POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Deep transfer learning-based performance prediction of URLLC in independent and not necessarily identically distributed Interference Networks
Auteurs: Authors:	Mujtaba Ghous, Thanh Luan Nguyen, Tri Nhu Do, & Georges Kaddoum
Date:	2024
Type:	Article de revue / Article
Référence: Citation:	Ghous, M., Nguyen, T. L., Do, T. N., & Kaddoum, G. (2024). Deep transfer learning-based performance prediction of URLLC in independent and not necessarily identically distributed Interference Networks. IEEE Access, 12, 99071-99093. https://doi.org/10.1109/access.2024.3427118

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Document publié chez l'éditeur officiel
Document issued by the official publisher

RESEARCH ARTICLE

Deep Transfer Learning-Based Performance Prediction of URLLC in Independent and Not Necessarily Identically Distributed Interference Networks

MUJTABA GHOUS [ID] [1]**, THANH LUAN NGUYEN** [ID] [1]**, (Graduate Student Member, IEEE),**
TRI NHU DO [ID] [2]**, (Member, IEEE), AND GEORGES KADDOUM** [ID] [1,3]**, (Senior Member, IEEE)**
[1]Department of Electrical Engineering, the École de Technologie Supérieure (ÉTS), Université du Québec, Montreal, QC H3C 1K3, Canada
[2]Department of Electrical Engineering, Polytechnique Montréal, Montreal, QC H3T 1J4, Canada
[3]Artificial Intelligence and Cyber Systems Research Center, Lebanese American University, Beirut 03797751, Lebanon

Corresponding author: Mujtaba Ghous (mujtaba.ghous.1@ens.etsmtl.ca)

ABSTRACT In the rapidly evolving sixth-generation (6G) networks, achieving ultra-high reliability for short packets poses a crucial challenge for network designers as classical Shanon capacity bounds become obsolete. This paper addresses this imperative task and presents a performance analysis of ultra-reliable low-latency communication (URLLC) systems in clustered networks with short packet communication (SPC) in the finite blocklength regime. In clustered networks, where multiple interferers are present, multiple ground users are grouped into a cluster and communicate with a central node called a cluster head (CH). The system adopts a CH as a wireless relay to accomplish URLLC between a base station and ground users. In this context, we first derive a closed-form expression for the overall block error rate of the system. The analysis is based on a theoretical model that considers the packet size, blocklength, and maximum achievable rate. Moreover, the paper proposes a transfer learning approach using fine-tuned target models with domain adaptation for real-time prediction of the system's performance. The prediction is necessary to accurately assess the performance of URLLC in statistically independent and not necessarily identically distributed (Non-IID) interference networks. Deep transfer learning is utilized to develop a model that can generalize interference scenario and provide reliable predictions. The transfer learning approach involves using a source model and fine-tuning it with domain-specific data to improve the prediction accuracy. Overall, the paper provides insights into the performance of URLLC systems with SPC and proposes an approach to improve the real-time prediction's accuracy. The proposed approach has the potential to facilitate the deployment of URLLC systems in practical applications where real-time performance prediction is crucial.

INDEX TERMS Ultra-reliable low latency communication (URLLC), transfer learning (TL), block error rate (BLER), convolutional neural network (CNN), performance analysis, non-IID interference network, machine learning-based performance prediction.

I. INTRODUCTION

The potential of short-packet communication (SPC) to deliver high reliability and low end-to-end (e2e) latency has led to its recognition as one of the core technologies

The associate editor coordinating the review of this manuscript and approving it for publication was Adao Silva [ID].

for beyond fifth-generation (B5G) and sixth-generation (6G) networks [1]. The authors in [2] provide a thorough examination of short-packet transmission in 5G networks by discussing the challenges in effectively sending small-sized packets and proposing new methods to improve performance. Their study demonstrates the practical applications of SPC in real-life scenarios, proving its viability and advantages

through comprehensive simulations and tests. Whereas in [3], the authors focus on improvements in SPC for upcoming 6G networks, where they investigate the particular requirements of ultra-reliable and low latency applications and suggest innovative approaches to enhance the transmission of small packets. Their study contains real-world applications and performance assessments that support the practical use of SPC. SPC supports a broad range of ultra-reliable low latency communication (URLLC) applications, including the Internet-of-Things (IoT), factory automation, high-speed trains, and drones [4]. Recent URLLC research has mostly focused on SPCs with single-hop or dual-hop transmissions for industrial automation, where the reliability and latency must meet stringent conditions of 99.999% and 1 ms, respectively.

From the perspective of the physical layer, the communication systems has witnessed substantial research works across wide spectrum of applications under the premise of infinite blocklength [5], [6]. However, it is clear that the assumption of infinite blocklength is no longer suitable for URLLC applications with SPC. Future applications that require high reliability and low latency simultaneously, such as in tactical wireless networks (TWNs), undoubtedly face substantial challenges due to the performance losses in both reliability and latency resulting from the limited blocklength. Moreover, drawing inferences about the performance of SPC systems using a Shannon information-theoretic framework based on infinite blocklength may be inaccurate. Therefore, it is advisable to reconsider the analysis and design of reliable transmissions under finite blocklength (FBL).

A. RELATED WORK

Transmission latency holds a pivotal position as the most frequently examined latency component in physical layer design [7]. In the FBL regime, the blocklength affects the throughput and is inversely correlated with the block transmission delay. However, the majority of earlier research using FBL assumed a fixed blocklength, where blocklength optimization was only considered in [8] and [9]. In [8] the authors investigate a multi-reconfigurable intelligent surface (RIS)-assisted rate-splitting multiple access (RSMA) to prompt URLLC service. In the aforementioned study, simulation results demonstrated that their proposed RIS-aided RSMA system surpasses both non orthogonal multiple access (NOMA) and SDMA systems in terms of sum throughput by approximately 10% and 17%, respectively. Additionally, employing RSMA with FBL was found to achieve equivalent transmission data rates as traditional NOMA and SDMA schemes with infinite block-length (IFBL) transmission. These findings validate the remarkable attributes of RIS-RSMA, including high-reliability, low latency, and enhanced spectral efficiency. The authors in [9] investigated a wireless powered IoT network using SPC, in which a hybrid access point (HAP) transmits radio frequency signals to wireless power scattered users before the users in turn send short data packets to the HAP. The authors specifically constructed two problems, using transmission time and each user's

packet error rate as variables: a total effective-throughput maximization problem and a total transmission time reduction problem. As stated in [10], the assumption that the channels between different transmit-receive antenna pairs are statistically identical typically holds when antennas in the system are co-located; consequently, the large-scale fading of the signaling branches are identical to each other. However, in the case where the antennas are distributed, such as in distributed antenna systems (DAS), the large-scale fading and small-scale fading on different links are expected to be different, which results in a statistically independent and not necessarily identically distributed (Non-IID) fading environment. Nevertheless, in these works on blocklength optimization together with re-transmission or power design, only homogeneous traffic requirements were taken into account, which is unable to handle the URLLC traffic in time-sensitive services.

