomes worse or accuracy drops. The second (strict) condition ensures the update happens only if at least one metric improves. This acts as a Pareto-dominance rule, maintaining both fairness and accuracy fronts without collapsing them into a single scalar early in search.

The logical conjunction enforces that a new position replaces the current best only when fairness does not worsen and accuracy does not decrease, with a strict improvement in at least one of them. The global best *gbest* is updated similarly, comparing each particle's new position against the current global best.

If scalar ranking is required ((e.g., to pick the current *gbest*)), the normalized composite score

$$S(x) = \alpha A(x) - (1 - \alpha)\mathcal{F}(x),\ \alpha = 0.6$$

is used purely for ordering candidates when ties occur, preserving emphasis on accuracy during late repair stages. Because the PSO uses Pareto updates internally, this final scalar is used only to choose the swarm leader when required (not to collapse the search objectives).

At each candidate evaluation, 1. Temporarily set the particle weights on the head (in full model or head model). 2. Run a forward pass for the relevant sets: $D^{\text{src}}_{pos}, D^{\text{src}}_{neg}, D^{\text{tgt}}_{pos}, D^{\text{tgt}}_{neg}$ – constructed from D_{rep} (unaligned repaired set) and $\hat{D}_{rep}$ (aligned set) as appropriate (unaligned for last-layer/evaluation of decision-level bias; aligned for deeper-layer consistency). 3. Compute per-sample losses and Arachne scores; aggregate into A_{src}, A_{tgt} and acc_score.

$$A = \frac{(n_{intact} + 1)}{loss_{pos} + 1} + \frac{(n_{patched} + 1)}{loss_{neg} + 1}$$

where, n_{intact} – number of correctly classified (intact) samples; $n_{patched}$ – number of repaired misclassifications; $loss_{pos}$ – task loss on the intact subset; $loss_{neg}$ – task loss on the patched subset. From which $A_{\mu}(x)$ and $\Delta A(x)$ are then computed normalized by the total samples.

Compute fairness metrics $\mathcal{F}_{\text{src}}$, $\mathcal{F}_{\text{tgt}}$ (e.g., *EOD*, *SPD*) and domain penalty D_{val} from latent features (feature extractor outputs).

Stopping criteria and stability Stop after maximum PSO iterations or if no Pareto improvement in a patience window. Because the search is restricted to local neighborhoods (Gaussian initialization, velocity bounds) and only head layers for images, the algorithm favors stable, small corrections, not extreme patches that accidentally flip many decisions. The explicit acc_gap term encourages solutions that generalize similarly across domains.

Performance optimizations

- Head-only PSO for images yields dramatic speedups (orders of magnitude) because the head input shapes are small; ensure head input shape matches the frozen feature extractor output used in alignment. Inserting a small bottleneck adapter (trainable) between extractor and head helps absorb representational mismatch– optional but recommended when alignment transforms changed embedding output shape.
- feature extractor outputs are precompute once per candidate population where possible (if head-only in-place edits do not change feature extractor outputs). This avoids repeated deep forward passes.
- Copy-back policy: PSO runs on a temporary head object, and the mutated head weights are copied into the full model at the end (layer-wise mapping).
- Tabular models: head-only trick is optional on tabular data—PSO can operate on the same layers because input shapes are small.

8.3 Experimental Design

To comprehensively evaluate `DShift-FRep`, we conduct experiments on both tabular and image datasets, as well as on synthetic data designed to simulate controlled shift scenarios. These experiments collectively assess the method's ability to sustain fairness and accuracy under covariate, label, and hybrid distribution shifts.

8.3.1 Research questions (RQs)

The evaluation of `DShift-FRep` is guided by the following five research questions (RQs):

RQ1: **Can fairness-aware repair approaches maintain fairness under distribution shifts?** This question assesses whether methods developed for static datasets remain effective when covariate, concept, or joint distribution shifts occur, and whether `DShift-FRep` can identify and mitigate unfair behavior across sensitive groups under these hybrid shifts, while highlighting which fairness metrics it handles most effectively.

RQ2: **How does `DShift-FRep` perform when applied to datasets with varying shift magnitude and bias severity?** This question examines the scalability and generalization of `DShift-FRep`, assessing its robustness on datasets of different complexities and levels of distribution and bias drift.

RQ3: **To what extent does the alignment improve accuracy across both domains?** This RQ evaluates whether the proposed label propagation and shift-based alignment preserve data structure, enforce fairness, and improve prediction consistency in the target domain.

RQ4: **How does `DShift-FRep`, adapt-then-repair approach compares to standard fairness-aware domain adaptation approaches in cross-domain stability?** This RQ investigates whether combining gentle adaptation with targeted repair reduces the risk of fairness forgetting or overwriting while maintaining predictive performance.

8.3.2 Dataset and Models

Tabular Data

We first evaluate `DShift-FRep` using UCI Adult and Folktables datasets, which are widely adopted for fairness benchmarking under real-world demographic and temporal variations.

Datasets and Setup The Adult dataset consists of $48,842$ records from the U.S. Census, containing demographic and occupational features such as education, age, and work class. The goal is to predict whether an individual's annual income exceeds $\$50K$, with gender (male or female) treated as the sensitive attribute. We use $30,000$ instances for training and the remaining for testing. Categorical attributes are one-hot encoded and concatenated with normalized continuous features, followed by data whitening for stable learning.

To simulate realistic temporal shifts, we use the Folktables dataset, a modern extension of the U.S. Census data that provides yearly subsets. Specifically, we adopt data from 2015 and 2018 as distinct temporal domains. This setup mirrors real-world scenarios where models trained on past data encounter population changes over time.

Shift Analysis and Motivation We further extend our evaluation by partitioning the Folktables dataset across U.S. states. Chen, et al. [73] quantified the degree of covariate and label shifts based on conditional independence–based divergence test [238], and showed that that features within the data such as class of worker, hours worked per week, sex, and race exhibit significantly larger shifts (on the order of 10^{-2}) than the label variable (10^{-4}), confirming the predominance of covariate shift in cross-state comparisons. Conversely, within-state temporal shifts are dominated by label drift (≈ 0.1 vs < 0.01 for covariates). This dual behavior—geographical covariate shift and temporal label shift—makes the dataset ideal for testing `DShift-FRep` 's ability to handle hybrid shifts.

Synthetic Data To gain theoretical insight into how fairness is affected by controlled distributional perturbations, we construct synthetic datasets using structural equation models (details in Appendix A), inspired by previous fairness–causality studies [70, 239].

Setup and Shift Simulation Based on setup in [70], we simulate binary variables using a logistic structural model where the sensitive attribute $\mathcal{A}_\gamma$ and target variable $\mathcal{T}_\gamma$ are connected through intermediate causal relations $\mathcal{R}$. Distribution shifts are introduced via soft interventions [239] that perturb the components of structural equations by a shift parameter Γ, effectively altering the marginal distributions of both $\mathcal{A}_\gamma$ and $\mathcal{T}_\gamma$. We generate 5 pairs of source–target domains, each containing $8,000$ samples. When γ increases (e.g., $\gamma = 15$), the representation of disadvantaged groups in the target domain rises from 0.5 to approximately 0.94, while class ratios decrease from 0.5 to 0.36. We train a 1D CNN on the source data (i.e., $\gamma = 0$), instead of traditional fully connected layers, leveraging Conv1D layers to extract feature patterns from tabular inputs, treating each sample as a 1D sequence of numerical features. The model is then evaluated under different $\gamma = \{1, 5, 10, 15, 20\}$ values to assess

fairness robustness as the magnitude of shift grows.

This setup allows `DShift-FRep` to be tested under controlled hybrid shifts, offering interpretable insights into how fairness–accuracy trade-offs evolve as nuisance factors change.

Image Data

We extend our evaluation to image classification tasks that experience both semantic (conceptual) and visual (covariate) shifts.

`CelebA` Dataset On the `CelebA` dataset, we train on 10,000 "young" face images and use 1,000 "young" images for in-distribution testing. To introduce shifts, we evaluate the model on:1. $1,000$ "not young" faces (age-based domain shift). 2. The same $1,000$ "young" faces with strong JPEG compression (quality-based covariate shift). CelebA includes over 200,000 images with 40 annotated attributes. The task is to classify gender while protecting sensitive features such as chubbiness and eyeglasses. Images are center-cropped and resized to 64×64 pixels, normalized to $[0, 1]$.

Cross-Dataset Shift (`UTKFace` → `FairFace`) To study fairness under heterogeneous image sources, we train on `UTKFace` and test on `FairFace`. Both datasets consist of facial images but differ significantly in demographic balance and acquisition quality. The target task is gender classification, with race as the sensitive attribute. We employ `VGG16` and `LeNet-5` architectures to evaluate both lightweight and deep model variants.

Synthetic 3D Shapes Additionally, we use a synthetic 3D Shapes dataset in [240], which provides complete control over six independent latent factors (shape, color, scale, orientation, wall hue, floor hue). Following [71], we define class (Y) as shape, sensitive attribute ($\mathcal{A}$) as color, and nuisance factor (D) as scale. We simulate four types of shifts: 1. `Sshift1` – Covariate shift: only D shifts (e.g., small → large objects). 2. `Sshift2` – Correlation shift: $(\mathcal{A}, Y)$ relationship changes (e.g., color–shape dependency differs). 3. `Dshift` – Domain shift: D sample space changes (e.g., only large objects in target). 4. `Hshift` – Hybrid shift: combination of `Sshift2` and `Dshift`. A multilayer perceptron (MLP) is trained to perform shape classification while `DShift-FRep` aims to maintain fairness across all scenarios.

8.3.3 Compared approaches

To evaluate the effectiveness of `DShift-FRep`, we compare it against both DNN repair techniques and domain adaptation / fair representation learning methods. The selected baselines

allow us to isolate the impact of (i) fairness-aware fault localization, (ii) fairness-aware repair, and (iii) explicit handling of distribution shift.

DNN repair techniques

`DShift-FRep`

This is our proposed adaptive fairness-aware repair framework that explicitly accounts for both fairness and distribution shift throughout the data alignment, fault localization, and repair phases. As described in Section 8.2.2, the repair process takes as input a repair dataset constructed from both the source and target domains. This dataset includes the original (unaligned) samples as well as their aligned counterparts obtained through the data alignment procedure described in Section 8.2.1. The repair dataset is further partitioned into misclassified and correctly classified samples to guide fairness-aware fault localization and constrained repair. Detailed information on the adaptation data construction is provided in Section 8.2.1 (Warm-Start Fine-Tuning).

`FairFLRep`

[15] It is a fairness-aware DNN repair method from which `DShift-FRep` is derived. In `FairFLRep`, both the fault localization and repair phases are explicitly optimized with respect to a specified fairness metric (e.g., *EOD*). Similar to `DShift-FRep`, `FairFLRep` uses a repair dataset composed of samples from both the source and target domains; however, it does not include aligned counterfactual samples. This comparison isolates the benefit of explicitly modeling distribution shift via alignment.

`MOON`

It is a spectrum-based fault localization technique that focuses on improving predictive accuracy rather than fairness [92]. Since `MOON` does not include a repair phase, we combine its fault localization component with the `FairFLRep` repair strategy. This variant allows us to evaluate the impact of accuracy-driven fault localization when paired with fairness-aware repair. The repair dataset is the same as that used for `FairFLRep` and `DShift-FRep`.

`Arachne`

It is a widely used DNN repair framework originally designed to correct misclassification errors [39]. Although not explicitly developed for fairness, the authors note its applicability

to fairness-related problems. We adapt Arachne to fairness repair by following the original guidelines and applying it to the same repair dataset used by `FairFLRep` and `DShift-FRep`, enabling a direct comparison.

`DeepFault`

It is another spectrum-based fault localization method that identifies suspicious neurons but does not include a repair mechanism [174]. Similar to `MOON`, we combine `DeepFault`'s localization phase with the `FairFLRep` repair strategy. This comparison further isolates the role of fairness-aware fault localization. The repair dataset remains consistent across methods.

Domain Adaptation techniques

`Laftr`

[241] is an adversarial learning framework for fair representation learning. It learns latent representations that obscure sensitive attributes while preserving task-relevant information. The adversarial objective derives theoretical upper bounds on group unfairness, enabling enforcement of demographic parity, equalized odds, or equal opportunity. By minimizing the adversary's ability to predict sensitive attributes, the learned representation aims to produce fair predictions.

`Cfair`

[242] learns fair representations that simultaneously mitigate accuracy parity and equalized odds violations across demographic subgroups. The method aligns conditional distributions of representations rather than marginal distributions and employs balanced error rate (BER) for both the target label and the sensitive attribute. Theoretically, this combination provides guarantees on accuracy parity and equalized odds without enforcing demographic parity. `Cfair` can be viewed as an extension of `Laftr` with two adversaries and BER-based objectives.

`Laftr+Dann`

[123] This variant combines `Laftr` with domain-adversarial training [66] to address distribution shift. The approach aims to control unfairness when the test distribution differs in the marginal distribution of sensitive attributes (e.g., gender or race). It integrates adversarial domain alignment with fairness-aware representation learning.

`Cfair` + Consistency (`Cfair+Consis`)

[71] This method extends `Cfair` by incorporating consistency regularization, inspired by recent work on transferring fairness under distribution shift via fair consistency constraints. The approach leverages self-training principles to encourage stable predictions under input transformations.

`Laftr` + Consistency (`Laftr+Consis`)

[71] This variant augments `Laftr` with consistency regularization, enabling a direct comparison between adversarial fairness learning with and without self-training–style regularization.

8.3.4 Experimental setup

To account for the stochastic nature of the search-based repair process, we repeat each experiment 10 times, reporting average results.

For `DShift-FRep`, `FairFLRep`, `Arachne`, and their variants, we configure the PSO algorithm using the same settings as in the original Arachne study [39]. Each repair instance uses a swarm of 100 particles, where each particle represents a candidate weight modification. The optimization runs for a maximum of 100 generations, with early stopping triggered if the best solution does not improve for 20 consecutive generations. For `MOON` and `DeepFault`-based fault localization, we adopt the `DStar` suspiciousness formula with exponent 3, as recommended in the original `MOON` paper [92]. We select the top 15% of suspicious neurons ($\lambda = 0.15$) for repair. Test input selection follows `MOON` 's Suspicious Neuron Activation (SNA) and Test Input Diversity (TID) objectives.

For `Laftr`, `Cfair`, and DANN-based methods, we used the default parameter configurations in the replication. For instance the adversarial networks are implemented as two-layer multilayer perceptrons (MLPs) with 512 hidden units and ReLU activation. All models are trained for 200 epochs, selecting the best checkpoint based on validation performance. For warm-start fine-tuning and domain adaptation, we use SGD as the optimizer. Moreover, `Cfair` and `Laftr` models have access only to source-domain data, and model selection is performed using the source validation set. Methods that utilize unlabeled target-domain data select models based on performance on a labeled target validation set.

8.4 Evaluation Results

8.4.1 How fairness-aware repair approaches mitigate bias under distribution shifts.

This section evaluates the effectiveness of fairness-aware repair methods under hybrid distribution shifts, comparing static repair techniques (e.g., `FairFLRep`, `Arachne`, `MOON`, `DeepFault`) with our shift-aware `DShift-FRep`. The evaluation covers both tabular datasets (Adult Census dataset) and image datasets (`CelebA` tasks for gender attributes), allowing us to assess performance across different data modalities and distributional challenges.

Evaluation on Tabular Data

Table 8.1 reports fairness (*EOD*) and accuracy (acc) across domain shifts where California (CA) is the source distribution and the states DE, MI, and NY (all 2018 subsets) serve as the target domains. Across all shifts, `DShift-FRep` consistently delivers the strongest fairness improvements, often achieving the lowest or second-lowest *EOD*, while maintaining accuracy close to the unrepaired baseline. In contrast, static and gradient-based baselines show inconsistent fairness behavior and often fail to generalize fairness gains across domains.

On the CA source domain, all methods start with similar accuracy ($\approx 66 - 67\%$). Accuracy shifts after repair are small (± 0.5 points), but fairness outcomes vary dramatically: 1. For DE, MI, NY, and ID as targets, baseline source *EOD* is 0.244, whereas `DShift-FRep` lowers this to $0.05 - 0.09$, depending on the state. 2. `FairFLRep` also reduces source *EOD* but not as strongly (e.g., $0.17 - 0.23$). 3. `DeepFault`, `MOON`, and `Arachne` sometimes increase source unfairness (e.g., $0.26 - 0.25$ for `Arachne`). `DShift-FRep` provides the best overall source fairness performance without meaningful accuracy loss.