In order to address the problems in the above-mentioned works, deep convolutional neural networks (CNNs) have recently gained attention as an effective method for addressing a range and time-sensitive issues, including resource allocation, queue management, and congestion control in contemporary wireless networks and IoT systems [11]. Since CNN models can accurately estimate desired performance metrics from high-dimensional raw data, even in dynamic environments and complicated radio scenarios where mathematical derivations are intractable, CNN-based prediction assist to improve the performance of real-time IoT networks. In [12], the authors presented a useful deep CNN for predicting the secrecy performance of a 6G wireless networks. Through comprehensive analyses and predictions, the study shed light on the security aspects of 6G networks, offering valuable insights for enhancing their confidentiality and reliability. It is useful to include projected quantitative requirements for 6G wireless networks. Projected metrics such as ultra-low latency ≤ 1 milliseconds, peak data rates exceeding 1 terabit per second (1 Tbps), massive device connectivity up to millions of devices per square kilometer ($> 10^6$), and spectral efficiency exceeding (> 100 $bits/s/Hz/km^2$) can be summarized as discussed in [13]. The research in [14] focused on OP prediction using CNN for IoT communication networks. Specifically, the authors address the issues of complexity and energy consumption in mobile IoT networks by employing transmit antenna selection (TAS) and cooperative communication schemes. To assess the performance of these networks, analytical OP expressions are provided to build a dataset for training the proposed CNN-based prediction model. The proposed prediction method outperforms other methods, such as radial basis function (RBF), generalized regression (GR), Elman, and extreme learning machine (ELM), in terms of prediction accuracy. Furthermore, time complexity of CNN models was investigated in [15] to identify crucial parameters influencing computational complexity. The results reported in [15] provide insights for designing CNN models. The authors highlighted differences in computational operations

between convolution and dense layers, and the impact of number of layers on computational complexity. The authors also introduced factor ϕ expressed by the phase imperfections that can be generally modeled as a random variable following the Von Mises distribution with zero mean and concentration parameter $\kappa \in [0, \infty]$. In another relevant study [16], the authors categorized the 6G URLLC vision into three connectivity characteristics with their corresponding unique QoS requirements. The authors considered an indoor downlink transmission scenario for the coexistence of WiFi and new radio unlicensed (NR-U) systems in the 5 GHz band, which occurs in a $120m \times 80\ m$ space with a $40m$ distance between two neighboring nodes. To verify the efficacy of ML solutions, both centralized deep reinforcement learning (CDRL) and federated DRL algorithms applied for downlink URLLC channel access optimization problems. The authors in [17] analyzed the performance of state-of-the-art ML-based channel prediction algorithms in high-mobility URLLC scenarios. The study evaluated two popular classes of ML-based algorithms: (i) algorithms using autoregression, and (ii) algorithms predicting the channel based on trained neural networks. In contrast to several previous studies that used various channel models for performance evaluation, this study compared the algorithms under the same conditions using modern channel models where the user speed were modeled at 15, 30, and 60 kmph, and the time axis was divided into slots having a duration of 0.5 *ms*. In URLLC, latency assumed critical significance, particularly concerning time-sensitive data gathering and accommodating real-time environmental behavior [18].

In order to meet the need for URLLC in next-generation networks, it is essential to conduct accurate network time delay analysis. However, current methods used to predict network latency in URLLC situations tend to ignore cases when several downstream activities are involved, such as identifying operational conditions of the network, detecting network attack, and scheduling network resources. Furthermore, it is important to strengthen adaptability of these models. The contrastive convolutional structure for network delay (CCSND), which is an adaptive contrastive learning framework with a temporal representation extraction module designed to capture statistical characteristics and temporal periodicity, was first presented in [19]. Here, the MSE and MAE values of the proposed CCSND method were on average 18.2% and 15.1% lower than CoST [20], 21.8% and 14.6% lower than TS2Vec [21], 26.9% and 19.0% lower than Autoformer [22], 38.7% and 23.4% lower than Informer [23], and 46.4% and 32.1% lower than Transformer [24]. In URLLC networks the performance of 5G networks can be affected by various factors, such as interference, congestion, signal attenuation, or attacks, could impact performanceVarious issues, like interference, congestion, signal attenuation, or attacks, may affect the performance of 5G networks, resulting in packet loss and retransmissions. While high retransmission rates were previously considered to be essential to improve network

strength, they frequently pointed to underlying problems that network operators had to address. To predict downlink retransmissions in 5G networks, in [25], a deep learning-based method was presented, and showed significant gains over conventional prediction algorithms.

In order to support numerous primary users, in [26], the authors investigated short-packet communications within multiple-input multiple-output underlay cognitive multihop relay networks. In order to guarantee URLLC, transceivers were used to both send and receive brief packets. In order to predict system performance metrics such end-to-end throughput, energy efficiency, latency, and reliability in real-time scenarios, several innovative machine learning (ML)-assisted estimators were adapted based on analytical results. The authors in [17] analyzed the performance of state-of-the-art ML-based channel prediction algorithms in high-mobility URLLC scenarios. The study evaluated the following two popular classes of ML-based algorithms: (i) algorithms using autoregression and (ii) algorithms predicting the channel based on trained neural networks. In contrast to several previous studies using various channel models for performance evaluation, this study compared the algorithms under the same conditions using modern channel models. Full-duplex (FD) energy-harvesting Internet-of-Things (IoT) networks deployed in automated industries in [27]. In order to improve BLER and system performance, it presented a full relay selection (FRS) strategy accounting for residual interference models into account. Furthermore, to enable effective packet transmissions, a deep learning framework based on FRS was created for real-time average BLER and throughput prediction.

In [28], the authors provide a generic precoder design challenge and a unique predictive transmission methodology. The orthogonal time frequency space (OTFS) transmission technique was the primary focus of this study. To train a DL model, the authors used historical delay-Doppler domain channels (DDCs). This methodology predicts the precoder a specific time frame by extracting implicit characteristics from these DDCs, thus avoiding the requirement for real-time acquisition of instantaneous channel status information at the transmitter (ICSIT). Using an unsupervised learning mechanism, the DL-based predictive precoder design approach is pragmatic and flexible, promoting improvements in URLLC performance and driving innovation in wireless communication systems. Aiming to contribute to the advancement of URLLC by leveraging innovative methodologies and machine learning techniques and thus to optimize system performance and reliability in wireless communication systems, in [26], the authors focused on URLLC in short-packet communications in multiple-input multiple-output underlay cognitive multihop relay networks. In order to estimate system performance measures including throughput, latency, and reliability in real-time scenarios, the aforementioned study investigated machine learning-aided estimators and developed closed-form expressions for end-to-end (E2E) block error rate (BLER). Furthermore, asymptotic E2E

BLER was minimized by deriving optimal power-allocation and relay-location techniques that met URLLC limits and accurately tolerated interference power with minimal computing complexity and significant power savings.

By utilizing pre-trained models and knowledge transferred from one domain to another, transfer learning has evolved as a remedy for the problem of insufficient data and the computational cost associated with training deep neural networks from scratch [29]. On the other hand, learning from a small dataset poses significant challenges due to the limited amount of available examples. These challenges include inadequate representation of the target task, overfitting, reduced model capacity, and limited exploration of the feature space. In this context, transfer learning saves time by allowing models to learn from larger datasets and transfer this knowledge using trained weights to smaller ones, hence enhancing generalization and performance [30]. Transfer learning is a helpful approach for dealing with the problems of managing an overwhelming volume of data and learning from small samples, as it effectively utilizes prior knowledge obtained from pre-trained models on large datasets to enhance the performance of learning tasks in new domains or with limited data. As the name implies, this method transfers the parameters from the previously learned models to the new model, speeding up model training [31], [32]. Transfer learning can be categorized into three types: inductive transfer learning approaches, unsupervised transfer learning, and transductive transfer learning approaches. The transductive transfer learning approach leads to domain adaptation in the case of single task with different domains [33].

Recently, deep neural networks (DNNs) were combined with transfer learning, where the combined approach is referred to as deep transfer learning. Transfer learning incorporated into the deep learning system for Industrial IoT (IIoT) networks was proposed in [34] so that training data and test data can come from various feature spaces and distributions. Deep learning networks may construct models for new but related tasks in the target domain without utilizing new data by applying transfer learning, which transfers intrinsic knowledge learned by the deep learning network from the source domain to the target domain [33]. In [35] and [36], transductive transfer learning methodologies were suggested as a way to make use of the knowledge obtained from other signal types and environments. For spectrum sensing in cognitive radio networks, the authors of [35] suggested a fine-tuning source model-based transfer learning approach. A baseline CNN model trained using data from different types of signals was specifically applied to a target task (one type of signal), where the model was then fine-tuned using the target data. Similarly, a transfer learning technique that fine-tunes pre-trained models was discussed in [36]. The method specifically applies a CNN trained with various signal and noise data types to the target task and fine-tunes the model with target data. This transfer learning method is quite similar to that in [35], except that [35] not

take noise data into account. To the best of our knowledge, transfer learning-based performance prediction of URLLC in Non-IID interference networks has not yet been investigated in-depth.

In our approach, we create analytical models that are based on fixed and well-defined configurations and are derived from theoretical principles and assumptions [37]. Neural network models are utilized to predict block error rate (BLER) across a broader and possibly more random range of configurations. This approach allows the model to learn from a diverse set of data, making it capable of handling real-world variations and complexities [38]. By combining these two approaches, we can leverage the strengths of analytical models for well-understood scenarios and neural network models for handling a broader, potentially more complex range of configurations in our BLER predictions.

B. MOTIVATIONS AND CONTRIBUTIONS

Analytical formulas for calculating the average BLER are generally complex due to the involvement of complex functions, e.g., Q-function, render them infeasible for time-sensitive scenarios, such as real-time, low-latency, and URLLC systems. On the other hand, while CNNs are excellent at providing accurate predictions, they often require a large amount of training data, with potentially high network complexity for accurate predictions. However, having a large dataset is infeasible in dynamic environments. Similarly, complex neural network models are unsuitable for URLLC-based application.

In response, our research aims to find a balance by introducing a more efficient model that avoids the computational challenges of the analytical method and the complexity associated with CNNs, allowing for both speed and accuracy in predicting BLER for real-time applications. For this, we propose a transfer learning method for URLLC-based clustered wireless networks with short packet communication. Transfer learning allows us to leverage knowledge gained from a different but related task or dataset. Instead of starting from scratch, we design machine learning models that exploit pre-learned features from a source model. The designed models require small dataset, and significantly reduces the computational complexity and training time, making it more applicable for real-time applications and for dynamic wireless environments. To the best of the authors' knowledge, such a URLLC system has not been considered in the literature. We summarize our contributions as follows:

- Utilizing the Q-function approximation, we derive closed-form expressions for the average per-hop BLERs and the average e2e BLER of the considering SPC system consisting of multiple users operating on a shared frequency resource with Non-IID co-channel interference (CCI) at both the cluster head and individual users. It is noted that the presence of CCI increases the
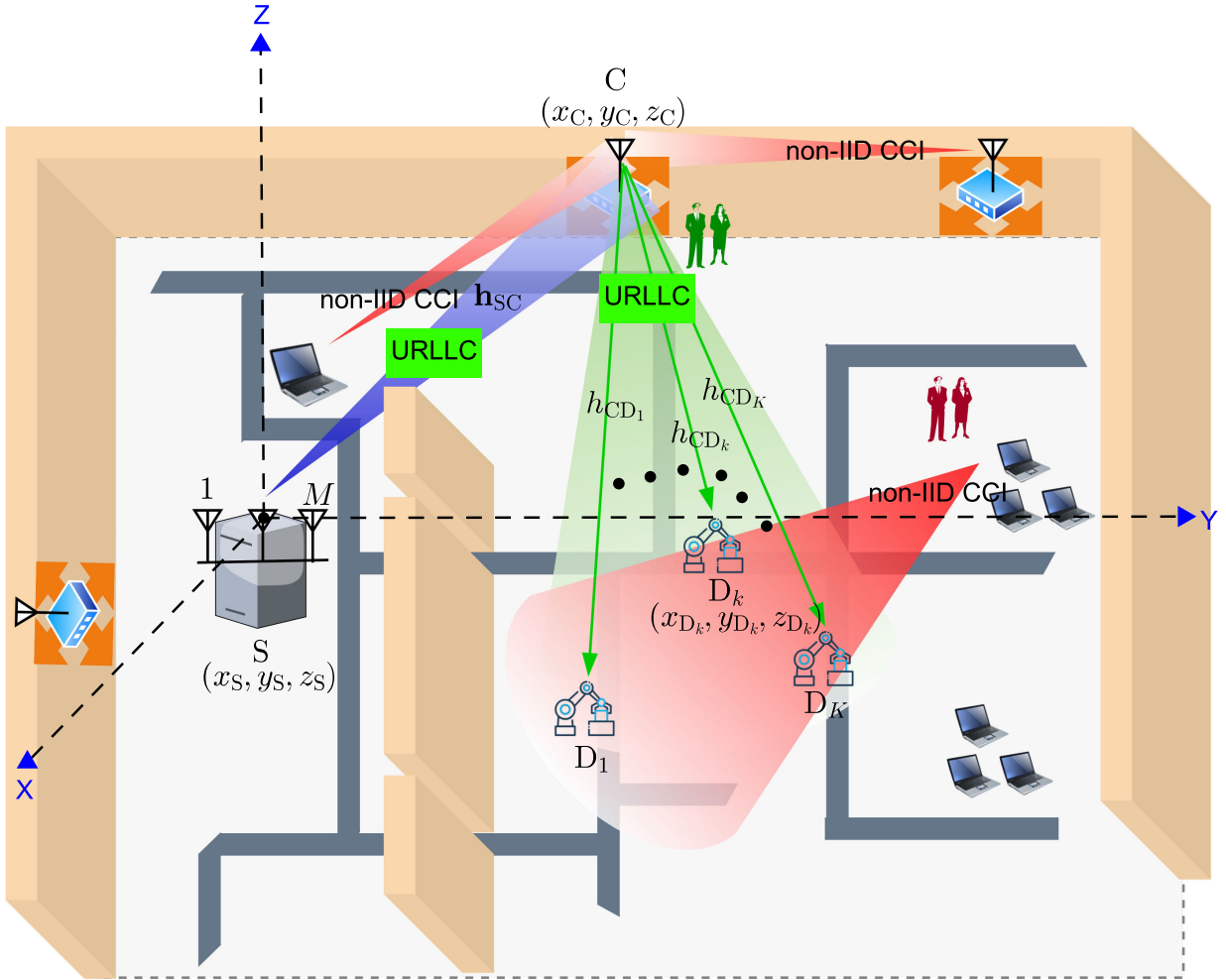
FIGURE 1. Illustration of the URLLC system in the presence of non-IID interferers, where a *M*-antenna base station (S) communicate with the user equipments (D_k) with the help of a cluster head (C).

TABLE 1. List of mathematical notations.