A notable pattern in Table 8.1 is that the unrepaired baseline model is surprisingly fairer on most target domains than on the source domain (CA). For DE, MI, and NY, the baseline *EOD* drops substantially when evaluated on the target populations (e.g., CA source $EOD = 0.244$ vs. DE target 0.01, MI target 0.047, NY target 0.11). This indicates that demographic drift from CA to these states does not necessarily make the classifier more unfair; in fact, the baseline model tends to be less biased in the target domain than on the source. As a result, the headroom for fairness improvement on these target states is limited, and fairness gains from repair methods often appear modest.

The exception is ID, where the baseline model is unfair on both the source and target domains (source $EOD = 0.244$, ID target $EOD = 0.255$). In this scenario, `DShift-FRep` achieves

Table 8.1 Unfairness and accuracy comparison between methods on the ForkTable (NewAdult) dataset under state-based distribution shifts. *California (CA) is used as the source domain, and each target domain corresponds to a different state (DE, MI, NY, and ID). For each state pair, acc reports classification accuracy and EOD reports equalized odds difference on the source, target, and mixed-domain evaluations. Results compare the proposed* ***DShift-FRep*** *against baseline (unrepaired),* ***FairFLRep****,* ***DeepFault****, and* ***MOON****, demonstrating fairness improvements and accuracy retention under demographic distribution shifts in census-derived tabular data.*

State (Year)	Method	Source			Target			Mixed		
		Acc	Fairness		Acc	Fairness		Acc	Fairness	
			EOD	Vacc		*EOD*	Vacc		*EOD*	Vacc
DE (2018)	Base	**67.1**	0.244	**0.029**	66.6	**0.01**	0.025	**66.9**	0.19	**0.028**
	FairFLRep	66.4±0.01	0.18±0.09	0.04±0.02	66.2±0.01	0.03±0.03	0.02±0.01	66.3±0.01	0.13±0.07	0.03±0.01
	DeepFault	67.0±0.0	0.22±0.02	0.03±0.0	66.8±0.0	**0.01±0.0**	0.02±0.0	**66.9±0.0**	0.17±0.01	0.03±0.0
	MOON	66.6±0.0	0.2±0.01	0.04±0.0	66.8±0.0	**0.01±0.01**	0.02±0.0	66.7±0.0	0.15±0.01	0.03±0.0
	Arachne	67.0±0.0	0.26±0.01	0.03±0.0	66.5±0.0	0.02±0.01	0.02±0.0	66.8±0.0	0.21±0.01	0.03±0.0
	DShift-FRep	66.5±0.01	**0.08±0.05**	0.05±0.01	66.3±0.0	**0.01±0.02**	0.02±0.01	66.5±0.01	**0.07±0.04**	0.04±0.01
MI (2018)	Base	67.1	0.244	**0.029**	65.5	0.047	0.024	66.3	0.211	0.03
	FairFLRep	**67.4±0.0**	0.23±0.0	0.03±0.0	65.6±0.0	**0.03±0.0**	0.02±0.0	66.5±0.0	0.19±0.0	0.03±0.0
	DeepFault	67.1±0.0	0.24±0.0	0.03±0.0	65.1±0.0	0.05±0.01	0.02±0.0	66.1±0.0	0.21±0.0	0.03±0.0
	MOON	67.1±0.0	0.24±0.0	0.03±0.0	65.2±0.0	0.04±0.01	0.02±0.0	66.2±0.0	0.2±0.0	0.03±0.0
	Arachne	67.2±0.0	0.25±0.01	0.03±0.0	65.3±0.0	0.05±0.0	**0.01±0.01**	66.2±0.0	0.22±0.01	0.03±0.0
	DShift-FRep	67.2±0.0	**0.09±0.0**	0.05±0.0	**66.1±0.0**	0.06±0.0	0.05±0.0	**66.7±0.0**	**0.06±0.0**	0.05±0.0
NY (2018)	Base	67.1	0.244	**0.029**	66.1	0.11	**0.006**	**66.6**	0.193	**0.019**
	FairFLRep	65.8±0.02	0.17±0.05	0.05±0.02	64.9±0.02	0.08±0.04	0.02±0.02	65.3±0.02	0.13±0.05	0.03±0.02
	DeepFault	**67.2±0.0**	0.14±0.05	0.03±0.0	65.8±0.0	0.04±0.03	0.01±0.0	66.5±0.0	0.1±0.04	0.02±0.0
	MOON	66.7±0.0	0.21±0.01	0.04±0.01	66.0±0.0	0.09±0.01	0.01±0.0	66.3±0.0	0.17±0.01	0.02±0.0
	DShift-FRep	65.7±0.0	**0.07±0.0**	0.06±0.0	**66.3±0.0**	**0.03±0.0**	0.02±0.0	66.0±0.0	**0.04±0.0**	0.02±0.0
ID (2018)	Base	**67.1**	0.244	**0.029**	**68**	0.255	0.154	**67.6**	0.254	**0.047**
	FairFLRep	66.9±0.0	0.23±0.0	0.03±0.0	67.6±0.0	0.25±0.0	**0.15±0.0**	67.3±0.0	0.25±0.0	0.05±0.0
	DeepFault	66.9±0.0	0.23±0.01	0.03±0.0	67.7±0.0	0.26±0.02	**0.15±0.0**	67.3±0.0	0.24±0.01	0.05±0.0
	MOON	67.0±0.0	0.23±0.01	0.03±0.0	67.7±0.0	0.25±0.0	**0.15±0.0**	67.4±0.0	0.24±0.01	0.05±0.0
	DShift-FRep	65.9±0.01	**0.05±0.02**	0.06±0.01	66.1±0.0	**0.16±0.03**	0.18±0.02	66.0±0.01	**0.06±0.03**	0.07±0.01

far more pronounced fairness improvements than other methods—reducing *EOD* on both the source (0.05) and target (0.16) domains—demonstrating that its shift-aware repair is especially effective when the underlying model exhibits substantial demographic disparity in both distributions.

It is important to note that `DeepFault`, `MOON`, and `Arachne` —whose fault localization procedures primarily target misclassification rather than fairness—occasionally achieve slightly higher accuracies than `DShift-FRep` on individual states or domains. However, these modest accuracy gains come with a clear tradeoff: they consistently preserve or even amplify unfairness, often yielding substantially higher *EOD* values across both source and target domains. This pattern underscores a key limitation of misclassification-focused repair approaches—improving accuracy alone does not reduce demographic disparities and may, in fact, reinforce them.

> ◉ **Answer to RQ1 (Tabular Data).** *Fairness-aware repair approaches designed for stationary distributions show limited or inconsistent fairness improvements under distribution shift. While some (`DeepFault`, `MOON`, `Arachne`) occasionally achieve slightly higher accuracy than `DShift-FRep`, these gains come at a cost of preserving or exacerbating unfairness. This reflects a fundamental limitation of misclassification-focused repair strategies: reducing prediction errors does not address—and often reinforces—demographic disparities. In contrast, `DShift-FRep` consistently improves fairness across shifts, balances accuracy retention, and provides the most reliable repair behavior under demographic drift.*

Evaluation on Image Data

Table 8.2 extends the comparison to the `CelebA` dataset across three fairness tasks. Overall, the results show that `DShift-FRep` is the only method that consistently improves both accuracy and fairness across the source, target, and mixed domains, demonstrating clear advantages over static repair approaches.

Table 8.2 Performance of `DShift-FRep` against baseline model-repair approaches. *Accuracy (acc) and unfairness (EOD) comparison across source, target, and mixed domains on `CelebA` for three settings: Gender (Young) (gender prediction; age ("young/older") as sensitive attribute), Gender (Eyeglass) (gender prediction; "eyeglasses" as sensitive attribute), and Gender (Cubby) (gender prediction; "cubby" as sensitive attribute).*

Subgroup	Method	Source			Target			Mixed		
		Acc	Fairness		Acc	Fairness		Acc	Fairness	
			EOD	Vacc		*EOD*	Vacc		*EOD*	Vacc
Young	Base	95.9±0	0.227±0	0.012±0	87.2±0	0.139±0	0.04±0	91.5±0	0.178±0	0.063±0
	FairFLRep	90.2±0.01	0.076±0.02	0.045±0.01	90.5±0	0.155±0.01	0.02±0	90.4±0	0.121±0.01	0.01±0
	DeepFault	95.7±0	0.181±0.02	0.013±0	89±0.01	0.101±0.03	0.024±0.01	92.4±0	0.136±0.02	0.049±0
	MOON	95.9±0	0.227±0	0.012±0	87.2±0	0.139±0	0.04±0	91.5±0	0.178±0	0.063±0
	Arachne	95.3±0	0.132±0.01	0.006±0	91±0	0.023±0.01	0.014±0	93.2±0	0.044±0.01	0.029±0
	DShift-FRep	**96±0**	0.159±0.02	0.012±0.01	**92.5±0.01**	**0.021±0.02**	0.014±0.01	**94.2±0**	0.076±0.02	0.028±0.01
Eyeglass	Base	94.3±0	0.128±0	0.134±0	93.6±0	0.108±0	0.112±0	93.9±0	0.104±0	0.114±0
	FairFLRep	**94.5±0**	0.134±0	0.136±0	93.4±0	0.114±0.01	0.113±0.01	93.9±0	0.113±0.01	0.117±0
	DeepFault	**94.5±0**	0.134±0	0.137±0	93.2±0	0.128±0	0.12±0	93.9±0	0.124±0	0.123±0
	MOON	94.3±0	0.123±0.01	0.133±0.01	93.6±0	0.108±0	0.112±0	93.9±0	0.103±0.002	0.114±0
	Arachne	87.1±0.01	0.148±0.04	**0.061±0.02**	91.9±0	0.148±0.03	0.062±0.01	89.5±0.01	0.151±0.03	0.043±0.01
	DShift-FRep	94.3±0	**0.121±0.05**	0.125±0.03	92.9±0.01	**0.088±0.04**	0.095±0.01	93.6±0	**0.094±0.04**	0.102±0.01
Cubby	Base	93.5±0	0.047±0	0.066±0	91.3±0	0.06±0	0.032±0	92.4±0	0.051±0	0.033±0
	FairFLRep	93.5±0	0.061±0	0.066±0	92.7±0	0.014±0	0.021±0	93.1±0	**0.012±0**	0.017±0
	DeepFault	93.4±0	0.058±0	0.066±0	92.6±0	0.029±0.02	0.027±0.01	93±0	0.022±0.01	0.022±0.01
	MOON	93.4±0	0.049±0	0.066±0	91.5±0	0.058±0	0.032±0	92.4±0	0.049±0	0.032±0
	Arachne	84.1±0.01	0.234±0.01	0.129±0.03	90.3±0.01	0.259±0.02	0.073±0	87.2±0.01	0.24±0.01	0.101±0.01
	DShift-FRep	**94.1±0**	**0.046±0.02**	**0.036±0.02**	**92.8±0.01**	0.036±0.02	0.02±0.02	**93.5±0**	0.035±0.02	0.023±0.02

For Gender (Young), `DShift-FRep` achieves the highest source accuracy (96 ± 0) and sig-

nificantly outperforms all baselines in the target domain with 92.5 ± 0.01 accuracy and the lowest target unfairness ($EOD = 0.021 \pm 0.02$). Static repair approaches such as `FairFLRep` and Arachne improve fairness on one domain but fail to generalize the improvements to the target domain. A similar trend holds for Gender (Eyeglass), where `DShift-FRep` again yields the lowest source EOD (0.127 ± 0.05) and lowest target EOD (0.088 ± 0.04) while maintaining high accuracy in both domains—something none of the baselines achieve simultaneously. Finally, in Gender (Cubby), `DShift-FRep` produces the best fairness scores across all domains (e.g., source $EOD = 0.046 \pm 0.02$, target $EOD = 0.036 \pm 0.02$) while also achieving the highest accuracies, whereas static methods often degrade performance or only partially address fairness.

> **Answer to RQ1 (Image Data).** *Static repair methods are insufficient–typically improves fairness in one domain while harming accuracy or failing to transfer fairness gains to another domain. On the other hand, `DShift-FRep` provides stable, consistently superior fairness–accuracy trade-offs, effectively repairing biased neurons in a distribution-aware manner. These results show the necessity of incorporating distribution information, as done in `DShift-FRep`, to maintain fair and reliable performance across both source and target domains.*

8.4.2 RQ2: `DShift-FRep` performance under varying shift magnitude and bias severity

Evaluation on Synthetic Tabular Data

Table 8.3 reports the performance of `DShift-FRep` under progressively increasing shift magnitudes ($\gamma \in \{1, 5, 10, 15, 20\}$) on synthetic tabular data, comparing it to baseline repair methods `FairFLRep`, `DeepFault`, `MOON`, and `Arachne`. Across source, target, and mixed domains, `DShift-FRep` consistently maintains high accuracy while also mitigating fairness violations (*EOD*, *Vacc*) under different data shift. Under small shifts ($\gamma = 1$) , most methods preserve the base model's accuracy; similarly, `DShift-FRep` already produces substantial fairness improvements—showing noticeably lower *EOD* and *Vacc* on both the source and target domains. This result highlights an important property of `DShift-FRep`: even when shifts are mild, its combination of shift-aware alignment and targeted fairness repair reduces decision-boundary disparities without sacrificing predictive utility. As shift magnitude increases ($\gamma = 5, 10, 15, 20$), the base model's performance deteriorates sharply on the target and mixed domains, reflecting its inability to generalize under covariate or concept drift. Similarly, all baselines experience significant degradation as γ grows—marked by steep accuracy

Table 8.3 Performance of `DShift-FRep` across source, target, and mixed domains on the synthetic tabular data with varying shift magnitude (γ). The base model is kept the same while only varying γ in target domain.

γ	Method	Source			Target			Mixed		
		Acc	Fairness		Acc	Fairness		Acc	Fairness	
			EOD	*Vacc*		*EOD*	*Vacc*		*EOD*	*Vacc*
1	Base	84	0.083	0.029	81.3	0.087	0.008	81.3	0.087	0.008
	FairFLRep	83.8±0.0	0.059±0.01	0.032±0.01	79.5±0.02	0.034±0.01	0.022±0.03	82.0±0.01	0.04±0.02	0.027±0.01
	DeepFault	84.0±0.0	0.042±0.0	**0.022±0.0**	81.1±0.0	**0.016±0.0**	**0.008±0.0**	82.8±0.0	**0.03±0.0**	**0.009±0.0**
	MOON	84.1±0.0	0.083±0.0	0.029±0.0	81.3±0.0	0.087±0.0	**0.008±0.0**	82.9±0.0	0.084±0.0	0.013±0.0
	Arachne	84.1±0.0	0.047±0.02	0.028±0.0	81.4±0.0	0.04±0.02	0.013±0.01	83.0±0.0	0.044±0.02	0.021±0.0
	DShift-FRep	**84.3±0.0**	**0.039±0.01**	0.025±0.0	**81.5±0.0**	0.032±0.01	0.012±0.0	**83.2±0.0**	0.035±0.01	0.019±0.0
5	Base	84	0.083	0.029	72.3	0.054	**0.006**	72.3	0.054	0.006
	FairFLRep	**84.5±0.0**	0.058±0.01	0.026±0.0	72.4±0.0	0.071±0.01	0.008±0.0	79.5±0.0	0.03±0.0	0.006±0.0
	DeepFault	84.1±0.0	0.083±0.0	0.029±0.0	72.3±0.0	**0.054±0.0**	**0.006±0.0**	79.2±0.0	0.039±0.0	0.004±0.0
	MOON	84.1±0.0	0.083±0.0	0.029±0.0	72.3±0.0	**0.054±0.0**	**0.006±0.0**	79.2±0.0	0.039±0.0	0.004±0.0
	Arachne	83.8±0.0	0.076±0.06	0.031±0.01	72.9±0.0	0.068±0.02	**0.006±0.0**	79.3±0.0	0.049±0.04	0.003±0.0
	DShift-FRep	83.6±0.01	**0.057±0.01**	**0.024±0.01**	**73.7±0.01**	0.064±0.01	0.012±0.01	**79.6±0.0**	**0.028±0.01**	**0.002±0.0**
10	Base	84	0.083	0.029	66.2	0.048	0.045	66.2	0.048	0.045
	FairFLRep	84.0±0.0	0.061±0.0	0.031±0.0	66.3±0.0	**0.046±0.0**	0.038±0.0	76.7±0.0	**0.001±0.0**	0.055±0.0
	DeepFault	84.1±0.0	0.083±0.0	0.029±0.0	66.2±0.0	0.048±0.0	0.045±0.0	76.7±0.0	0.015±0.0	0.059±0.0
	MOON	84.1±0.0	0.083±0.0	0.029±0.0	66.2±0.0	0.048±0.0	0.045±0.0	76.7±0.0	0.015±0.0	0.059±0.0
	Arachne	83.2±0.0	0.061±0.0	0.038±0.0	68.9±0.0	0.122±0.0	0.028±0.0	77.3±0.0	0.038±0.0	0.023±0.0
	DShift-FRep	**84.2±0.0**	**0.036±0.01**	**0.027±0.0**	**71.3±0.0**	0.057±0.03	**0.02±0.01**	**78.9±0.0**	0.014±0.0	**0.027±0.0**
15	Base	84	0.083	0.029	63.6	**0.061**	**0.001**	63.6	0.061	**0.001**
	FairFLRep	81.3±0.03	0.136±0.01	0.076±0.05	61.6±0.02	0.102±0.02	0.034±0.03	73.2±0.02	0.033±0.0	0.06±0.02
	DeepFault	83.8±0.0	0.111±0.01	0.033±0.0	63.8±0.0	0.077±0.01	0.002±0.0	75.5±0.0	0.018±0.01	0.076±0.0
	MOON	84.1±0.0	0.083±0.0	0.029±0.0	63.6±0.0	**0.061±0.0**	**0.001±0.0**	75.6±0.0	**0.013±0.0**	0.082±0.0
	Arachne	83.7±0.0	0.052±0.01	0.032±0.01	68.8±0.02	0.288±0.04	0.06±0.05	77.5±0.01	0.105±0.02	0.046±0.01
	DShift-FRep	**84.2±0.0**	**0.045±0.02**	**0.023±0.01**	**70.8±0.01**	**0.061±0.05**	0.02±0.02	**78.7±0.0**	0.042±0.02	0.049±0.01
20	Base	84	**0.083**	0.029	58.9	0.205	**0.062**	58.9	0.205	**0.062**
	FairFLRep	83.4±0.0	0.212±0.01	0.036±0.01	57.9±0.0	**0.189±0.01**	0.072±0.0	72.9±0.0	**0.031±0.0**	0.118±0.0
	DeepFault	83.8±0.0	0.105±0.02	0.031±0.0	58.8±0.0	0.191±0.0	0.063±0.0	73.5±0.0	0.038±0.02	0.119±0.0
	MOON	**84.1±0.0**	0.083±0.0	0.029±0.0	58.9±0.0	0.205±0.0	0.062±0.0	73.7±0.0	0.065±0.0	0.121±0.0
	Arachne	83.5±0.0	0.04±0.0	0.031±0.01	68.6±0.01	0.599±0.03	0.074±0.04	77.3±0.01	0.16±0.02	0.059±0.01
	DShift-FRep	83.6±0.01	0.092±0.06	**0.018±0.01**	**68.5±0.04**	0.205±0.12	0.063±0.04	**77.4±0.02**	0.101±0.06	0.076±0.02

drops and increasing fairness disparities, especially in the target and mixed evaluations. In contrast, `DShift-FRep` exhibits stable cross-domain behavior, avoiding performance collapse and frequently achieving the best fairness–accuracy trade-offs across all shift levels.

At the extreme shift scenario ($\gamma = 20$), for example, `DShift-FRep` attains a repaired model accuracy of 68.5 ± 0.04 on the target domain and a markedly improved mixed-domain accuracy of 77.4 ± 0.02 while also producing substantially better fairness metrics. By comparison, the Base model reaches only 58.9 accuracy on the target domain and suffers a severe fairness violation ($EOD = 0.205$). These findings illustrate that, under the harshest distribution shifts, the Base model simultaneously loses accuracy and exhibits large fairness gaps, whereas `DShift-FRep` recovers a sizeable portion of utility and significantly reduces fairness disparities.

◉ Answer to RQ2 (Synthetic Tabular Data). *Across progressively larger shifts,* ***DShift-FRep*** *remains the most stable and reliable method, preserving accuracy while consistently reducing fairness violations on both source and target domains. Unlike the Base model and existing repair baselines—which show sharp accuracy drops and worsening EOD as shift magnitude increases—****DShift-FRep*** *maintains strong generalization and fairness even under extreme drift.*

Evaluation on Synthetic Image Data (3D-shapes)

Table 8.4 Performance of DShift-FRep compared with baseline model-repair approaches across source, target, and mixed domains on the 3D-Shapes dataset. The table reports accuracy and fairness (*EOD*, Vacc) for each shift type, showing how different repair methods affect predictive performance and subgroup fairness.

Subgroup	Method	Source			Target			Mixed		
		Acc	Fairness		Acc	Fairness		Acc	Fairness	
			EOD	*Vacc*		*EOD*	*Vacc*		*EOD*	*Vacc*
Sshift1	Base (Keras)	93.4	**0.078**	0.062	90.7	0.001	0.07	91.8	0.035	0
	FairFLRep	93.5±0.01	**0.078**±	0.058±0.01	90.9±0.01	0.024±0.02	0.071±0.01	91.9±0.01	**0.024±0.01**	0.066±0.01
	DeepFault	93.6±0.01	**0.078**±	0.057±0.01	90.7±0.01	0.019±0.0	0.072±0.01	91.9±0.01	**0.024±0.0**	0.067±0.01
	MOON	93.4±0.0	**0.078**±	0.061±0.0	90.7±0.0	**0.007±0.0**	0.069±0.0	91.8±0.0	0.039±0.0	0.066±0.0
	Arachne	93.2±0.01	0.21±	0.013±0.01	90.8±0.0	0.203±0.0	0.028±0.01	91.7±0.0	0.206±0.01	0.021±0.01
	DShift-FRep	**97.8±0.0**	0.091±0.008	**0.009±0.0**	**97.2±0.0**	0.107±0.02	**0.014±0.0**	**97.4±0.0**	0.1±0.01	**0.012±0.0**
Sshift2	Base (Keras)	80.6	0.262	0.216	82.4	0.314	0.19	81.7	0.307	0
	FairFLRep	75.1±0.03	0.173±	0.099±0.01	73.6±0.02	0.126±0.03	0.072±0.05	74.2±0.01	0.134±0.03	0.043±0.05
	DeepFault	70.9±0.06	0.183±	0.139±0.1	73.1±0.01	0.123±0.01	0.049±0.03	72.2±0.03	0.132±0.01	0.058±0.04
	MOON	70.0±0.09	0.177±	0.188±0.15	73.5±0.02	0.111±0.0	0.037±0.03	72.1±0.05	0.121±0.0	0.087±0.06
	Arachne	80.9±0.01	0.352±	0.209±0.1	78.9±0.07	0.328±0.2	0.178±0.18	79.7±0.04	0.331±0.2	0.117±0.1
	DShift-FRep	**96.9±0.01**	**0.01±0.006**	**0.034±0.01**	**95.1±0.0**	**0.023±0.01**	**0.021±0.01**	**95.8±0.0**	**0.02±0.01**	**0.024±0.01**
Dshift	Base (Keras)	98.8	0.032	0.001	61.1	0.468	0.215	78.2	0.275	0
	FairFLRep	86.1±0.01	0.216±	0.182±0.02	51.0±0.01	**0.069±0.03**	**0.018±0.01**	66.9±0.01	0.133±0.01	0.08±0.01
	DeepFault	98.7±0.0	0.026±	**0.003±0.0**	60.5±0.0	0.448±0.0	0.195±0.0	77.8±0.0	0.262±0.0	0.098±0.0
	MOON	88.9±0.03	0.184±	0.141±0.05	52.6±0.02	0.07±0.04	0.053±0.03	69.1±0.02	0.093±0.04	0.05±0.02
	Arachne	74.9±0.02	0.181±	0.123±0.02	72.9±0.01	0.123±0.05	0.019±0.01	73.8±0.01	0.149±0.05	0.067±0.01
	DShift-FRep	**99.4±0.0**	**0.024±0.015**	0.005±0.0	**93.2±0.01**	0.099±0.05	0.03±0.0	**96.0±0.0**	**0.056±0.03**	**0.02±0.0**
Hshift	Base (Keras)	99.5	0.016	0.004	60.8	0.616	0.368	78.4	0.502	0
	FairFLRep	79.5±0.0	0.256±	0.288±0.0	48.3±0.0	0.027±0.01	0.026±0.01	62.5±0.0	0.075±0.01	0.105±0.0
	DeepFault	**99.7±0.0**	**0.0**±	**0.002±0.0**	60.3±0.0	0.594±0.0	0.353±0.0	78.2±0.0	0.481±0.0	0.177±0.0
	MOON	82.0±0.0	0.292±	0.231±0.01	47.1±0.0	**0.026±0.01**	0.032±0.0	62.9±0.0	0.081±0.0	0.075±0.01
	Arachne	73.1±0.01	0.446±	0.26±0.04	63.9±0.02	0.293±0.06	0.15±0.04	68.1±0.01	0.322±0.05	0.033±0.02
	DShift-FRep	97.9±0.02	0.024±0.041	0.016±0.01	**90.1±0.01**	0.049±0.02	**0.009±0.01**	**93.6±0.02**	**0.037±0.02**	**0.011±0.01**

Table 8.4 summarizes the full repair-based evaluation and reveals several consistent patterns across all shift scenarios (Sshift1, Sshift2, Dshift, and Hshift). Across these settings, DShift-FRep achieves the most reliable fairness gains in both the source and target domains, often reducing *EOD* by $2^{\smile}10^{\times}$ relative to the uncorrected base model and outperforming existing repair baselines including FairFLRep, DeepFault, MOON, and Arachne. Crucially, DShift-FRep not only avoids accuracy degradation but often improves accuracy substantially,

even under strong distribution shifts. For example, accuracy increases from 93.4% → 97.8% (source) and 90.0% → 97.2% (target) under `Sshift1`, 80.6% → 96.9% (source) and 82.4% → 95.1% (target) under `Sshift2`, 98.8% → 99.4% (source) and 61.1% → 93.2% (target) under `Dshift`, and 99.5% → 97.9% (source) and 60.8% → 90.1% (target) under `Hshift`. Standard fairness-aware repair approaches typically maintain accuracy close to the base model while targeting fairness improvements, but rarely enhance both simultaneously across domains. By contrast, the joint gains in fairness and accuracy achieved by `DShift-FRep` highlight that its shift-aware warm-start alignment and targeted neuron repair reinforce predictive performance while improving fairness. This leads to substantially greater stability compared to competing repair or adaptation baselines, which frequently suffer from over-correction, fairness degradation, or accuracy collapse—especially under severe or heterogeneous shifts.

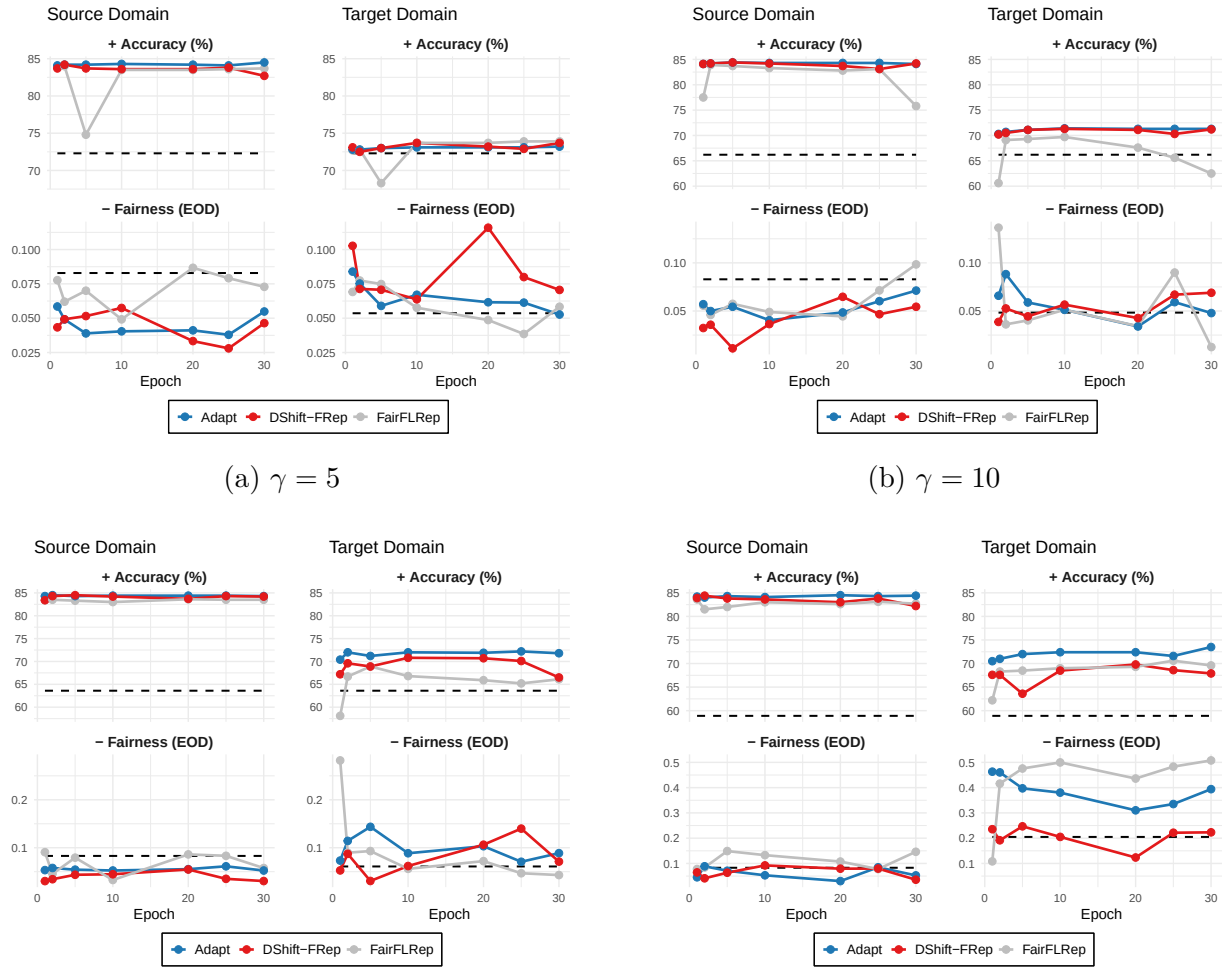
> ◉ **Answer to RQ2 (Synthetic Image Data).** *On the `3D-shapes` benchmarks, `DShift-FRep` again shows clear advantages by simultaneously boosting accuracy and improving fairness across all shift settings. Whereas standard repair baselines typically improve fairness at the cost of utility—or fail under severe shifts—`DShift-FRep` achieves large accuracy gains (often $+10-30\%$) alongside major EOD reductions in both domains. This demonstrates that its combination of shift-aware alignment and targeted neuron repair is highly effective in complex visual settings, yielding strong joint improvements in fairness and predictive performance.*

8.4.3 RQ3: How label propagation and alignment improve fairness across both domains

Evaluation on Tabular Data

Figure 8.4 illustrates how accuracy (top row) and fairness (*EOD*, bottom row) evolve over 30 adaptation epochs for increasing shift magnitudes ($\gamma = 5, 10, 15, 20$) on the synthetic tabular dataset. The dashed black line represents the baseline model before any adaptation or repair. Across all γ values, both `DShift-FRep` and `FairFLRep` show substantial accuracy improvements even after a single warm-start epoch. Accuracy remains nearly constant with additional fine-tuning, indicating that peak performance is reached early, while consistently staying well above the baseline on both source and target domains. Fairness, measured via *EOD*, exhibits more nuanced behavior. For the smallest shift ($\gamma = 5$), *EOD* fluctuates during early epochs, whereas for larger shifts, *EOD* steadily decreases with minimal oscillations. In contrast, pure Adapt (blue) can improve target-domain accuracy early in training but often

exhibits unstable fairness, particularly at $\gamma = 10$ and $\gamma = 20$, highlighting its sensitivity to shift severity.