Notation	Description
S	Source
C	Cluster head
D_K	K-th ground user
$\mathcal{I}$	Interferer
$\mathbf{h}_{SC}$	Channel vector from source-to-cluster head
$\mathbf{h}_{I_iC}$	Channel vector from interferer-to-cluster head
N	Number of antennas at transmitter
P_S	Transmit power of the source
P_{I_i}	Transmit power of the interferer
x_{I_i}.	Normalized energy signal of the interferer
$\sqrt{l}_{SC}$	Large scale fading of the S-to-C link
$\mathbf{w}_S$	Beamforming vector at S
m_{I_iC}	Shape parameter of the interfere, $i \in \mathcal{I}$
γ_{SC}	Instantaneous SNR from S-to-C
γ_{SC}	SINR at the C
$f_{I_C}(x)$	PDF of the aggregated interference power at C
$F_{I_C}(x)$	CDF of the aggregated interference power at C
y_{CD_k}	Received signal at D_k
Γ_{sys}	SINR of the system
ϕ_{SC}	Event during S-to-C transmission
ϕ_{CD}	Event during C-to-D transmission

computational complexity of the resulting expressions due to the presence of a bivariate Meijer G-function.

- To mitigate the complexity of the derived expressions, we propose a novel approximation to express the average e2e BLER in terms of the average per-hop per-user BLER using elementary functions, effectively addressing the analytical complexity challenge. Using these derived expressions, we assess system reliability through the error floor, asymptotic performance, and the diversity order.

- We propose a transfer learning-based performance prediction approach for real-time accurate predictions of the overall average BLER of the system. Numerical results show that the proposed fine-tuned target model requires less training data while achieving comparable prediction accuracy compared to the source model.

- Through numerical and simulation results, we validate the use of short packet communication in a clustered URLLC system. Simulation results show that the derived closed-form expressions provide a reasonable approximation for the downlink average e2e BLER of the system. In addition, the results show

that IID interference yields lower average e2e BLER compared to Non-IID interference. We also investigate the asymptotic analysis and discuss the error floor results, where the average e2e BLER found unaffected by the interference power distribution and rather the total Interference-to-Noise Ratio (INR).

- By incorporating domain adaptation techniques, we improve the accuracy of our system model. Specifically, a discrepancy between the source and target domain decreases, the accuracy of the proposed transfer learning model increases.
- By investigating the BLER, exploring the impact of block length (M), investigating convergence and performance of transfer learning, and evaluating the impact of domain adaptation and transferred knowledge, we provide readers with invaluable insights into the potential performance and robustness of our proposed approach. This comprehensive analysis emphasizes the practical relevance and applicability of our research in real-world scenarios.

II. SYSTEM MODEL

We consider a system model consisting of a base station (S) as a gateway, a tentative cluster head (C), multiple interferers $\mathcal{I} \in \{1, 2, \ldots, I\}$, where I denotes the number of ambient interferers, and $K \geq 1$ ground users, denoted as $D_1, D_2, \ldots, D_K$. Due to the presence of multiple interferers, there is aggregated interference at C, and hence C experiences aggregated INR during the communication from S to C. This considered clustering strategy is depicted in Fig. 1, where the cluster head forwards the information from S to all ground users. Direct connectivity between S and D_k, for $k \in \{1, 2, \ldots, K\}$, is missing due to several obstacles. Therefore, it is supposed that S delivers short packets to D_k with the help of C, which functions as a relay and utilizes a decode-and-forward method while operating in half-duplex mode. C broadcasts the same information to each user. When significant message broadcasting delays are encountered, delivering urgent information requires URLLC. This need becomes particularly crucial in scenarios involving downlink multi-casting and broadcasting, where communication channels are Non-IID.

The S has N antennas while C and D_k have single antennas and can acquire perfect CSI. The choice of a single antenna cluster head is deliberate, as it represents a realistic scenario where resource-constrained devices, such as IoT devices or sensors, may have limited antenna capabilities. This aligns with the challenges faced in real-world deployments of URLLC, where devices frequently have limited power and hardware capabilities [39]. The transmission is completed in two phases, i.e., phase I: S-to-C and phase II: C-to-D_k. During phase I, S delivers messages to C with transmit power P_S. If C successfully decodes the received signals, in phase II, it delivers the decoded signals to the D_k with transmit power P_C.

A. S-TO-C COMMUNICATION

The received signal at C from S is given as

$$y_{SC} = \sqrt{P_S}\sqrt{\ell_{SC}}h_{SC}^H w_S x_S + \sum_{i \in \mathcal{I}} \sqrt{\ell_{I_iC}}h_{I_iC}\sqrt{P_{I_i}}x_{I_i} + n_C, \tag{1}$$

where x_S is the normalized signals transmitted by the source node S for C, n_C is the complex additive white Gaussian noise (AWGN) with zero-mean and variance σ^2, $\mathbf{w}_S \in \mathbb{C}^{N \times 1}$ is the beamforming vector at S. Herein, $\sqrt{\ell_{SC}}$ is the large-scale fading of the S-to-C link, $h_{SC}^{(1)}, h_{SC}^{(2)}, \ldots, h_{SC}^{(N)}$ are the complex small-scale fading, h_{I_iC} and ℓ_{I_iC} denote the complex small-scale fading and the path loss from interferer $i \in \mathcal{I}$ to C, respectively. P_{I_i} and x_{I_i} are the transmit power and the normalized unit energy signal of the interferer $i \in \mathcal{I}$, respectively. We assume that $|h_{I_iC}|^2$'s are Non-IID Gamma random variables, each with unit mean and respective shape parameter m_{I_iC}. The system model's large-scale fading is presented as follows.

1) CHANNEL MODEL OF THE S-TO-C LINK

We consider the 3GPP Urban Micro (UMi) path loss model [40, Table B.1.2.1-1]. In particular, the path loss from a transmitter Tx $\in \{S, C\}$ to a receiver Rx $\in \{C, D_1, D_2, \ldots, D_K\}$ is formulated as

$$\ell_{\text{Tx-Rx}} [\text{dB}] = G_{\text{Tx}} + G_{\text{Rx}} \\ - 22.7 - 26 \log_{10}(f_c) - 36.7 \log_{10}\left(\frac{d_{\text{Tx-Rx}}}{d_0}\right), \tag{2}$$

where $d_0 = 1$ [m] is the reference distance, f_c [GHz] is the carrier frequency, $d_{\text{Tx-Rx}}$ [m] is the distance from the Tx to Rx, G_{Tx} and G_{Rx} are the antenna gains (in dBi) at the Tx and the Rx, respectively.

Moreover, for decoding x_S, the instantaneous SNR γ_{SC}, is expressed as $\gamma_{SC} = \bar{\gamma}_S \ell_{SC} |\mathbf{h}_{SC}^H \mathbf{w}_S|^2$, where $\bar{\gamma}_S \triangleq \frac{P_S}{\sigma^2}$, $(\cdot)^H$ denotes the conjugate transpose operator and $\mathbf{w}_S = \frac{\mathbf{h}_{SC}}{\|\mathbf{h}_{SC}\|}$ is the beamforming vector, where $\|\mathbf{h}_{SC}\|^2 = \sum_{n=1}^{N} |h_{SC}^{(n)}|^2$. Hence, γ_{SC} is written as follows

$$\gamma_{SC} = \bar{\gamma}_S \ell_{SC} \left|\frac{\mathbf{h}_{SC}^H \mathbf{h}_{SC}}{\|\mathbf{h}_{SC}\|}\right|^2 = \bar{\gamma}_S \ell_{SC} \sum_{n=1}^{N} |h_{SC}^{(n)}|^2, \tag{3}$$

where the first equality is obtained by substituting the value of $\mathbf{w}_S$ and adopting the identity $\mathbf{h}_{SC}^H \mathbf{h}_{SC} = \|\mathbf{h}_{SC}\|^2$.