(a) $\gamma = 5$

(b) $\gamma = 10$

(c) $\gamma = 15$

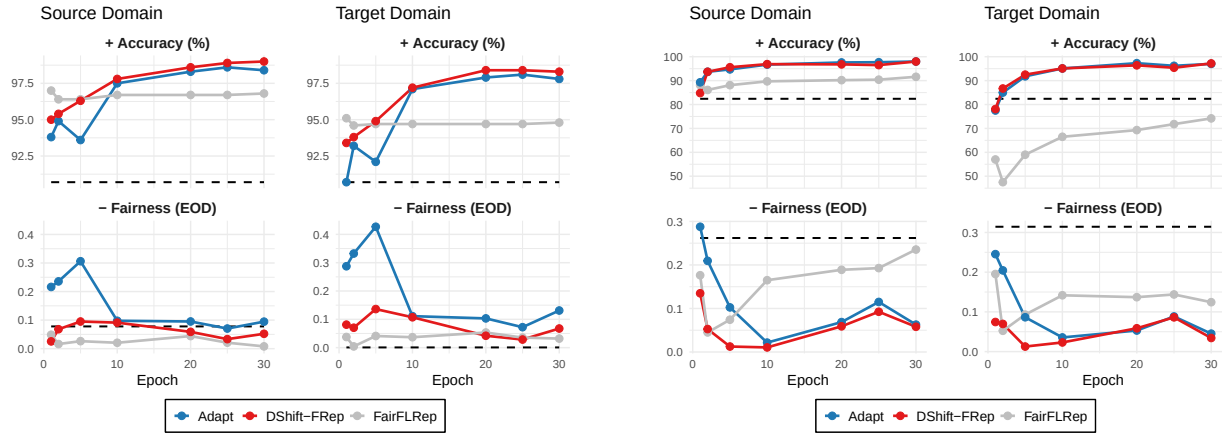
(d) $\gamma = 20$

Figure 8.4 Evolution of accuracy and fairness across domains during adaptation for synthetic Tabular Data.

`FairFLRep` (gray) achieves competitive improvements in both accuracy and fairness for smaller shifts but becomes less stable under large shifts $\gamma \geq 15$. `DShift-FRep`, however, maintains the most reliable fairness–accuracy trajectory as shifts become more severe. At $\gamma = 20$, it avoids the sharp target-domain *EOD* spikes observed in Adapt and `FairFLRep` while maintaining stable accuracy across epochs. Overall, the temporal dynamics demonstrate that `DShift-FRep` not only converges to superior final outcomes but also follows a smoother, more stable adaptation path, making it particularly dependable under progressive distribution drift.

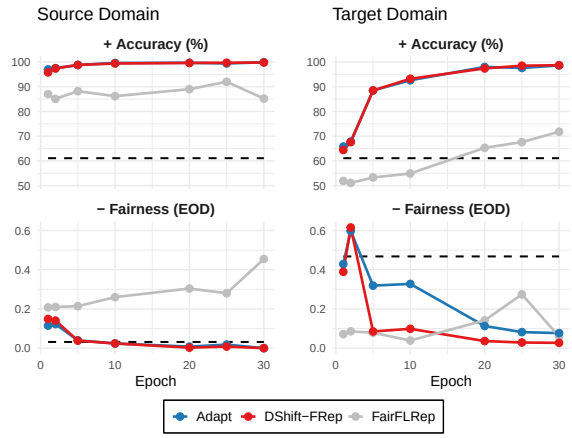
Evaluation on Image Data

The Figure 8.5 presents the evolution of model accuracy and fairness (*EOD*) during adaptation across four different types of distribution shifts in the `3D-shapes` dataset. Each subplot corresponds to one shift setting: (8.5a) `Sshift1`, (8.5b) `Sshift2`, (8.5c) `Dshift`, and (8.5d) `Hshift`. Within each setting, results are shown side-by-side for the source and target domains, with accuracy curves on the top row and fairness curves on the bottom row. The plots compare performance of multiple baselines (Base model, `Arachne`, `DeepFault`, `FairFLRep`, `MOON`) against the two phases of our method: Adapt and Repair.

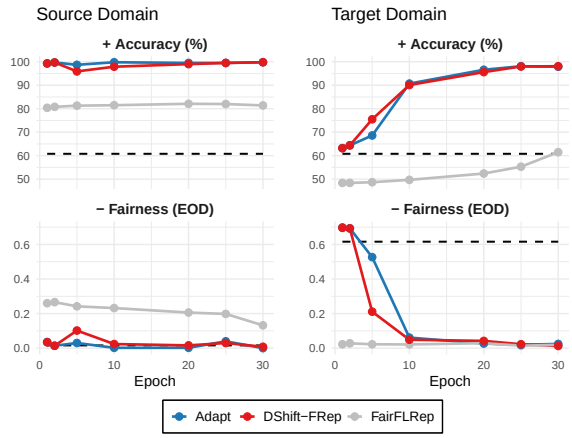


(a) `Sshift1`

(b) `Sshift2`



(c) `Dshift`



(d) `Hshift`

Figure 8.5 Evolution of accuracy and fairness across domains during adaptation for `3D-shapes`.

Across all shift scenarios, the `DShift-FRep` Adapt phase quickly boosts target-domain ac-

curacy, often matching or exceeding baselines within only a few epochs. However, this gain alone sometimes comes at the expense of fairness stability-especially under large domain gap conditions such as `Dshift` and `Hshift`. The Repair phase consistently corrects this, delivering substantial reductions in *EOD* while preserving or slightly improving accuracy. This stability is visible in the way Repair curves flatten toward lower unfairness values without drifting upward—an issue that affects several baselines. Notably, the proposed Adapt then Repair achieves these improvements without costly full-model retraining or adversarial domain-adaptation cycles. Instead, it performs lightweight alignment followed by targeted neuron-level repair, making it significantly more efficient than deep adversarial training while still maintaining strong performance under severe shifts.

Figure **??** presents the evolution of accuracy and fairness (*EOD*) on the `CelebA` dataset across source and target domains. Similar to the results observed on the `3D-shapes` dataset, both Adapt and Repair phases exhibit stable and predictable behavior over training epochs. Accuracy remains consistently high in the source domain and improves early in the target domain without suffering from the instability typically introduced by adversarial fine-tuning.

> ◉ **Answer to RQ3.** *The proposed two-stage strategy in `DShift-FRep` provides robust cross-domain fairness correction with minimal computational cost, while avoiding the instability and preventing fairness collapse even under severe distribution shifts.*

8.4.4 RQ4: Performance of `DShift-FRep`, compared to domain adaptation approaches

Table 8.5 reports results for DA approaches evaluated under identical shift settings on the `3D-shapes` dataset. Although these experiments were implemented in PyTorch (unlike the repair-based experiments conducted in Keras), both base models were trained under identical conditions with equivalent architectures and hyperparameters.

Across all shifts, DA approaches (e.g., LAFTR, CFair, consistency-regularized variants) generally improve upon the base model but tend to prioritize balanced source–target performance. This balance often comes at the cost of substantial degradation in domains where the base model already performs well. For example, For CFair+consistency, accuracies fall from 99.9 $\rightarrow$ 97.9 (source) and 99.1 $\rightarrow$ 97.0 (target) under `Sshift1`, and from 99.7 $\rightarrow$ 97.7/96.8 (source) and 98.5 $\rightarrow$ 97.5/96.0 (target) under `Sshift2` for LAFTR+consistency and CFair+consistency. These declines reflect the over-adjustment effect: when fairness and alignment are already well satisfied, additional adaptation disrupts an otherwise calibrated model.

Table 8.5 Performance of DA baselines experiments across source, target, and mixed domains on the `3D-shapes` dataset. The table reports accuracy and fairness (*EOD*, *Vacc*) for each shift type, showing how different Domain Adaptation approaches affect predictive performance and subgroup fairness.

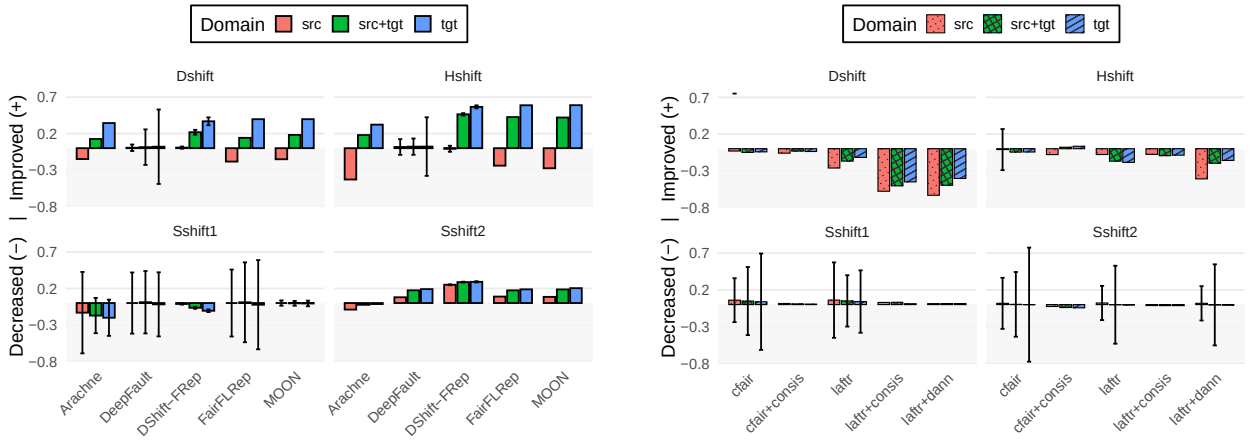
Shift	Method	Source			Target			Mixed		
		Acc	Fairness		Acc	Fairness		Acc	Fairness	
			EOD	Vacc		*EOD*	Vacc		*EOD*	Vacc
Sshift1	Base (Torch)	99.9±0.0	0.087±0.03	0.991±0.01	99.1±0.0	0.056±0.02	0.991±0.01	99.1±0.0	0.069±0.02	0.991±0.01
	`Laftr`	99.1±0.0	0.026±0.02	0.991±0.01	99.0±0.0	0.015±0.02	0.99±0.01	99.0±0.0	0.019±0.01	0.99±0.01
	`Cfair`	99.0±0.0	0.026±0.02	0.99±0.01	98.9±0.01	0.016±0.0	0.989±0.01	98.9±0.0	0.02±0.01	0.989±0.01
	`Laftr+Dann`	98.6±0.02	0.074±0.07	0.986±0.02	98.1±0.01	0.042±0.03	0.981±0.02	98.3±0.01	0.055±0.05	0.983±0.02
	`Laftr+Consis`	97.0±0.02	0.058±0.05	0.971±0.02	96.1±0.01	0.044±0.02	0.961±0.02	96.5±0.01	0.038±0.04	0.965±0.01
	`Cfair+Consis`	97.9±0.01	0.072±0.06	0.979±0.01	97.0±0.01	0.049±0.04	0.97±0.01	97.3±0.01	0.059±0.05	0.973±0.01
Sshift2	Base (Torch)	99.7±0.0	0.022±0.02	0.997±0.004	98.5±0.01	0.004±0.0	0.985±0.01	99.0±0.01	0.007±0.0	0.99±0.01
	`Laftr`	99.8±0.0	0.0±0.0	0.998±0.0	99.3±0.0	0.01±0.01	0.993±0.01	99.5±0.0	0.008±0.01	0.995±0.0
	`Cfair`	99.6±0.01	0.004±0.01	0.996±0.01	99.4±0.0	0.005±0.0	0.994±0.0	99.5±0.0	0.003±0.0	0.995±0.0
	`Laftr+Dann`	99.8±0.0	0.004±0.01	0.998±0.0	99.3±0.0	0.012±0.01	0.993±0.01	99.5±0.0	0.011±0.01	0.995±0.0
	`Laftr+Consis`	97.7±0.0	0.035±0.03	0.976±0.01	97.5±0.01	0.018±0.01	0.975±0.01	97.6±0.01	0.021±0.01	0.976±0.01
	`Cfair+Consis`	96.8±0.01	0.048±0.04	0.968±0.02	96.0±0.02	0.048±0.04	0.96±0.03	96.3±0.01	0.046±0.04	0.963±0.02
Dshift	Base (Torch)	98.8±0.0	0.018±0.02	0.988±0.005	66.4±0.03	0.357±0.05	0.663±0.12	81.1±0.02	0.234±0.03	0.813±0.06
	`Laftr`	80.6±0.2	0.282±0.33	0.811±0.24	60.3±0.06	0.475±0.31	0.601±0.18	69.5±0.12	0.403±0.3	0.698±0.19
	`Cfair`	98.4±0.01	0.05±0.03	0.984±0.01	65.4±0.01	0.401±0.05	0.652±0.15	80.4±0.01	0.285±0.03	0.806±0.08
	`Laftr+Dann`	63.3±0.17	0.653±0.4	0.648±0.29	53.8±0.05	0.762±0.29	0.535±0.23	58.1±0.1	0.732±0.32	0.587±0.25
	`Laftr+Consis`	67.7±0.23	0.597±0.47	0.69±0.32	55.0±0.06	0.809±0.22	0.547±0.25	60.7±0.14	0.739±0.31	0.613±0.27
	`Cfair+Consis`	94.2±0.02	0.079±0.05	0.943±0.03	65.3±0.02	0.395±0.18	0.651±0.13	78.5±0.01	0.267±0.13	0.786±0.07
Hshift	Base (Torch)	98.5±0.0	0.04±0.03	0.985±0.005	61.8±0.01	0.363±0.04	0.621±0.1	78.5±0.01	0.314±0.03	0.784±0.06
	`Laftr`	91.5±0.11	0.119±0.17	0.918±0.14	63.5±0.04	0.549±0.16	0.64±0.17	76.2±0.07	0.484±0.16	0.761±0.11
	`Cfair`	98.7±0.0	0.052±0.02	0.987±0.0	66.0±0.02	0.408±0.07	0.664±0.11	80.9±0.01	0.361±0.06	0.808±0.07
	`Laftr+Dann`	73.0±0.21	0.451±0.43	0.741±0.28	55.3±0.04	0.523±0.44	0.558±0.21	63.4±0.11	0.512±0.44	0.633±0.12
	`Laftr+Consis`	89.1±0.05	0.117±0.09	0.893±0.07	62.5±0.03	0.451±0.12	0.63±0.13	74.6±0.03	0.409±0.1	0.745±0.08
	`Cfair+Consis`	95.2±0.02	0.12±0.08	0.952±0.03	65.9±0.03	0.33±0.2	0.662±0.11	79.2±0.02	0.294±0.17	0.792±0.06

Next, we visually analyses the relative fairness and accuracy changes, contrasting the performance of `DShift-FRep` and the baselines repair and DA approaches. Figure 8.6 (i.e., Figure 8.6a for repair-based approaches and Figure 8.6b for DA experiments) shows the relative fairness change in *EOD* achieved by each method on `3D-shapes` under four types of distribution shifts. Each subplot corresponds to one shift type, and within each subplot the results are further separated by domain (source, target, and combined). Positive values indicate fairness improvement, while negative values indicate fairness degradation relative to the baseline model. The shaded region below zero highlights harmful fairness drops.

Across most shift conditions (`Dshift`, `Hshift`, `Sshift2`), `DShift-FRep` consistently provides more stable fairness and accuracy than standard domain adaptation approaches, especially in the target domain where fairness forgetting and accuracy collapse are most prominent. By combining mild adaptation with targeted neuron-level repair, `DShift-FRep` avoids the over-corrections commonly observed in adversarial and contrastive adaptation methods, leading to smoother fairness gains and fewer volatile trade-offs. This trend is further confirmed across

Table 8.6 Performance of DA baselines on `CelebA` dataset, reporting the accuracy and fairness (*EOD*, *Vacc*).

Shift	Method	Source			Target			Mixed		
		Acc	Fairness		Acc	Fairness		Acc	Fairness	
			EOD	Vacc		*EOD*	Vacc		*EOD*	Vacc
Young	Base	77.0±0.01	0.329±0.03	0.257±0.026	54.9±0.02	0.489±0.04	0.175±0.02	65.9±0.02	0.484±0.03	0.273±0.02
	laftr	76.1±0.01	0.268±0.04	0.237±0.04	57.7±0.04	0.391±0.08	0.152±0.03	66.9±0.02	0.39±0.08	0.235±0.04
	cfair	59.1±0.2	0.531±0.3	0.333±0.13	59.0±0.06	0.662±0.24	0.22±0.04	59.0±0.08	0.657±0.23	0.332±0.07
	laftr+dann	75.8±0.01	0.365±0.14	0.299±0.1	53.1±0.04	0.495±0.13	0.183±0.05	64.5±0.02	0.491±0.12	0.286±0.06
	laftr+consis	75.7±0.01	0.288±0.12	0.244±0.08	56.7±0.04	0.409±0.13	0.153±0.04	66.2±0.02	0.405±0.13	0.238±0.06
	cfair+consis	75.6±0.01	0.324±0.04	0.264±0.03	55.3±0.03	0.437±0.08	0.166±0.03	65.4±0.02	0.44±0.06	0.257±0.04



(a) Relative fairness (Repair techniques)

(b) Relative fairness (Domain Adaptation)

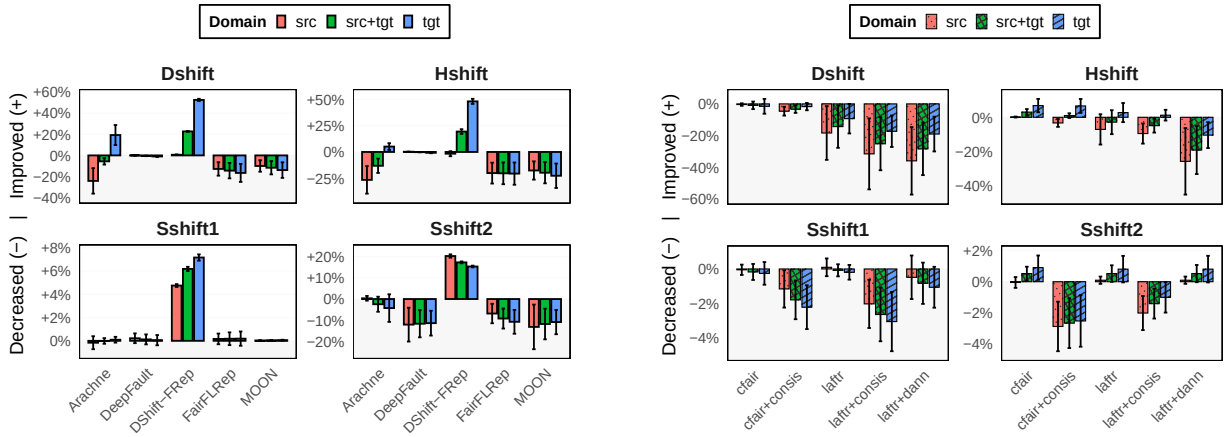
Figure 8.6 Comparison of relative improvement fairness by `DShift-FRep` across methods and domains. Left: Keras repair; Right: Pytorch baselines.

other baselines. Many repair methods (e.g., `Arachne` and `DeepFault`) improve fairness for one domain–typically the source–but simultaneously degrade fairness on the target domain, indicating domain-specific overfitting. `DShift-FRep` instead achieves the most stable and balanced fairness improvements, reducing cross-domain interference. Unlike existing repair approaches, which often help one domain at the expense of another, `DShift-FRep` delivers improvements consistently across source, target and combined, validating its shift-adaptive fairness mechanism.

Considerable differences also appear among fairness baselines. `FairFLRep`, while effective in some conditions, exhibits higher variance and occasionally large negative drops (especially under `Sshift1`), reflecting the sensitivity of layer-wise fault localization to distribution shift. `MOON`, included as a contrastive fairness-aware reference, produces mixed results: it improves fairness on average but fails to generalize reliably across the four shifts, showing that con-

trastive fairness optimization alone does not guarantee robustness under shift. Overall, the fairness results show that most repair baselines fail to generalize their fairness improvements across domains, whereas `DShift-FRep` systematically reduces *EOD* disparities in both source and target distributions.

The corresponding accuracy-change results in Figure 8.7 (i.e., Figure 8.7a for repair-based experiments and Figure 8.7b for DA experiments) follow the same layout but report relative changes in classification accuracy rather than fairness. Values above zero indicate accuracy improvement; values below zero indicate degradation.



(a) Relative Accuracy (Repair techniques)

(b) Relative Accuracy (Domain Adaptation)

Figure 8.7 Comparison of the relative accuracy change of the resulting models when different repair and DA approaches are applied for different shift types in `3D-shapes`. Left is the accuracy achieved by repair experiments, while the right plot are the performance by DA baselines experiments.

Fine-tuning and domain adaptation baselines (e.g., `Cfair`, `Laftr`, `Laftr+Consis`, `Laftr+Dann`, `Cfair+Consis`) exhibit highly heterogeneous behavior depending on the shift. Under `Dshift`, most methods slightly improve accuracy in the combined domain. However, under `Hshift` and `Sshift2`, several baselines show negative shifts—indicating that adaptation often over-specializes to the source distribution, harming target-domain accuracy.

Similar trade-offs occur in fairness-oriented repair approaches. Several methods reduce accuracy on the target domain, demonstrating that fairness repair without shift-awareness often comes at the cost of predictive performance under unseen distributions. In contrast, `DShift-FRep` maintains accuracy stability while improving fairness. Across all four shift scenarios, `DShift-FRep` avoids the large negative accuracy swings observed in many baselines. It preserves accuracy in both source and target, demonstrating that its shift-aware align-

ment and warm-start repair steps avoid excessively perturbing task-relevant features. The variance bars further reveal that models trained without shift awareness are unstable; the high variability under `Sshift1`, in particular, shows that both fine-tuning and fairness repair often overfit to stochastic effects of the shifted data.

> **Answer to RQ4.** *Overall, `DShift-FRep` achieves superior cross-domain stability, improving both fairness and accuracy when the model requires correction, and behaving conservatively when the model is already well aligned. This reduces the risk of fairness forgetting, instability, and performance collapse across domains. Standard fairness-aware repair approaches typically maintain accuracy close to the base model while targeting fairness improvements, but they rarely enhance both fairness and accuracy simultaneously across domains. In contrast, DA approaches generally improve upon the base model but often prioritize balanced source–target performance, leading to substantial degradation in domains where the base model already performs well.*

8.5 Discussion

Our results show that `DShift-FRep` generally achieves superior cross-domain stability, improving both fairness and accuracy when the model requires correction, and behaving conservatively when the model is already well aligned. This reduces the risk of fairness forgetting, instability, and performance collapse across domains. Standard fairness-aware repair approaches typically maintain accuracy close to the base model while targeting fairness improvements, but they rarely enhance both fairness and accuracy simultaneously across domains. In contrast, domain adaptation (DA) approaches generally improve upon the base model but often prioritize balanced source–target performance, leading to substantial degradation in domains where the base model already performs well. This is further compounded by the fact that DA methods involve deep retraining, which can produce uncertain or unstable parameter updates, making the fine-tuned model less reliable than the original base model—especially under mild shifts where little adaptation is needed. In comparison, `DShift-FRep` encourages only minimal, targeted fine-tuning, using warm-start alignment and neuron-level repair to avoid unnecessary parameter drift. As a result, it provides more stable and effective shift-aware corrections than both standard repair and DA baselines, yielding consistent fairness gains without sacrificing predictive performance as the data distribution shifts.

8.6 Summary

This paper introduced `DShift-FRep`, a modular framework for fairness-aware repair of deep neural networks under hybrid distribution shifts. Unlike conventional fairness-aware learning or domain adaptation methods that assume static distributions or require extensive retraining, `DShift-FRep` explicitly combines adaptation and targeted repair, enabling stable corrections while minimizing unnecessary parameter drift.

For classical DNNs, our results show that `DShift-FRep` achieves superior cross-domain stability, improving both fairness and accuracy when correction is needed, while behaving conservatively when the base model is already well aligned. By integrating fairness-aware label propagation, shift-aware data alignment, warm-start fine-tuning, and neuron-level repair, the framework avoids the instability and fairness forgetting commonly observed in retraining-based domain adaptation approaches. This makes `DShift-FRep` particularly well suited for real-world deployments where data distributions evolve gradually and reliability must be preserved.

For transformer-based language models, our analysis provides complementary insights into how bias is encoded and controlled. We show that although a small set of Pareto-ranked neurons is strongly associated with biased behavior, bias is not localized to a single component. Instead, it is redundantly and distributively encoded across interacting neurons, making direct neuron suppression alone ineffective. In contrast, token-level interventions are highly disruptive, revealing bias triggers but severely degrading performance. These findings position neuron-level localization not as a deletion target, but as a stable search space for optimization-based and interaction-aware repair.

On the one hand, our results suggest that effective fairness repair requires coordinated, optimization-driven adjustments rather than hard suppression in large language models. On the other hand, unifying adaptation, localization, and targeted repair, `DShift-FRep` offers a practical and interpretable pathway for maintaining fairness and reliability in neural systems operating under evolving and uncertain conditions.

CHAPTER 9 CONCLUSION

In this chapter, we summarize our main findings, conclude the thesis, and discuss directions for future work.

9.1 Summary

This thesis investigates how unfair behavior emerges, persists, and can be mitigated in modern ML systems, spanning classical ML models, DNNs, and transformer-based LLMs. Moving beyond the prevailing view that bias is not solely a data issue, this thesis demonstrates that unfair behavior also arises from architectural and representational properties embedded within model parameters and internal representations. Consequently, bias can be systematically detected, analyzed, and repaired through targeted interventions across different phases of the machine learning lifecycle.

The thesis is structured around five (5) interconnected contributions. First, it introduces a systematic framework for bias detection and evaluation in classical machine learning models. The thesis develops principled methods to identify bias-inducing features using causal reasoning and feature-swapping analyses, without requiring prior knowledge of sensitive attributes. Extensive empirical studies show that certain features—while weakly relevant for predictive accuracy—can disproportionately drive unfair outcomes. Correcting or removing such bias-inducing yet low-utility features leads to substantial fairness improvements while preserving predictive performance.

Second, the thesis advances fairness repair for trained deep neural networks through `FairFLRep`, a fairness-aware fault localization and repair framework. `FairFLRep` treats unfair behavior as a localized defect in learned representations rather than a global model failure. It identifies neuron weights that disproportionately contribute to subgroup disparities and applies targeted, minimal parameter updates inspired by program repair techniques in traditional software engineering. By avoiding full retraining, `FairFLRep` provides a lightweight and interpretable mechanism for post-training fairness repair when retraining is infeasible or costly.

Third, building on these foundations, the thesis investigates fairness in pretrained transformer-based LLMs through two tightly coupled studies on bias localization and repair. An in-depth analysis using a dedicated dataset of biased relational triplets and neuron attribution techniques shows that biased knowledge in transformers is not uniformly distributed, but concentrated in relatively small subsets of neurons, while also being redundantly encoded across the

network. These findings establish bias as an architectural and representational phenomenon rather than a purely data-driven artifact and explain why naïve suppression strategies often yield unstable or ineffective mitigation.

Leveraging these insights, the thesis proposes a principled suppress-and-repair framework for pretrained transformers. Instead of relying on hard suppression alone, the approach introduces Pareto-optimal, contrastive neuron localization to identify compact, high-confidence sets of fairness-critical neurons that balance bias contribution and task relevance. Targeted repair is then performed by directly modifying a small set of outgoing weights using PSO, guided by correctness- and fairness-aware objectives. Experimental results demonstrate that global retraining is not always necessary for effective bias mitigation; precise neuron-level repair offers a lightweight, interpretable, and practical alternative. The results further show that suppression-only interventions—such as neuron zeroing or token masking—are insufficient for production systems, as they introduce information loss and uncertainty without explicitly optimizing fairness.

Finally, the thesis addresses fairness repair under distribution shifts, a pervasive challenge in real-world deployments where data distributions evolve and the nuisance factors driving bias shifts are often unknown or unmeasurable. To overcome the limitations of static repair approaches such as `FairFLRep`, the thesis proposes `DShift-FRep`, an adaptive framework that follows an Adapt-then-Repair paradigm. `DShift-FRep` integrates data-level alignment, lightweight warm-start fine-tuning, and targeted neuron repair via PSO to jointly improve predictive performance and fairness across source and target domains. Empirical results on both image and tabular datasets show that `DShift-FRep` consistently outperforms state-of-the-art domain adaptation, self-training, and static fairness repair techniques, while avoiding unnecessary parameter drift.

9.2 Key Findings and Implications

- *Bias is an internal representational phenomenon*, – Across classical models, DNNs, and transformer-based LLMs, bias is not solely a property of data or labels but is embedded within internal representations and parameters. These bias-inducing components—features, neurons, and weights—can be systematically localized and analyzed, enabling targeted and interpretable mitigation strategies.

- *Biased knowledge in transformers is compact yet redundantly encoded.* – n pretrained transformer models, biased knowledge is often concentrated in a small subset of neurons that disproportionately contribute to stereotypical behavior. At the same time, this in-

formation is redundantly encoded across the network, explaining why naïve suppression or random pruning yields limited, unstable, or short-lived fairness improvements.

- *Suppression alone is insufficient for production systems* – Hard suppression strategies—such as neuron zeroing or token masking—introduce information loss and uncertainty, frequently degrading task performance without explicitly optimizing fairness. While suppression can confirm causal responsibility, it is inadequate as a standalone mitigation strategy for deployed systems.

- *Targeted neuron-level repair is effective and efficient* – Pareto-optimal, contrastive localization combined with PSO-based repair enables precise, interpretable interventions that significantly improve fairness while preserving—or even improving—predictive accuracy. These results demonstrate that global retraining is not always necessary; small, carefully selected parameter updates can achieve substantial fairness gains.

- *Fairness is multi-dimensional and metric-dependent* – Optimizing a single fairness metric is often insufficient and can negatively impact others, especially in complex models. Fairness metric behavior varies by data modality and model complexity: simpler, group-level metrics (e.g., *DI*, *SPD*) generalize better in tabular settings, while metrics such as *EOD* are more effective in high-dimensional image domains. These findings underscore the need for multi-metric, context-aware fairness evaluation and repair.

- *Static fairness repair does not generalize under distribution shift* – Fairness improvements achieved on a source distribution do not reliably transfer to shifted target distributions. This finding highlights a fundamental limitation of static repair methods and motivates adaptive, shift-aware repair strategies.

- *Adapt-then-Repair improves robustness and fairness* – `DShift-FRep` shows that combining lightweight adaptation with targeted neuron repair yields superior fairness–performance trade-offs under realistic distribution shifts, outperforming domain adaptation, self-training, and static repair techniques while avoiding unnecessary parameter drift.

9.2.1 Implications

- **For ML practitioners:** Fairness mitigation does not always require expensive retraining or full model redesign. Targeted neuron- and weight-level repair provides a lightweight, deployable alternative that preserves model utility and reduces computational cost—especially valuable in production environments.

Also, the results caution against relying on a single fairness metric. Practitioners should adopt multi-metric evaluation strategies tailored to the application domain and data modality to avoid hidden trade-offs and unintended harms.

- **For high-stakes domains (e.g., healthcare, hiring, education)** Interpretable, parameter-level interventions improve auditability and accountability, which are critical for applications in healthcare, hiring, education, and finance. By making sources of bias traceable and repairable, the proposed methods support regulatory compliance and post-hoc accountability.

- **For AI governance and regulation** Repair-based approaches such as `FairFLRep` and `DShift-FRep` align with emerging regulatory frameworks (e.g., GDPR, EU AI Act, EEOC guidelines) by enabling post-deployment correction, transparency, and measurable reductions in disparity—without requiring access to original training data or retraining pipelines.

9.3 Future work

This thesis opens several promising directions for future research.

- *Extending fairness repair across tasks and settings* – Future work includes extending `FairFLRep` to multi-class classification and regression problems, as well as evaluating scalability on larger and more complex architectures.
- *Exploring alternative optimization strategies* – While PSO proved effective, other search and optimization methods may offer improved convergence, stability, or multi-objective trade-offs and warrant investigation.
- *Deepening understanding of repair depth and representation complexity* – A systematic analysis of how bias is distributed across layers—and how repair effectiveness varies with model architecture, scale, and data complexity—could inform adaptive strategies that automatically select repair depth.
- *Beyond activation-based localization* — Future work should investigate whether neuron non-activation (i.e., missing activations) can also contribute to biased behavior, extending current fault-localization assumptions.
- *Toward continual and human-in-the-loop repair* – For transformer-based LLMs, future research could explore continual or dynamic repair, integration with instruction-tuned and autoregressive models, and the incorporation of human or policy-driven constraints into the repair objective.