Hence, the SINR at the cluster head is formulated as

$$\Gamma_{SC} = \frac{\bar{\gamma}_S \ell_{SC} |h_{SC}|^2}{\sum_{i \in \mathcal{I}} \bar{\gamma}_{I_i} \ell_{I_iC} |h_{I_iC}|^2 + 1} = \frac{\gamma_{SC}}{I_C + 1}. \tag{4}$$

where $I_C = \sum_{i \in \mathcal{I}} \bar{\gamma}_{I_i} \ell_{I_iC} |h_{I_iC}|^2$ denotes the aggregated INR at C and $\bar{\gamma}_{I_i} = P_{I_i}/\sigma^2$. It is noted that γ_{SC} and I_C are independent RVs.

2) STATISTICALLY NON-IID TO MODEL CCI

Let us first explain the difference between the statistically independent and identically distributed (IID) and Non-IID assumptions. Specifically, IID RVs follow the same distribution with similar statistical characteristics [41, pp. 239]. Thus, it is apparent that the Non-IID assumption is more general than the IID assumption. On the other hands, the concept of Non-IID fading is essential and necessary in spatially distributed systems, and has been widely considered in the literatures [10], [42], [43], [44], [45], and [46].

Lemma 1: The PDF and CDF of the aggregated interference power at the cluster head, denoted as $f_{I_C}(x)$ and $F_{I_C}(x)$ for $x > 0$, respectively, can be obtained as follows

$$f_{I_C}(x) = \sum_{i=1}^{\varrho(\mathcal{A})} \sum_{j=1}^{\tau_i(\mathcal{A})} \chi_{i,j}(\mathcal{A}) \frac{a_{I_{\langle i \rangle}C}^j}{(j-1)!} x^{j-1} e^{-x a_{I_{\langle i \rangle}C}}, \quad (5)$$

$$F_{I_C}(x) = 1 - \sum_{i=1}^{\varrho(\mathcal{A})} \sum_{j=1}^{\tau_i(\mathcal{A})} \chi_{i,j}(\mathcal{A}) e^{-x a_{I_{\langle i \rangle}C}} \sum_{r=0}^{j-1} \frac{(x a_{I_{\langle i \rangle}C})^r}{r!}, \quad (6)$$

where $\mathcal{A} = \mathrm{diag}([\underbrace{a_{I_1C}, a_{I_1C}, \dots, a_{I_1C}}_{m_{I_1C} \text{ times}}, \dots, \underbrace{a_{I_LC}, \dots, a_{I_LC}}_{m_{I_LC} \text{ times}}])$, $a_{I_iC} \triangleq m_{I_iC}/(\bar{\gamma}_{I_i}\ell_{I_iC})$, $\varrho(\mathcal{A})$ is the number of distinct diagonal elements of $\mathcal{A}$, $a_{I_{(1)}C} > a_{I_{(2)}C} > \cdots > a_{I_{(\varrho(\mathcal{A}))}C}$ are the distinct diagonal elements in decreasing order, $\tau_i(\mathcal{A})$ is the multiplicity of $a_{I_{(i)}C}$, and $\chi_{i,j}(\mathcal{A})$ is the $(i,j)^{th}$ characteristic coefficient of $\mathcal{A}$.

Proof: The proof of Lemma 1 is given in Appendix A. ∎

3) PDF AND CDF OF γ_{SC}

From (3), the PDF of γ_{SC}, which is a Gamma variate with shape parameter Nm_{SC} and scale parameter $\frac{\bar{\gamma}_S \ell_{SC}}{m_{SC}}$, is obtained as

$$f_{\gamma_{SC}}(z) = \frac{(a_{SC})^{Nm_{SC}}}{\Gamma(Nm_{SC})} z^{Nm_{SC}-1} e^{-a_{SC}z}, \quad z > 0, \quad (7)$$

where $a_{SC} \triangleq \frac{m_{SC}}{\bar{\gamma}_S \ell_{SC}}$. In order to obtain the corresponding CDF, denoted as $F_{\gamma_{SC}}(z)$, we integrate (7) as follows

$$F_{\gamma_{SC}}(z) = \int_0^z f_{\gamma_{SC}}(z)dz = \frac{(a_{SC})^{Nm_{SC}}}{\Gamma(Nm_{SC})} \int_0^z z^{Nm_{SC}-1} e^{-a_{SC}z} dz, \quad (8)$$

where $\gamma(s,x)$ is the lower incomplete Gamma function [47, Eq. (8.350.1)]. Hence, we obtain

$$F_{\gamma_{SC}}(z) = \frac{1}{\Gamma(Nm_{SC})} \gamma(Nm_{SC}; a_{SC}z)$$

$$= 1 - e^{-a_{SC}z} \sum_{k=0}^{Nm_{SC}-1} \frac{(a_{SC}z)^k}{k!}, \quad z > 0. \quad (9)$$

Lemma 2: The CDF of Γ_{SC} can be expressed as

$$F_{\Gamma_{SC}}(\gamma) = 1 - e^{-a_{SC}\gamma} \sum_{m=0}^{Nm_{SC}-1} \frac{(a_{SC}\gamma)^m}{m!} \sum_{i=1}^{\varrho(\mathcal{A})} \sum_{j=1}^{\tau_i(\mathcal{A})} \chi_{i,j}(\mathcal{A})$$

$$\times \sum_{r=0}^{m} \binom{m}{r} \frac{(j)_r a_{I_{\langle i \rangle}C}^j}{(a_{SC}\gamma + a_{I_{\langle i \rangle}C})^{r+j}}, \quad \gamma > 0. \quad (10)$$

Proof: The CDF of γ_{SC}, defined as $F_{\Gamma_{SC}}(\gamma) = \mathrm{Pr}\left\{\frac{\gamma_{SC}}{I_C+1} < \gamma\right\}$, is written as

$$F_{\Gamma_{SC}}(\gamma) = 1 - e^{-a_{SC}\gamma} \sum_{m=0}^{Nm_{SC}-1} \frac{(a_{SC}\gamma)^m}{m!} \sum_{i=1}^{\varrho(\mathcal{A})} \sum_{j=1}^{\tau_i(\mathcal{A})} \chi_{i,j}(\mathcal{A})$$

$$\times \frac{a_{I_{\langle i \rangle}C}^j}{(j-1)!} \int_0^\infty x^{j-1} (x+1)^m e^{-(a_{SC}\gamma + a_{I_{\langle i \rangle}C})x} dx. \quad (11)$$

Using the definition of the Confluent hypergeometric function of the second kind, where $U(a,b,z) = \frac{1}{\Gamma(a)} \int_0^\infty e^{-zt} t^{a-1} (1+t)^{b-a-1} dt$, and with the fact that $U(a, a+b+1, z) = \sum_{r_k=0}^b \binom{b}{r_k} (a_{r_k})(z)^{-r_k-a}$, we obtain (10). This completes the proof of Lemma 2. ∎

B. C-TO-D_K COMMUNICATION

The received signal at D_k from C is written as

$$y_{CD_k} = \sqrt{\ell_{CD_k}} h_{CD_k} \sqrt{P_C} x_C + \sum_{i\in\mathcal{I}} \sqrt{\ell_{I_iD_k}} h_{I_iD_k} \sqrt{P_{I_i}} x_{I_i} + n_{D_k}, \quad (12)$$

where x_C is the normalized short packet signal dedicated to D_k and n_{D_k} is the complex AWGN with zero-mean and variance σ^2. $h_{I_iD_k}$ and $\ell_{I_iD_k}$ denote the complex small-scale fading and the path loss from interferer $i \in \mathcal{I}$ to D_k, respectively. Additionally, we assume that $|h_{I_iD_k}|^2$s for $k \in \mathcal{K}$, where $\mathcal{K} = 1, 2, \dots, K$, are Non-IID Gamma random variables, each with unit mean and corresponding shape parameter $m_{I_iD_k}$.