- *Broadening the scope of bias types in LLM repair* – In this thesis, fairness repair for transformer-based LLMs focuses on stereotyping bias, a form of representational harm that captures how social groups are portrayed in model outputs. Allocation bias– which manifests when model decisions unfairly distribute resources, opportunities, or outcomes across social groups—is not addressed in this study. Future research should investigate how internal bias representations in LLMs translate into allocation harms in downstream NLP tasks, such as hiring recommendation systems, content moderation, educational assessment, or decision-support applications.
- *Extending repair methods across transformer architectures and task paradigms* – Future work will extend the proposed localization and repair techniques to a broader range of pretrained transformer architectures, including RoBERTa and GPT-style autoregressive models. In addition, research should move beyond cloze-style probing toward downstream and task-driven evaluations, such as classification, generation, and instruction-following tasks, to better assess the robustness, generality, and real-world impact of neuron-level fairness repair.
- *Fairness in agentic AI systems* – Future work can extend this thesis to study bias in AI agents, particularly in systems where LLMs interact with external tools, environments, or decision-making modules. Such interactions may invalidate fairness guarantees established for the underlying LLM, as tool access, memory, or planning mechanisms can introduce new pathways for biased behavior.
- *Bias detection and mitigation in agentic reasoning processes* – Another promising direction is to investigate how biased behavior emerges during agent reasoning, including planning, tool selection, and intermediate reasoning steps. Analyzing internal reasoning traces, such as chains of thought or structured intermediate representations, may enable earlier detection of bias and support targeted mitigation strategies before biased outcomes are produced.
- *System-level fairness guarantees for agentic pipelines* – Future research can explore how fairness constraints defined at the model level can be preserved, adapted, or violated at the agent level, especially in multi-step, multi-tool workflows. This includes studying how bias propagates across agent components and designing intervention points that support fairness-aware agent behavior.
- *Repair mechanisms for agentic systems* – Building on neuron- and weight-level repair, future work may investigate repair strategies for agentic systems, including intervention at planning policies, tool-selection mechanisms, or memory modules, enabling fairness-aware repair beyond static model parameters.

9.4 Authors remarks

It is worth noting that parts of this thesis have been peer-reviewed and published in leading software engineering venues. The first contribution, which focuses on the systematic identification and evaluation of bias-inducing features in classical machine learning models, was published in Empirical Software Engineering (EMSE). The second contribution, presenting `FairFLRep`, a fairness-aware fault localization and repair technique for deep neural networks, was published in ACM Transactions on Software Engineering and Methodology (TOSEM, 2025). The third contribution, which investigates the diagnosis and manifestation of bias in transformer-based large language models, was accepted for publication at the ACM/IEEE International Conference on Mining Software Repositories (MSR 2026), to be held in Rio de Janeiro, Brazil. A follow-up study that proposes targeted intervention strategies—combining suppression and repair—to mitigate fairness issues in transformer models has been submitted for peer review to the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2026), research track. Finally, the work on fairness repair under distribution shifts, introducing `DShift-FRep`, is currently under review at ACM Transactions on Software Engineering and Methodology (TOSEM).

Once again, the author wishes to exceptionally thank Prof. Foutse, for his overall supervision of these studies, and for inviting other external professionals, consisting of both ML practitioners and researchers with extensive research experience in empirical software engineering, ML software testing and DNN Repair; to review and provide insightful comments throughout these studies, that greatly improved the results reported in this thesis.

The author further notes that the foundational ideas on fairness repair through fault localization and repair, inspired by traditional software engineering, were initiated during an internship at the National Institute of Informatics (NII), Tokyo, Japan, in collaboration with Polytechnique Montréal, under the supervision of Prof. Fuyuki Ishikawa, Prof. Paolo Arcaini, and Prof. Foutse Khomh. This work has since evolved to form a core component of the thesis.

REFERENCES

[1] X. Chen, X. Wang, K. Zhang, K.-M. Fung, T. C. Thai, K. Moore, R. S. Mannel, H. Liu, B. Zheng, and Y. Qiu, "Recent advances and clinical applications of deep learning in medical image analysis," *Medical image analysis*, vol. 79, p. 102444, 2022.

[2] S. Gunasekaran, P. S. Mercy Bai, S. K. Mathivanan, H. Rajadurai, B. D. Shivahare, and M. A. Shah, "Automated brain tumor diagnostics: Empowering neuro-oncology with deep learning-based MRI image analysis," *Plos one*, vol. 19, no. 8, p. e0306493, 2024.

[3] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[4] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014.

[5] M. El-Ghoul, M. M. Almassri, M. F. El-Habibi, M. H. Al-Qadi, A. Abou Eloun, B. S. Abu-Nasser, and S. S. Abu-Naser, "Ai in hrm: Revolutionizing recruitment, performance management, and employee engagement," 2024.

[6] B. Abimbola, Q. Tan, and E. A. De La Cal Marín, "Sentiment analysis of canadian maritime case law: a sentiment case law and deep learning approach," *International Journal of Information Technology*, vol. 16, no. 6, pp. 3401–3409, 2024.

[7] K. Anwar, A. Zafar, and A. Iqbal, "An efficient approach for improving the predictive accuracy of multi-criteria recommender system," *International Journal of Information Technology*, vol. 16, no. 2, pp. 809–816, 2024.

[8] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, ser. Proceedings of Machine Learning Research, S. A. Friedler and C. Wilson, Eds., vol. 81. PMLR, 23–24 Feb 2018, pp. 77–91. [Online]. Available: https://proceedings.mlr.press/v81/buolamwini18a.html

[9] J. Kuczmarski, "Reducing gender bias in Google Translate," 2018. [Online]. Available: https://blog.google/products/translate/reducing-gender-bias-google-translate/

[10] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 335–340. [Online]. Available: https://doi.org/10.1145/3278721.3278779

[11] M. Openja, G. Laberge, and F. Khomh, "Detection and evaluation of bias-inducing features in machine learning," *Empirical Softw. Engg.*, vol. 29, no. 1, Dec. 2023. [Online]. Available: https://doi.org/10.1007/s10664-023-10409-5

[12] N. Pham, H. Pham Ngoc, and A. Nguyen-Duc, "Fairness for machine learning software in education: A systematic mapping study," *Journal of Systems and Software*, vol. 219, p. 112244, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121224002887

[13] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin *et al.*, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature medicine*, vol. 24, no. 9, pp. 1342–1350, 2018.

[14] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and machine learning: Limitations and opportunities*. MIT press, 2023.

[15] M. Openja, P. Arcaini, F. Khomh, and F. Ishikawa, "Fairflrep: Fairness aware fault localization and repair of deep neural networks," *ACM Transactions on Software Engineering and Methodology*, 2025.

[16] T. Li, X. Xie, J. Wang, Q. Guo, A. Liu, L. Ma, and Y. Liu, "Faire: Repairing fairness of neural networks via neuron condition synthesis," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 1, Nov. 2023. [Online]. Available: https://doi.org/10.1145/3617168

[17] G. Maheshwari and M. Perrot, "FairGrad: Fairness aware gradient descent," *Transactions on Machine Learning Research*, 2023. [Online]. Available: https://openreview.net/forum?id=0f8tU3QwWD

[18] X. Lin, S. Kim, and J. Joo, "FairGRAPE: Fairness-Aware GRAdient Pruning mEthod for Face Attribute Classification," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*. Berlin, Heidelberg: Springer-Verlag, 2022, pp. 414–432. [Online]. Available: https://doi.org/10.1007/978-3-031-19778-9_24

[19] X. Gao, J. Zhai, S. Ma, C. Shen, Y. Chen, and Q. Wang, "FairNeuron: improving deep neural network fairness with adversary games on selective neurons," in *Proceedings of the 44th International Conference on Software Engineering*, ser. ICSE '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 921–933. [Online]. Available: https://doi.org/10.1145/3510003.3510087

[20] S. Alelyani, "Detection and evaluation of machine learning bias," *Applied Sciences*, vol. 11, no. 14, p. 6271, 2021.

[21] J. Chakraborty, S. Majumder, and T. Menzies, "Bias in machine learning software: why? how? what to do?" in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2021, pp. 429–440.

[22] A. Perera, A. Aleti, C. Tantithamthavorn, J. Jiarpakdee, B. Turhan, L. Kuhn, and K. Walker, "Search-based fairness testing for regression-based machine learning systems," *Empirical Software Engineering*, vol. 27, no. 3, pp. 1–36, 2022.

[23] F. Kamiran, A. Karim, and X. Zhang, "Decision theory for discrimination-aware classification," in *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 924–929.

[24] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 3323–3331.

[25] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," *Advances in neural information processing systems*, vol. 30, 2017.

[26] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic decision making and the cost of fairness," in *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*, 2017, pp. 797–806.

[27] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, "Software engineering for machine learning: A case study," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 2019, pp. 291–300.

[28] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine learning testing: Survey, landscapes and horizons," *IEEE Transactions on Software Engineering*, 2020.

[29] C. F. Kemerer, "Progress, obstacles, and opportunities in software engineering economics," *Communications of the ACM*, vol. 41, no. 8, pp. 63–66, 1998.

[30] C. Jones and O. Bonsignour, *The economics of software quality*. Addison-Wesley, 2012.

[31] K. Arrow, "The theory of discrimination," Princeton University, Department of Economics, Industrial Relations Section., Working Papers 403, 1971. [Online]. Available: https://EconPapers.repec.org/RePEc:pri:indrel:30a

[32] E. S. Phelps, "The statistical theory of racism and sexism," *The american economic review*, vol. 62, no. 4, pp. 659–661, 1972.

[33] "Fisher v. university of texas at austin," p. 2198, 2016.

[34] C. Ferrara, G. Sellitto, F. Ferrucci, F. Palomba, and A. De Lucia, "Fairness-aware machine learning engineering: how far are we?" *Empirical software engineering*, vol. 29, no. 1, p. 9, 2024.

[35] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," *Advances in neural information processing systems*, vol. 28, 2015.

[36] G. Voria, M. Openja, F. Khomh, G. Catolino, and F. Palomba, "Tracing stereotypes in pre-trained transformers: From biased neurons to fairer models," International Conference on Mining Software Repositories (MSR), 2026. [Online]. Available: https://arxiv.org/abs/2601.05663

[37] D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei, "Knowledge neurons in pretrained transformers," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 8493–8502.

[38] D. D. Lundstrom, T. Huang, and M. Razaviyayn, "A rigorous study of integrated gradients method and extensions to internal neuron attributions," in *International Conference on Machine Learning*. PMLR, 2022, pp. 14 485–14 508.

[39] J. Sohn, S. Kang, and S. Yoo, "Arachne: Search-based repair of deep neural networks," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 4, May 2023. [Online]. Available: https://doi.org/10.1145/3563210

[40] H. Zhang and W. K. Chan, "Apricot: A weight-adaptation approach to fixing deep learning models," in *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '19. IEEE Press, 2019, pp. 376–387. [Online]. Available: https://doi.org/10.1109/ASE.2019.00043

[41] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[42] N. Nangia, C. Vania, R. Bhalerao, and S. R. Bowman, "CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Online: Association for Computational Linguistics, Nov. 2020.

[43] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. Shieber, "Investigating gender bias in language models using causal mediation analysis," *Advances in neural information processing systems*, vol. 33, pp. 12 388–12 401, 2020.

[44] G. Voria, G. Sellitto, C. Ferrara, F. Abate, A. De Lucia, F. Ferrucci, G. Catolino, and F. Palomba, "A catalog of fairness-aware practices in machine learning engineering," in *Proceedings of the 2025 29th International Conference on Evaluation and Assessment in Software Engineering Companion*, 2025, pp. 72–81.

[45] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[46] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2012, pp. 35–50.

[47] T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," *Data mining and knowledge discovery*, vol. 21, no. 2, pp. 277–292, 2010.

[48] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International conference on machine learning*. PMLR, 2013, pp. 325–333.

[49] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3457607

[50] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ser. ITCS '12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 214–226. [Online]. Available: https://doi.org/10.1145/2090236.2090255

[51] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 259–268. [Online]. Available: https://doi.org/10.1145/2783258.2783311

[52] J. Kleinberg, J. Ludwig, S. Mullainathan, and A. Rambachan, "Algorithmic fairness," *AEA Papers and Proceedings*, vol. 108, pp. 22–27, May 2018. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/pandp.20181018

[53] Z. Chen, J. M. Zhang, M. Hort, M. Harman, and F. Sarro, "Fairness testing: A comprehensive survey and analysis of trends," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 5, Jun. 2024. [Online]. Available: https://doi.org/10.1145/3652155

[54] W. E. Wong, R. Gao, Y. Li, R. Abreu, and F. Wotawa, "A survey on software fault localization," *IEEE Trans. Softw. Eng.*, vol. 42, no. 8, pp. 707–740, aug 2016. [Online]. Available: https://doi.org/10.1109/TSE.2016.2521368

[55] M. Du, F. Yang, N. Zou, and X. Hu, "Fairness in deep learning: A computational perspective," *IEEE Intelligent Systems*, vol. 36, no. 4, pp. 25–34, 2020.

[56] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith, "Annotation artifacts in natural language inference data," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 107–112. [Online]. Available: https://doi.org/10.18653/v1/n18-2017

[57] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, "Men also like shopping: Reducing gender bias amplification using corpus-level constraints," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, M. Palmer,

R. Hwa, and S. Riedel, Eds. Association for Computational Linguistics, 2017, pp. 2979–2989. [Online]. Available: https://doi.org/10.18653/v1/d17-1323

[58] N. Kallus, X. Mao, and A. Zhou, "Assessing algorithmic fairness with unobserved protected class using data combination," *Management Science*, vol. 68, no. 3, pp. 1959–1981, 2022.

[59] I. Y. Chen, P. Szolovits, and M. Ghassemi, "Can AI help reduce disparities in general medical and mental health care?" *AMA journal of ethics*, vol. 21, no. 2, pp. 167–179, 2019.

[60] C. Dulhanty and A. Wong, "Auditing ImageNet: Towards a model-driven framework for annotating demographic attributes of large-scale image datasets," *CoRR*, vol. abs/1905.01347, 2019. [Online]. Available: http://arxiv.org/abs/1905.01347

[61] A. Gillioz, J. Casas, E. Mugellini, and O. Abou Khaled, "Overview of the transformer-based models for nlp tasks," in *2020 15th Conference on computer science and information systems (FedCSIS)*. IEEE, 2020, pp. 179–183.

[62] M. Geva, R. Schuster, J. Berant, and O. Levy, "Transformer feed-forward layers are key-value memories," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 5484–5495.

[63] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation." *Journal of Machine Learning Research*, vol. 8, no. 5, 2007.

[64] W. Lee, J. Lee, and S. Park, "Instance weighting domain adaptation using distance kernel," *Industrial Engineering & Management Systems*, vol. 17, no. 2, pp. 334–340, 2018.

[65] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2200–2207.

[66] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of machine learning research*, vol. 17, no. 59, pp. 1–35, 2016.

[67] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[68] Y. Zhang, N. Wang, S. Cai, and L. Song, "Unsupervised domain adaptation by mapped correlation alignment," *IEEE Access*, vol. 6, pp. 44 698–44 706, 2018.

[69] M. Zhang, D. Wang, W. Lu, J. Yang, Z. Li, and B. Liang, "A deep transfer model with wasserstein distance guided multi-adversarial networks for bearing fault diagnosis under different working conditions," *IEEe Access*, vol. 7, pp. 65 303–65 318, 2019.

[70] H. Singh, R. Singh, V. Mhasawade, and R. Chunara, "Fairness violations and mitigation under covariate shift," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 3–13.

[71] B. An, Z. Che, M. Ding, and F. Huang, "Transferring fairness under distribution shifts via fair consistency regularization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 582–32 597, 2022.

[72] H. Wang, J. Hong, J. Zhou, and Z. Wang, "How robust is your fairness? evaluating and sustaining fairness under unseen distribution shifts," *Transactions on machine learning research*, vol. 2023, pp. https–openreview, 2023.

[73] Y. Chen, R. Raab, J. Wang, and Y. Liu, "Fairness transferability subject to bounded distribution shift," *Advances in neural information processing systems*, vol. 35, pp. 11 266–11 278, 2022.

[74] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[75] C. Villani *et al.*, *Optimal transport: old and new*. Springer, Part of the book series: Grundlehren der mathematischen Wissenschaften, 2009, vol. 338.

[76] K. Falahkheirkhah, A. Lu, D. Alvarez-Melis, and G. Huynh, "Domain adaptation using optimal transport for invariant learning using histopathology datasets," *arXiv preprint arXiv:2303.02241*, 2023.