1) CHANNEL MODEL OF THE C-TO-D_K LINKS

The path loss from C to each D_k, denoted as ℓ_{CD_k} for $k \in \mathcal{K}$, are not necessarily identical and follow the model in (2). In other words, $\ell_{CD_k} = G_C + G_{D_k} - 22.7 - 26\log_{10}(f_c) - 36.7\log_{10}(\frac{d_{CD_k}}{d_0})$ [dB], where d_{CD_k} is the distance from C to D_k. Moreover, the SINR of the second phase at each user $k \in \mathcal{K}$, can be written as

$$\Gamma_{CD_k} = \frac{\bar{\gamma}_C \ell_{CD_k} |h_{CD_k}|^2}{\sum_{i\in\mathcal{I}} \bar{\gamma}_{I_i} \ell_{I_iD_k} |h_{I_iD_k}|^2 + 1} = \frac{\gamma_{CD_k}}{I_{D_k} + 1}. \quad (13)$$

where $\bar{\gamma}_C \triangleq P_C/\sigma^2$, $\gamma_{CD_k} \triangleq \bar{\gamma}_C \ell_{CD_k} |h_{CD_k}|^2$ and $I_{D_k} \triangleq \sum_{i\in\mathcal{I}} \bar{\gamma}_{I_i} \ell_{I_iD_k} |h_{I_iD_k}|^2$ are the instantaneous SNR and INR at each D_k, where $k \in \mathcal{K}$, respectively.

In this work, we primarily focus on the e2e error performance, as determined via the user with the lowest e2e SINR. In other words, the e2e SINR used for evaluated the error performance is formulated as

$$\Gamma_{sys} = \min_{k\in\mathcal{K}} \left\{ \min\{\Gamma_{SC}, \Gamma_{CD_k}\} \right\} = \min\left\{ \Gamma_{SC}, \min_{k\in\mathcal{K}}\{\Gamma_{CD_k}\} \right\}. \quad (14)$$

Lemma 3: The PDF and CDF of the aggregated interference power at D_k, denoted as $f_{I_{D_k}}(x)$ and $F_{I_{D_k}}(x)$ for $x > 0$,

respectively, are obtained as

$$f_{I_{D_k}}(x) = \sum_{i=1}^{\varrho(\mathcal{B}_k)} \sum_{j=1}^{\tau_i(\mathcal{B}_k)} \chi_{i,j}(\mathcal{B}_k) \frac{d_{I_{\langle i \rangle} D_k}^j}{(j-1)!} x^{j-1} e^{-x a_{I_{\langle i \rangle} D_k}}, \quad (15)$$

$$F_{I_{D_k}}(x) = 1 - \sum_{i=1}^{\varrho(\mathcal{B}_k)} \sum_{j=1}^{\tau_i(\mathcal{B}_k)} \chi_{i,j}(\mathcal{B}_k) e^{-\frac{x}{\mu_{I_{\langle i \rangle} D_k}}} \sum_{r=0}^{j-1} \frac{(x a_{I_{\langle i \rangle} D_k})^r}{r!}, \quad (16)$$

where $\mathcal{B}_k = \mathrm{diag}([\underbrace{a_{I_1 D_k}, \ldots, a_{I_1 D_k}}_{m_{I_1 D_k} \text{ times}}, \ldots, \underbrace{a_{I_L D_k}, \ldots, a_{I_L D_k}}_{m_{I_L D_k} \text{ times}}])$, $a_{I_i D_k} \triangleq \frac{m_{I_i D_k}}{\bar{\gamma}_{I_i} \ell_{I_i D_k}}$, $\varrho(\mathcal{B})$ denotes the number of distinct diagonal elements of $\mathcal{B}_k$, $a_{I_{\langle 1 \rangle} D_k} > a_{I_{\langle 2 \rangle} D_k} > \ldots > a_{I_{\langle \varrho(B_k) \rangle} D_k}$ are the distinct diagonal elements in decreasing order, $\tau_i(\mathcal{B}_k)$ is the multiplicity of $a_{I_{\langle i \rangle} D_k}$, and $\chi_{i,j}(\mathcal{B}_k)$ is the $(i,j)^{th}$ characteristic coefficient of $\mathcal{B}_k$.

Proof: The proof of Lemma 3 is given in Appendix B. ∎

Lemma 4: The CDF of Γ_{CD_k} can be expressed as

$$F_{\Gamma_{CD_k}}(\gamma) = 1 - e^{-a_{CD_k}\gamma} \sum_{m_k=0}^{m_{CD_k}-1} \frac{(a_{CD_k}\gamma)^{m_k}}{m_k!} \sum_{i_k=1}^{\varrho(\mathcal{B}_k)} \sum_{j_k=1}^{\tau_{i_k}(\mathcal{B}_k)}$$
$$\times \chi_{i_k,j_k}(\mathcal{B}_k) \sum_{r_k=0}^{m_k} \binom{m_k}{r_k}$$
$$\frac{d_{I_{\langle i_k \rangle} D_k}^{j_k} (j_k)_{r_k}}{(a_{CD_k}\gamma + a_{I_{\langle i_k \rangle} D_k})^{r_k+j_k}}, \quad \gamma > 0. \quad (17)$$

Proof: The CDF of γ_{CD_k}, defined as $F_{\Gamma_{CD_k}}(\gamma) = \Pr\left\{\frac{\gamma_{CD_k}}{I_{D_k}+1} < \gamma\right\}$, is written as

$$F_{\Gamma_{CD_k}}(\gamma) = 1 - e^{-a_{CD_k}\gamma} \sum_{m=0}^{Nm_{CD_k}-1} \frac{(a_{CD_k}\gamma)^m}{m!} \sum_{i=1}^{\varrho(\mathcal{B}_k)} \sum_{j=1}^{\tau_i(\mathcal{B}_k)}$$
$$\times \chi_{i,j}(\mathcal{B}_k) \frac{d_{I_{\langle i \rangle} C}^j}{(j-1)!} \int_0^\infty x^{j-1} (x+1)^m$$
$$\times e^{-(a_{CD_k}\gamma + a_{I_{\langle i \rangle} C})x} dx. \quad (18)$$

Using the Confluent hypergeometric function of the second kind, we obtain (17). This completes the proof of Lemma 4. ∎

Lemma 5: The CDF of Γ_{sys} can be written as

$$F_{\Gamma_{sys}}(\gamma) = 1 - \left[1 - F_{\Gamma_{SC}}(\gamma)\right] \prod_{k \in \mathcal{K}} \left[1 - F_{\Gamma_{CD_k}}(\gamma)\right], \quad \gamma > 0. \quad (19)$$