[77] F. Zhou, Z. Jiang, C. Shui, B. Wang, and B. Chaib-draa, "Domain generalization via optimal transport with metric similarity learning," *Neurocomputing*, vol. 456, pp. 469–480, 2021.

[78] B. Custers, T. Calders, B. Schermer, and T. Zarsky, "Discrimination and privacy in the information society," *Studies in applied philosophy, epistemology and rational ethics*, vol. 3, 1866.

[79] A. Romei and S. Ruggieri, "A multidisciplinary survey on discrimination analysis," *The Knowledge Engineering Review*, vol. 29, no. 5, pp. 582–638, 2014.

[80] C. R. Sunstein, *Legal reasoning and political conflict*. Oxford University Press, 2018.

[81] K. Lang and A. Kahn-Lang Spitzer, "Race discrimination: An economic perspective," *Journal of Economic Perspectives*, vol. 34, no. 2, pp. 68–89, 2020.

[82] S. Hajian and J. Domingo-Ferrer, "A methodology for direct and indirect discrimination prevention in data mining," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 7, pp. 1445–1459, 2012.

[83] S. Hajian, J. Domingo-Ferrer, and A. Martinez-Balleste, "Discrimination prevention in data mining for intrusion and crime detection," in *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*. IEEE, 2011, pp. 47–54.

[84] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and information systems*, vol. 33, no. 1, pp. 1–33, 2012.

[85] A. Aggarwal, P. Lohia, S. Nagar, K. Dey, and D. Saha, "Black box fairness testing of machine learning models," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019, pp. 625–635.

[86] F. Tramèr, V. Atlidakis, R. Geambasu, D. Hsu, J.-P. Hubaux, M. Humbert, A. Juels, and H. Lin, "Fairtest: Discovering unwarranted associations in data-driven applications," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017, pp. 401–416.

[87] T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," *Data mining and knowledge discovery*, vol. 21, no. 2, pp. 277–292, 2010.

[88] B. T. Luong, S. Ruggieri, and F. Turini, "k-nn as an implementation of situation testing for discrimination discovery and prevention," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 502–510.

[89] S. Ruggieri, D. Pedreschi, and F. Turini, "Data mining for discrimination discovery," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 2, pp. 1–40, 2010.

[90] K. Mancuhan and C. Clifton, "Combating discrimination using bayesian networks," *Artificial intelligence and law*, vol. 22, no. 2, pp. 211–238, 2014.

[91] D. Pedreschi, S. Ruggieri, and F. Turini, "Integrating induction and deduction for finding evidence of discrimination," in *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ser. ICAIL '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 157–166. [Online]. Available: https://doi.org/10.1145/1568234.1568252

[92] H. Guo, C. Tao, and Z. Huang, "Multi-objective white-box test input selection for deep neural network model enhancement," in *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, 2023, pp. 521–532.

[93] H. Zheng, Z. Chen, T. Du, X. Zhang, Y. Cheng, S. Ji, J. Wang, Y. Yu, and J. Chen, "NeuronFair: interpretable white-box fairness testing through biased neuron identification," in *Proceedings of the 44th International Conference on Software Engineering*, ser. ICSE '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 1519–1531. [Online]. Available: https://doi.org/10.1145/3510003.3510123

[94] S. Biswas and H. Rajan, "Fairify: Fairness verification of neural networks," in *Proceedings of the 45th International Conference on Software Engineering*, ser. ICSE '23. IEEE Press, 2023, pp. 1546–1558. [Online]. Available: https://doi.org/10.1109/ICSE48619.2023.00134

[95] Y. Xiao, A. Liu, T. Li, and X. Liu, "Latent Imitator: Generating natural individual discriminatory instances for black-box fairness testing," in *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, ser. ISSTA 2023. New York, NY, USA: Association for Computing Machinery, 2023, pp. 829–841. [Online]. Available: https://doi.org/10.1145/3597926.3598099

[96] Y. Xing, Q. Guo, X. Cao, I. W. Tsang, and L. Ma, "MetaRepair: Learning to repair deep neural networks from repairing experiences," in *Proceedings of the 32nd ACM International Conference on Multimedia*, ser. MM '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 1781–1790. [Online]. Available: https://doi.org/10.1145/3664647.3680638

[97] G. Nguyen, S. Biswas, and H. Rajan, "Fix fairness, don't ruin accuracy: Performance aware fairness repair using AutoML," in *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2023. New York, NY,

USA: Association for Computing Machinery, 2023, pp. 502–514. [Online]. Available: https://doi.org/10.1145/3611643.3616257

[98] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 2755–2763.

[99] A. Das, A. Dantcheva, and F. Bremond, "Mitigating bias in gender, age and ethnicity classification: A multi-task convolution neural network approach," in *Computer Vision – ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part I.* Berlin, Heidelberg: Springer-Verlag, 2018, p. 573–585. [Online]. Available: https://doi.org/10.1007/978-3-030-11009-3_35

[100] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach, "Women also snowboard: Overcoming bias in captioning models," in *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III.* Berlin, Heidelberg: Springer-Verlag, 2018, pp. 793–811. [Online]. Available: https://doi.org/10.1007/978-3-030-01219-9_47

[101] S. Wang, Y. Zhu, H. Liu, Z. Zheng, C. Chen, and J. Li, "Knowledge editing for large language models: A survey," *ACM Computing Surveys*, vol. 57, no. 3, pp. 1–37, 2024.

[102] S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, X. Xu, H. Li *et al.*, "Rethinking machine unlearning for large language models," *Nature Machine Intelligence*, pp. 1–14, 2025.

[103] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. [Online]. Available: https://aclanthology.org/D19-1250/

[104] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3319–3328. [Online]. Available: https://proceedings.mlr.press/v70/sundararajan17a.html

[105] Z. Z. Chen, J. Ma, X. Zhang, N. Hao, A. Yan, A. Nourbakhsh, X. Yang, J. McAuley, L. Petzold, and W. Y. Wang, "A survey on large language models for critical societal domains: Finance, healthcare, and law," *arXiv preprint arXiv:2405.01769*, 2024.

[106] R. Navigli, S. Conia, and B. Ross, "Biases in large language models: Origins, inventory, and discussion," *J. Data and Information Quality*, vol. 15, no. 2, Jun. 2023. [Online]. Available: https://doi.org/10.1145/3597307

[107] F. A. Khan, N. Sivakumar, Y. O. Wang, K. Metcalf, C. Camacho, B.-J. Theobald, L. Zappella, and N. Apostoloff, "Investigating intersectional bias in large language models using confidence disparities in coreference resolution," in *Second Conference on Language Modeling*, 2025. [Online]. Available: https://openreview.net/forum?id=zOw2it5Ni6

[108] M. Sloane, "Boolean clashes: Discretionary decision making in ai-driven recruiting," *Commun. ACM*, vol. 68, no. 5, p. 24–26, Apr. 2025. [Online]. Available: https://doi.org/10.1145/3708596

[109] M. Arzaghi, F. Carichon, and G. Farnadi, *Understanding Intrinsic Socioeconomic Biases in Large Language Models*. AAAI Press, 2025, p. 49–60.

[110] V. Hofmann, P. R. Kalluri, D. Jurafsky, and S. King, "Ai generates covertly racist decisions about people based on their dialect," *Nature*, vol. 633, no. 8028, pp. 147–154, 2024.

[111] T. Nakano, K. Shimari, R. G. Kula, C. Treude, M. Cheong, and K. Matsumoto, "Nigerian software engineer or american data scientist? github profile recruitment bias in large language models," in *2024 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2024, pp. 624–629.

[112] C. Treude and H. Hata, "She elicits requirements and he tests: Software engineering gender bias in large language models," in *2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR)*. IEEE, 2023, pp. 624–629.

[113] M. Mamta, R. Chigrupaatii, and A. Ekbal, "Biaswipe: Mitigating unintended bias in text classifiers through model interpretability," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 21 059–21 070.

[114] Z. Yu and S. Ananiadou, "Understanding and mitigating gender bias in llms via interpretable neuron editing," *arXiv preprint arXiv:2501.14457*, 2025.

[115] X. Xu, W. Xu, N. Zhang, and J. McAuley, "Biasedit: Debiasing stereotyped language models via model editing," in *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, 2025, pp. 166–184.

[116] J. Chakraborty, H. Tu, S. Majumder, and T. Menzies, "Can we achieve fairness using semi-supervised learning?" *arXiv preprint arXiv:2111.02038*, 2021.

[117] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil, "Leveraging labeled and unlabeled data for consistent fair binary classification," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[118] T. Zhang, T. Zhu, J. Li, M. Han, W. Zhou, and P. S. Yu, "Fairness in semi-supervised learning: Unlabeled data help to reduce discrimination," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1763–1774, 2020.

[119] C. Schumann, X. Wang, A. Beutel, J. Chen, H. Qian, and E. H. Chi, "Transfer of machine learning fairness across domains," *arXiv preprint arXiv:1906.09688*, 2019.

[120] T. Yoon, J. Lee, and W. Lee, "Joint transfer of model knowledge and fairness over domains using wasserstein distance," *IEEE Access*, vol. 8, pp. 123 783–123 798, 2020.

[121] J. Schrouff, N. Harris, O. Koyejo, I. Alabdulmohsin, E. Schnider, K. Opsahl-Ong, A. Brown, S. Roy, D. Mincu, C. Chen *et al.*, "Maintaining fairness across distribution shift: do we have viable solutions for real-world applications," *arXiv preprint arXiv:2202.01034*, p. 107, 2022.

[122] A. Coston, K. N. Ramamurthy, D. Wei, K. R. Varshney, S. Speakman, Z. Mustahsan, and S. Chakraborty, "Fair transfer learning with missing protected attributes," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 91–98.

[123] S. Giguere, B. Metevier, B. C. da Silva, Y. Brun, P. S. Thomas, and S. Niekum, "Fairness guarantees under demographic shift," in *International Conference on Learning Representations*, 2022.

[124] D. Mandal, S. Deng, S. Jana, J. Wing, and D. J. Hsu, "Ensuring fairness beyond the training data," *Advances in neural information processing systems*, vol. 33, pp. 18 445–18 456, 2020.

[125] A. Rezaei, A. Liu, O. Memarrast, and B. D. Ziebart, "Robust fairness under covariate shift," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9419–9427.

[126] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1, pp. 151–175, 2010.

[127] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *Advances in neural information processing systems*, vol. 31, 2018.

[128] R. Tachet des Combes, H. Zhao, Y.-X. Wang, and G. J. Gordon, "Domain adaptation with conditional distribution matching and generalized label shift," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 276–19 289, 2020.

[129] B. Li, Y. Wang, T. Che, S. Zhang, S. Zhao, P. Xu, W. Zhou, Y. Bengio, and K. Keutzer, "Rethinking distributional matching based domain adaptation," *arXiv preprint arXiv:2006.13352*, 2020.

[130] Y. Wu, E. Winston, D. Kaushik, and Z. Lipton, "Domain adaptation with asymmetrically-relaxed distribution alignment," in *International conference on machine learning*. PMLR, 2019, pp. 6872–6881.

[131] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon, "On learning invariant representations for domain adaptation," in *International conference on machine learning*. PMLR, 2019, pp. 7523–7532.

[132] D. A. Freedman, "On specifying graphical models for causation, and the identification problem," *Identification and Inference for Econometric Models*, pp. 56–79, 2005.

[133] P. W. Holland, "Statistics and causal inference," *Journal of the American statistical Association*, vol. 81, no. 396, pp. 945–960, 1986.

[134] ——, "Causation and race," *ETS Research Report Series*, vol. 2003, no. 1, pp. i–21, 2003.

[135] J. Pearl *et al.*, "Models, reasoning and inference," *Cambridge, UK: CambridgeUniversityPress*, vol. 19, no. 2, 2000.

[136] R. M. Blank, M. Dabady, C. F. Citro, and R. M. Blank, *Measuring racial discrimination*. National Academies Press Washington, DC, 2004.

[137] C. Hitchcock, "Probabilistic Causation," in *The Stanford Encyclopedia of Philosophy*, Winter 2012 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2012.

[138] D. Shin and Y. J. Park, "Role of fairness, accountability, and transparency in algorithmic affordance," *Computers in Human Behavior*, vol. 98, pp. 277–284, 2019.

[139] A. Bhattacharya, *Applied Machine Learning Explainability Techniques: Make ML models explainable and trustworthy for practical applications using LIME, SHAP, and more*. Packt Publishing Ltd, 2022.

[140] S. Barocas, M. Hardt, and A. Narayanan, "Fairness in machine learning," *Nips tutorial*, vol. 1, p. 2, 2017.

[141] P. Suppes, "A theory of probabilistic causality," 1970.

[142] H. A. Simon and H. A. Simon, "Causal ordering and identifiability," *Models of Discovery: And Other Topics in the Methods of Science*, pp. 53–80, 1977.

[143] J. de Kleer and J. S. Brown, "Theories of causal ordering," *Artificial Intelligence*, vol. 29, no. 1, pp. 33–61, 1986. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370286900901

[144] B. Johnson, Y. Brun, and A. Meliou, "Causal testing: understanding defects' root causes," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 87–99.

[145] D. P. MacKinnon, A. J. Fairchild, and M. S. Fritz, "Mediation analysis," *Annual review of psychology*, vol. 58, p. 593, 2007.

[146] C. Berzuini, P. Dawid, and L. Bernardinell, *Causality: Statistical perspectives and applications*. John Wiley & Sons, 2012.

[147] D. B. Rubin, "Estimating causal effects of treatments in randomized and nonrandomized studies." *Journal of educational Psychology*, vol. 66, no. 5, p. 688, 1974.

[148] J. M. Robins and S. Greenland, "Identifiability and exchangeability for direct and indirect effects," *Epidemiology*, pp. 143–155, 1992.

[149] J. Pearl, "Direct and indirect effects," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, p. 411–420.

[150] J. Pearl and D. Mackenzie, *The book of why: the new science of cause and effect*. Basic books, 2018.

[151] L. O. Loohuis, G. Caravagna, A. Graudenzi, D. Ramazzotti, G. Mauri, M. Antoniotti, and B. Mishra, "Inferring tree causal models of cancer progression with probability raising," *PloS one*, vol. 9, no. 10, p. e108358, 2014.

[152] C. Frye, C. Rowat, and I. Feige, "Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1229–1239, 2020.

[153] L. Willenborg and T. De Waal, *Elements of statistical disclosure control*. Springer Science & Business Media, 2012, vol. 155.

[154] L. Richiardi, R. Bellocco, and D. Zugna, "Mediation analysis in epidemiology: methods, interpretation and bias," *International journal of epidemiology*, vol. 42, no. 5, pp. 1511–1519, 2013.

[155] B. Fuglede and F. Topsoe, "Jensen-shannon divergence and hilbert space embedding," in *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, 2004, pp. 31–.

[156] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach, "A reductions approach to fair classification," in *International Conference on Machine Learning*. PMLR, 2018, pp. 60–69.

[157] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[158] A. Janos, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Heart disease data set," Jul 1998. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/heart+disease

[159] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big data*, vol. 5, no. 2, pp. 153–163, 2017.

[160] A. Kalousis, J. Prados, and M. Hilario, "Stability of feature selection algorithms: a study on high-dimensional spaces," *Knowledge and information systems*, vol. 12, no. 1, pp. 95–116, 2007.

[161] P. Arcidiacono, "Professor peter arcidiacono provides expert analysis for nonprofit's lawsuit against harvard," Jun 2018. [Online]. Available: https://econ.duke.edu/news/professor-peter-arcidiacono-provides-expert-analysis-nonprofit%E2%80%99s-lawsuit-against-harvard

[162] ——, "Expert report of peter s. arcidiacono students for fair admissions, inc. v. harvard no. 14-cv-14176-adb (d. mass)," Jun 2018.

[163] P. Arcidiacono, J. Kinsler, and T. Ransom, "Legacy and athlete preferences at harvard," *Journal of Labor Economics*, vol. 40, no. 1, pp. 133–156, 2022.

[164] P. Schmidt and F. Bießmann, "Quantifying interpretability and trust in machine learning systems," *CoRR*, vol. abs/1901.08558, 2019. [Online]. Available: http://arxiv.org/abs/1901.08558

[165] U.S. Equal Employment Opportunity Commission, "Uniform guidelines on employee selection procedures," 2025. [Online]. Available: https://www.eeoc.gov/

[166] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council." [Online]. Available: https://data.europa.eu/eli/reg/2016/679/oj

[167] D. Li Calsi, M. Duran, X.-Y. Zhang, P. Arcaini, and F. Ishikawa, "Distributed repair of deep neural networks," in *2023 IEEE Conference on Software Testing, Verification and Validation (ICST)*, 2023, pp. 83–94.