Proof: The SINR of the system is defined as the minimum SINR obtained by all users

$$\Gamma_{sys} = \min\left\{\Gamma_{SC}, \min_{k \in \mathcal{K}}\{\Gamma_{CD_k}\}\right\}, \quad (20)$$

where $\mathcal{K} \triangleq \{1, 2, \ldots, K\}$. Hence, the CDF of Γ_{sys}, defined as $F_{\Gamma_{sys}}(\gamma) = \Pr\{\Gamma_{sys} < \gamma\}$, is obtained as

$$F_{\Gamma_{sys}}(\gamma) = 1 - \Pr\left\{\Gamma_{SC} > \gamma, \bigcap_{k \in \mathcal{K}}\left[\Gamma_{CD_k} > \gamma\right]\right\}$$

$$= 1 - \Pr\{\Gamma_{SC} > \gamma\} \prod_{k \in \mathcal{K}} \Pr\left\{\Gamma_{CD_k} > \gamma\right\}, \quad (21)$$

which eventually yields (19). Note that Γ_{SC} is a function of h_{SC} and Γ_{CD_k} is a function of h_{CD_k} and $h_{I_i D_k}$, $k = 1, \ldots, K$ and $i = 1, \ldots, I$. Since h_{SC}, h_{CD_k}, and $h_{I_i D_k}$, for $k = 1, \ldots, K$ and $i = 1, \ldots, I$, are statistically independent, specifically non-IID, we conclude that Γ_{SC} and Γ_{CD_k}, $k = 1, \ldots, K$, are also statistically independent. Hence, the intersection (joint event) in the second equality in Eq. (21) occurs due to the property $\Pr\{E_1 E_2 \ldots\} = \prod_i \Pr\{E_i\}$ when E_i are mutually independent random events. This completes the proof of Lemma 5. ∎

III. PERFORMANCE ANALYSIS

To reduce the physical-layer transmission latency, URLLC often uses short-packet communications. The best design for short-packet communications is based on a detailed examination of the relationship between the BLER, maximum attainable rate (MAR), and blocklength. Given a blocklength of size M symbol/block, the MAR $R(M, \gamma, \epsilon)$ (in bits per channel usage (BPCU)) is expressed as [48]

$$R(M, \gamma, \epsilon) = C(\gamma) - \sqrt{\frac{V(\gamma)}{M}} Q^{-1}(\epsilon) + O\left(\frac{\log_2 M}{M}\right), \quad (22)$$

where $C(\gamma) = \log_2(1+\gamma)$ is the Shannon capacity, channel dispersion is denoted by $V(\gamma) = (1 - \frac{1}{(1+\gamma)^2})(\log_2 e)^2$, $Q^{-1}(x)$ is the Gaussian inverse Q-function, and $O\left(\frac{\log_2 M}{M}\right)$ is the remainder term which can be omitted when $M \geq 100$ [49]. Hence, the MAR is expressed as $R = \frac{F}{M}$, where F denotes the number of data bits for D_k.

We are using a half duplex relaying (HDR) decode-and-forward (DF) system. Each transmission block in HDR is divided into two equal-length time slots. The source transmits information to C during the first time slot, then C decodes and forwards the received signal to the destination in the second time slot. Loop interference at C is fully avoided in HDR.

A. BLER ANALYSIS

In this subsection, we study the average e2e BLER. Using DF relaying at C, errors can arise from two possible scenarios: (i) C identifies an error in the received data and successfully corrects it before relaying the data to the destination, (ii) C correctly relays the data, but the destination still detects errors in the received data.

Consider a transmission from S to C, let ϕ_{SC} and ϕ_{CD_k} represent the events related to the transmission process between S-to-C and C-to-D_k, respectively. The instantaneous BLER at C can be written as [50, Eq. (12)]

$$\Pr(\phi_{SC}) = \tilde{\epsilon}_{SC} \approx Q\left(\frac{C(\Gamma_{SC}) - R}{\sqrt{V(\Gamma_{SC})/M}}\right). \quad (23)$$

Provided that C successfully decodes and forwards that signal to D_k, the instantaneous BLER at D_k $\tilde{\epsilon}_{CD_k | C \text{succeeds}}$ is

written as

$$\Pr(\phi_{\mathrm{CD}_k}|\bar{\phi}_{\mathrm{SC}}) = \tilde{\epsilon}_{\mathrm{CD}_k|\mathrm{SC\ succeeds}} \approx Q\left(\frac{C(\Gamma_{\mathrm{CD}_k}) - R}{\sqrt{V(\Gamma_{\mathrm{CD}_k})/M}}\right),$$

(24)

where $\Pr(\phi_{\mathrm{CD}_k}|\phi_{\mathrm{SC}})$ is the conditional probability of ϕ_{CD_k} given ϕ_{SC}. It is noted that D_k's signal i.e., x_{CD_k} can be received successfully at D_k when the signal at C i.e., x_{SC} is decoded correctly. Meanwhile, due to the high level of interference, x_{CD_k} is decoded incorrectly when x_{SC} is erroneously decoded. As a result, it is assumed that $\Pr(\phi_{\mathrm{CD}_k}|\phi_{\mathrm{SC}})$ is equal to one [48]. Finally, since $\Pr(\phi_{\mathrm{CD}_k}) = \Pr(\phi_{\mathrm{CD}_k}|\phi_{\mathrm{SC}})\Pr(\phi_{\mathrm{SC}}) + \Pr(\phi_{\mathrm{CD}_k}|\bar{\phi}_{\mathrm{SC}})\Pr(\bar{\phi}_{\mathrm{SC}})$, the e2e instantaneous BLER ($\tilde{\epsilon}_{e2e}$), where $\tilde{\epsilon}_{\mathrm{CD}|\mathrm{SC\ succeeds}} = \tilde{\epsilon}_{\mathrm{CD}_k}$ is written as

$$\tilde{\epsilon}_{e2e} = Q\left(\frac{C(\Gamma_{e2e}) - R}{\sqrt{V(\Gamma_{e2e})/M}}\right)$$

(25)

$$\approx 1 \times \tilde{\epsilon}_{\mathrm{SC}} + \tilde{\epsilon}_{\mathrm{CD}_k} \times (1 - \tilde{\epsilon}_{\mathrm{SC}}) = \tilde{\epsilon}_{\mathrm{SC}} + (1 - \tilde{\epsilon}_{\mathrm{SC}})\tilde{\epsilon}_{\mathrm{CD}_k}$$

.

(26)

B. AVERAGE BLER OF EACH TRANSMISSION (EACH HOP)
For SPC, the average BLER can be approximated as [51]

$$\bar{\epsilon}_{\chi} \approx \mathbb{E}\left[Q\left(\frac{C(\gamma_{\chi}) - R}{\sqrt{V(\gamma_{\chi})/M}}\right)\right],$$

(27)

where $\chi \in \{\mathrm{SC}, \mathrm{CD}, e2e\}$, $\mathbb{E}[\cdot]$ is the expectation operator, and $Q(\cdot)$ denotes the Gaussian Q-function. It is intractable to derive analytically closed-form expression of (27) due to the Gaussian Q-function. To approximate the expression, the estimated linear form of $Q\left(\frac{C(\gamma_{\chi}) - R}{\sqrt{V(\gamma_{\chi})/M}}\right)$ is invoked as [52]

$$Z(\gamma_{\chi}) = \frac{1}{2} - \tau\left(\gamma_{\chi} - \omega\right), \quad \psi < \gamma_{\chi} < \eta,$$

(28)

where $\tau \triangleq \frac{\sqrt{M}}{\sqrt{2\pi(4^R - 1)}}$, $\eta \triangleq \omega + \frac{1}{2\tau}$, and $\omega \triangleq 2^R - 1$. Hence, substituting (28) in (27), the average BLER is tightly approximated as

$$\bar{\epsilon}_{\chi} \approx \int_0^{\infty} Z(\gamma_{\chi}) f_{\gamma_{\chi}}(x) dx = \tau \int_{\psi}^{\eta} F_{\gamma_{\chi}}(x) dx,$$

(29)

where $F_{\gamma_{\chi}}(x)$ is the CDF of γ_{χ}, and second equality is due to the partial integration method.

1) AVERAGE BLER OF THE FIRST HOPE (S-TO-C)
We derive the closed-form expression for the average BLER at C, i.e., $\bar{\epsilon}_{\mathrm{SC}}$, in Theorem 1.