[168] X. Ren, J. Chen, F. Juefei-Xu, W. Xue, Q. Guo, L. Ma, J. Zhao, and S. Chen, "DART-SRepair: Core-failure-set guided DARTS for network robustness to common corruptions," *Pattern Recognition*, vol. 131, p. 108864, 2022.

[169] S. Tokui, S. Tokumoto, A. Yoshii, F. Ishikawa, T. Nakagawa, K. Munakata, and S. Kikuchi, "NeuRecover: Regression-controlled repair of deep neural networks with training history," in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2022, pp. 1111–1121.

[170] D. Li Calsi, M. Duran, T. Laurent, X. Zhang, P. Arcaini, and F. Ishikawa, "Adaptive search-based repair of deep neural networks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 1527–1536. [Online]. Available: https://doi.org/10.1145/3583131.3590477

[171] D. Lyu, Z. Zhang, P. Arcaini, X.-Y. Zhang, F. Ishikawa, and J. Zhao, "SpectAcle: Fault localisation of AI-Enabled CPS by exploiting sequences of DNN controller inferences," *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 4, Apr. 2025. [Online]. Available: https://doi.org/10.1145/3705307

[172] D. Lyu, Y. Li, Z. Zhang, P. Arcaini, X. Zhang, F. Ishikawa, and J. Zhao, "Fault localization of AI-enabled cyber–physical systems by exploiting temporal neuron activation," *Journal of Systems and Software*, vol. 229, p. 112475, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121225001438

[173] J. Kim, N. Humbatova, G. Jahangirova, P. Tonella, and S. Yoo, "Repairing DNN architecture: Are we there yet?" in *2023 IEEE Conference on Software Testing, Verification and Validation (ICST)*, 2023, pp. 234–245.

[174] B. Yu, H. Qi, Q. Guo, F. Juefei-Xu, X. Xie, L. Ma, and J. Zhao, "DeepRepair: Style-guided repairing for deep neural networks in the real-world operational environment," *IEEE Transactions on Reliability*, vol. 71, no. 4, pp. 1401–1416, 2022.

[175] K. Kärkkäinen and J. Joo, "FairFace: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1547–1557.

[176] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4352–4360.

[177] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*, 2008.

[178] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. USA: IEEE Computer Society, 2015, pp. 3730–3738. [Online]. Available: https://doi.org/10.1109/ICCV.2015.425

[179] P. Cortez, "Student Performance," UCI Machine Learning Repository, 2008, DOI: https://doi.org/10.24432/C5TG7T.

[180] C. Wadsworth, F. Vera, and C. Piech, "Achieving fairness through adversarial learning: an application to recidivism prediction," *CoRR*, vol. abs/1807.00199, 2018. [Online]. Available: http://arxiv.org/abs/1807.00199

[181] B. Becker and R. Kohavi, "Adult," UCI Machine Learning Repository, 1996, DOI: https://doi.org/10.24432/C5XW20.

[182] Agency for Healthcare Research and Quality, "Medical Expenditure Panel survey Public use file details," 8 2018. [Online]. Available: https://meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-192

[183] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilović, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias," *IBM Journal of Research and Development*, vol. 63, no. 4/5, pp. 4:1–4:15, 2019.

[184] B. d'Alessandro, C. O'Neil, and T. LaGatta, "Conscientious classification: A data scientist's guide to discrimination-aware classification," *Big data*, vol. 5, no. 2, pp. 120–134, 2017.

[185] K. Kloos, Q. Meertens, S. Scholtus, and J. Karch, "Comparing correction methods to reduce misclassification bias," in *Artificial Intelligence and Machine Learning*, M. Baratchi, L. Cao, W. A. Kosters, J. Lijffijt, J. N. van Rijn, and F. W. Takes, Eds. Cham: Springer International Publishing, 2021, pp. 64–90.

[186] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007. [Online]. Available: https://doi.org/10.1007/s11721-007-0002-0

[187] S. Sengupta, S. Basak, and R. A. Peters, "Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 157–191, 2018.

[188] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical problems in engineering*, vol. 2015, no. 1, p. 931256, 2015.

[189] A. P. Piotrowski, "Review of differential evolution population size," *Swarm and Evolutionary Computation*, vol. 32, pp. 1–24, 2017.

[190] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, pp. 341–359, 1997.

[191] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial intelligence review*, vol. 33, pp. 61–106, 2010.

[192] keras team, "Keras: Deep Learning for humans," 2015. [Online]. Available: https://github.com/keras-team/keras

[193] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[194] M. Openja, P. Arcaini, F. Khomh, and F. Ishikawa, "Code and results of the paper "FairFLRep: Fairness aware fault localization and repair of Deep Neural Networks"," 2025. [Online]. Available: https://github.com/openjamoses/FairFLRep

[195] A. Arcuri and L. Briand, "A practical guide for using statistical tests to assess randomized algorithms in software engineering," in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11. New York, NY, USA: ACM, 2011, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/1985793.1985795

[196] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. Charters, S. Gibbs, and A. Pohthong, "Robust statistical methods for empirical software engineering," *Empirical Software Engineering*, vol. 22, pp. 579–630, 2017.

[197] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, "Fairness in criminal justice risk assessments: The state of the art," *Sociological Methods & Research*, vol. 50, no. 1, pp. 3–44, 2021.

[198] S. Corbett-Davies, J. D. Gaebler, H. Nilforoshan, R. Shroff, and S. Goel, "The measure and mismeasure of fairness," *The Journal of Machine Learning Research*, vol. 24, no. 1, pp. 14 730–14 846, 2023.

[199] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 5684–5693.

[200] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln, *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012.

[201] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach,

R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[202] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423/

[203] OpenAI, "Gpt-4o system card," 2024. [Online]. Available: https://arxiv.org/abs/2410.21276

[204] F. C. Peña and S. Herbold, "Evaluating large language models on non-code software engineering tasks," *arXiv preprint arXiv:2506.10833*, 2025.

[205] G. Voria, F. Casillo, C. Gravino, G. Catolino, and F. Palomba, "Recover: Toward requirements generation from stakeholders' conversations," *IEEE Transactions on Software Engineering*, 2025.

[206] T. Zhang, I. C. Irsan, F. Thung, and D. Lo, "Revisiting sentiment analysis for software engineering in the era of large language models," *ACM Transactions on Software Engineering and Methodology*, vol. 34, no. 3, pp. 1–30, 2025.

[207] T. Sun, J. Xu, Y. Li, Z. Yan, G. Zhang, L. Xie, L. Geng, Z. Wang, Y. Chen, Q. Lin *et al.*, "Bitsai-cr: Automated code review via llm in practice," in *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, 2025, pp. 274–285.

[208] T. Fukuda, T. Nakagawa, K. Miyazaki, and S. Tokumoto, "Development of automated software design document review methods using large language models," in *2025 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2025, pp. 91–101.

[209] I. O. Gallegos, R. A. Rossi, J. Barrow, M. M. Tanjim, S. Kim, F. Dernoncourt, T. Yu, R. Zhang, and N. K. Ahmed, "Bias and fairness in large language models: A survey," *Computational Linguistics*, vol. 50, no. 3, pp. 1097–1179, 2024.

[210] P. Ralph, S. Baltes, D. Bianculli, Y. Dittrich, M. Felderer, R. Feldt, A. Filieri, C. A. Furia, D. Graziotin, P. He, R. Hoda, N. Juristo, B. A. Kitchenham, R. Robbes, D. Méndez, J. S. Molléri, D. Spinellis, M. Staron, K. Stol, D. A. Tamburri, M. Torchiano, C. Treude, B. Turhan, and S. Vegas, "ACM SIGSOFT empirical standards," *CoRR*, vol. abs/2010.03525, 2020. [Online]. Available: https://arxiv.org/abs/2010.03525

[211] H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, and E. Simperl, "T-REx: A large scale alignment of natural language with knowledge base triples," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1544/

[212] Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, and Y. Goldberg, "Measuring and improving consistency in pretrained language models," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1012–1031, 2021. [Online]. Available: https://aclanthology.org/2021.tacl-1.60/

[213] S. Baltes, F. Angermeir, C. Arora, M. M. Barón, C. Chen, L. Böhme, F. Calefato, N. Ernst, D. Falessi, B. Fitzgerald, D. Fucci, M. Kalinowski, S. Lambiase, D. Russo, M. Lungu, L. Prechelt, P. Ralph, R. van Tonder, C. Treude, and S. Wagner, "Guidelines for empirical studies in software engineering involving large language models," 2025. [Online]. Available: https://arxiv.org/abs/2508.15503

[214] Z. Chen, J. M. Zhang, F. Sarro, and M. Harman, "Fairness improvement with multiple protected attributes: How far are we?" in *Proceedings of the IEEE/ACM 46th international conference on software engineering*, 2024, pp. 1–13.

[215] B. Rosner, R. J. Glynn, and M.-L. T. Lee, "The wilcoxon signed rank test for paired comparisons of clustered data," *Biometrics*, vol. 62, no. 1, pp. 185–192, 2006.

[216] J. H. Zar, "Spearman rank correlation," *Encyclopedia of biostatistics*, vol. 7, 2005.

[217] A. Parziale, G. Voria, G. Giordano, G. Catolino, G. Robles, and F. Palomba, "Fairness on a budget, across the board: A cost-effective evaluation of fairness-aware practices across contexts, tasks, and sensitive attributes," *Information*

and Software Technology, vol. 188, p. 107858, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584925001971

[218] N. A. de Pieuchon, A. Daoud, C. T. Jerzak, M. Johansson, and R. Johansson, "Benchmarking debiasing methods for llm-based parameter estimates," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 19 768–19 783.

[219] P. Han, R. Kocielnik, A. Saravanan, R. Jiang, O. Sharir, and A. Anandkumar, "Chatgpt based data augmentation for improved parameter-efficient debiasing of llms," in *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, 2024, pp. 73–105.

[220] J. Chen and J. Mueller, "Automated data curation for robust language model fine-tuning," *arXiv preprint arXiv:2403.12776*, 2024.

[221] S. Das, M. Romanelli, C. Tran, Z. Reza, B. Kailkhura, and F. Fioretto, "Low-rank finetuning for llms: A fairness perspective," *arXiv preprint arXiv:2405.18572*, 2024.

[222] H. Liu, "Structural regularization and bias mitigation in low-rank fine-tuning of llms," *Transactions on Computational and Scientific Methods*, vol. 3, no. 2, 2023.

[223] B. Peng, K. Chen, M. Li, P. Feng, Z. Bi, J. Liu, X. Song, and Q. Niu, "Securing large language models: Addressing bias, misinformation, and prompt attacks," *arXiv preprint arXiv:2409.08087*, 2024.

[224] Z. Huang, Z. Qiu, Z. Wang, E. M. Ponti, and I. Titov, "Post-hoc reward calibration: A case study on length bias," *arXiv preprint arXiv:2409.17407*, 2024.

[225] S. Krishna, J. Ma, D. Slack, A. Ghandeharioun, S. Singh, and H. Lakkaraju, "Post hoc explanations of language models can improve language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 65 468–65 483, 2023.

[226] R. Dunford, Q. Su, E. Tamang, and A. Wintour, "The pareto principle," *The Plymouth Student Scientist*, vol. 7, no. 1, pp. 140–148, 2014.

[227] V. Roger, J. Farinas, and J. Pinquier, "Deep neural networks for automatic speech processing: a survey from large corpora to limited data," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2022, no. 1, p. 19, 2022.

[228] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, "Recent developments in recommender systems: A survey," *IEEE Computational Intelligence Magazine*, vol. 19, no. 2, pp. 78–95, 2024.

[229] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, B. Schölkopf *et al.*, "Covariate shift by kernel mean matching," *Dataset shift in machine learning*, vol. 3, no. 4, p. 5, 2009.

[230] F. Johansson, U. Shalit, and D. Sontag, "Learning representations for counterfactual inference," in *International conference on machine learning*. PMLR, 2016, pp. 3020–3029.

[231] I. D. Raji and J. Buolamwini, "Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 429–435.

[232] B. Kim, H. Kim, K. Kim, S. Kim, and J. Kim, "Learning not to learn: Training deep neural networks with biased data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9012–9020.

[233] J. Zhang, A. Menon, A. Veit, S. Bhojanapalli, S. Kumar, and S. Sra, "Coping with label shift via distributionally robust optimisation," *arXiv preprint arXiv:2010.12230*, 2020.

[234] H. Li, L. Ding, M. Fang, and D. Tao, "Revisiting catastrophic forgetting in large language model tuning," *arXiv preprint arXiv:2406.04836*, 2024.

[235] C. Wei, K. Shen, Y. Chen, and T. Ma, "Theoretical analysis of self-training with deep networks on unlabeled data," *arXiv preprint arXiv:2010.03622*, 2020.

[236] T. Cai, R. Gao, J. Lee, and Q. Lei, "A theory of label propagation for subpopulation shift," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1170–1182.

[237] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[238] S. Kulinski, S. Bagchi, and D. I. Inouye, "Feature shift detection: Localizing which features have shifted via conditional distribution tests," *Advances in neural information processing systems*, vol. 33, pp. 19 523–19 533, 2020.

[239] S. Magliacane, T. Van Ommen, T. Claassen, S. Bongers, P. Versteeg, and J. M. Mooij, "Domain adaptation by using causal inference to predict invariant conditional distributions," *Advances in neural information processing systems*, vol. 31, 2018.

[240] H. Kim and A. Mnih, "Disentangling by factorising," in *International conference on machine learning*. PMLR, 2018, pp. 2649–2658.

[241] D. Madras, E. Creager, T. Pitassi, and R. Zemel, "Learning adversarially fair and transferable representations," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3384–3393.

[242] H. Zhao, A. Coston, T. Adel, and G. J. Gordon, "Conditional learning of fair representations," *arXiv preprint arXiv:1910.07162*, 2019.

APPENDIX A DATA-GENERATING PROCESS (STRUCTURAL EQUATIONS)

We construct a synthetic causal model involving both binary and continuous variables. The data are generated using the following structural equations:

$$\begin{aligned}
\mathcal{A}_\gamma &\sim \text{Bernoulli}\Big(\sigma(\gamma \cdot \lambda_1 \cdot C + u_1)\Big), \\
\mathcal{R} &\sim \mathcal{N}(0,1) + \lambda_2 \mathcal{A} + u_2, \\
Y &\sim \text{Bernoulli}\Big(\sigma(\lambda_3 \mathcal{A}_\gamma + \lambda_4 \mathcal{R} + u_3)\Big), \\
\mathcal{T}_\gamma &= \lambda_5 Y + \lambda_6 \mathcal{R} + \lambda_7 \mathcal{A}_\gamma + \mathcal{N}\Big(0, \gamma \cdot \lambda_8 \cdot C\Big) + u_4,
\end{aligned}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ s the logistic (sigmoid) link and $u_1, u_2, u_3 \sim \mathcal{N}(0, 0.82)$, $u_4 \sim \mathcal{N}(0, 1.0)$.

The fixed parameter values used in experiments are

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = ((0.2, -0.1, -0.8, 0.8)) \text{ and } (\lambda_5, \lambda_6, \lambda_7, \lambda_8) = (0.8, 0.1, -0.8, 0.2)$$

In this case,

- $\mathcal{A}_\gamma$ the sensitive (binary) attribute; its marginal is perturbed by γ through the term $\gamma\lambda_1 C$ inside the logistic link.
- $\mathcal{R}$ is a continuous intermediate variable (Gaussian baseline plus an effect of $\mathcal{A}_\gamma$).
- Y is an intermediate binary variable (depends on $\mathcal{A}_\gamma$ and $\mathcal{R}$ via a logistic link).
- $\mathcal{T}_\gamma$ is the target variable constructed from Y, $\mathcal{R}$, and $\mathcal{A}_\gamma$, plus an additive Gaussian noise whose variance is scaled by $\gamma\lambda_8 C$
- C is an exogenous covariate used to modulate the soft intervention; increasing γ corresponds to stronger soft interventions that change the marginal distributions of $\mathcal{A}_\gamma$ and $\mathcal{T}_\gamma$.