Theorem 1: Given a number of transmit data bits, the average BLER $\bar{\epsilon}_{\mathrm{SC}}$ at C can be written in closed-form expression as

$$\bar{\epsilon}_{\mathrm{SC}} = 1 - \tau \sum_{m=0}^{Nm_{\mathrm{SC}}-1} \sum_{i=1}^{\varrho(\boldsymbol{\mathcal{A}})} \sum_{j=1}^{\tau_i(\boldsymbol{\mathcal{A}})} \chi_{i,j}(\boldsymbol{\mathcal{A}})$$

$$\times \sum_{r=0}^{m} \frac{d_{I_{(i)}\mathrm{C}}^j a_{\mathrm{SC}}^{r-1}}{r!}\left[\Theta(r, \psi) - \Theta(r, \eta)\right],$$

(30)

where $\Theta(r, \psi) \triangleq \psi^r e^{-a_{\mathrm{SC}}\psi} U\left(j, r+j, a_{I_{(i)}\mathrm{C}} + a_{\mathrm{SC}}\psi\right)$.

Proof: The proof of Theorem 1 is given in Appendix C. ∎

Corollary 1: Let us assume an interference-free scenario, where the number of interferers is zero, then we can write the average BLER without interference at C i.e., $\hat{\epsilon}_{\mathrm{SC}}$, as

$$\hat{\epsilon}_{\mathrm{SC}} = 1 - \tau \sum_{k=0}^{Nm_{\mathrm{SC}}-1} \sum_{q=0}^{k} \frac{(a_{\mathrm{SC}})^{q-1}}{q!}(e^{-a_{\mathrm{SC}}\psi}\psi^q - e^{-a_{\mathrm{SC}}\eta}\eta^q).$$

(31)

Proof: The proof of Corollary 1 is obtained following the same approach in Appendix C by taking zero interferers. ∎

2) AVERAGE BLER OF THE SECOND HOPE FOR EACH INDIVIDUAL USER K (C-TO-D_K)
In the C-to-D_k (Cluster head to k^{th} user) links, the small scale fading follows IID Nakagami-m distributions and h_{CD_k} follows IID Gamma distributions. We derive the closed-form expression of the average BLER at D_k, i.e., $\bar{\epsilon}_{\mathrm{CD}_k}$, in Theorem 2.

Theorem 2: The average BLER at D_k, denoted as $\bar{\epsilon}_{\mathrm{CD}_k}$, is obtained as

$$\bar{\epsilon}_{\mathrm{CD}_k} = 1 - \tau \sum_{m_k=0}^{Nm_{\mathrm{CD}_k}-1} \sum_{i_k=1}^{\varrho(\boldsymbol{\mathcal{B}}_k)} \sum_{j_k=1}^{\tau_{ik}(\boldsymbol{\mathcal{B}}_k)} \chi_{i_k,j_k}(\boldsymbol{\mathcal{B}}_k)$$

$$\times \sum_{r_k=0}^{m} \frac{d_{I_{(i_k)}\mathrm{C}}^{j_k}(a_{\mathrm{CD}_k})^{r_k-1}}{r_k!}\left[\Theta_1(r_k, \psi) - \Theta_1(r_k, \eta)\right],$$

(32)

where $\Theta_1(r_k, \psi) \triangleq \psi^{r_k} e^{-a_{\mathrm{CD}_k}\psi} U(j_k, r_k + j_k, a_{I_{(i_k)}\mathrm{C}} + a_{\mathrm{CD}_k}\psi)$.

Proof: The proof of Theorem 2 is given in Appendix D. ∎

Corollary 2: Let us assume an interference-free scenario, where the number of interferers is zero, then we can write the average BLER at D_k i.e., $\hat{\epsilon}_{\mathrm{CD}_k}$, as

$$\hat{\epsilon}_{\mathrm{CD}_k} = 1 - \tau \sum_{k=0}^{Nm_{\mathrm{CD}_k}-1} \sum_{q=0}^{k} (a_{\mathrm{CD}_k})^{q-1} \frac{e^{-a_{\mathrm{CD}_k}\psi}\psi^q - e^{-a_{\mathrm{CD}_k}\eta}\eta^q}{q!}.$$

(33)

Proof: The proof of Corollary 2 is obtained following the same approach in Appendix D by taking zero interferers. ∎

3) AVERAGE e2e BLER
The average e2e BLER of the system is given in Theorem 3.

Theorem 3: The average e2e BLER $\bar{\epsilon}_{e2e}$ for the system can be written as

$$\bar{\epsilon}_{e2e} \approx 1 - \sum_{m=0}^{Nm_{\mathrm{SC}}-1} \sum_{m_1=0}^{m_{\mathrm{CD}_1}-1} \cdots \sum_{m_k=0}^{m_{\mathrm{CD}_k}-1} \sum_{i=1}^{\varrho(\boldsymbol{\mathcal{A}})} \sum_{i_1=1}^{\varrho(\boldsymbol{\mathcal{B}}_1)} \cdots \sum_{i_k=1}^{\varrho(\boldsymbol{\mathcal{B}}_k)}$$

$$\times \sum_{j=1}^{\tau_i(\boldsymbol{\mathcal{A}})} \chi_{i,j}(\boldsymbol{\mathcal{A}}) \sum_{j_1=1}^{\tau_1(\boldsymbol{\mathcal{B}}_1)} \cdots \sum_{j_k=1}^{\tau_k(\boldsymbol{\mathcal{B}}_k)} \left\{\prod_{k=1}^{K} \chi_{i_k,j_k}(\boldsymbol{\mathcal{B}}_k)\right\} \Lambda_1,$$

(34)

where Λ_1, Λ_2, χ_{k,l_k}, Λ_3, and χ_l are given in (35), (36), (37), (38), and (39), as shown at the bottom of the next page, respectively. Herein, $m_\Sigma = \sum_{k=1}^{K} m_k$ and $a_\Sigma = a_{SC} + \sum_{k=1}^{K} a_{CD_k}$.

Proof: The proof of Theorem 3 is given in Appendix E. ∎

4) APPROXIMATION FOR $\bar{\epsilon}_{E2E}$

The $\bar{\epsilon}_{e2e}$ can be approximated as follow

$$\bar{\epsilon}_{e2e} \approx \bar{\epsilon}_{SC} + (1 - \bar{\epsilon}_{SC})\left(1 - \prod_{k=1}^{K}(1 - \bar{\epsilon}_{CD_k})\right)$$

$$= 1 - (1 - \bar{\epsilon}_{SC})\prod_{k=1}^{K}(1 - \bar{\epsilon}_{CD_k}). \quad (40)$$

C. ASYMPTOTIC, ERROR FLOOR, AND DIVERSITY GAIN ANALYSES

1) AVERAGE BLER OF THE FIRST PHASE

In the high transmit SINR regime, where $\bar{\gamma}_S$ is relatively large, we have $F_{\gamma_{SC}(\gamma)} \to (a_{SC}\gamma)^{Nm_{SC}}/(Nm_{SC})!$, for $\gamma > 0$. Hence, the CDF of the SINR at C can be simplified as

$$F_{\Gamma_{SC}}(\gamma) \to \alpha_{SC}\frac{(a_{SC}\gamma)^{Nm_{SC}}}{(Nm_{SC}+1)!}, \quad \gamma > 0, \quad (41)$$

where $\alpha_{SC} \triangleq \sum_{i=1}^{\varrho(\mathcal{A})} \sum_{j=1}^{\tau_i(\mathcal{A})} \chi_{i,j}(\mathcal{A})a_{I_{(i)}C}^j U(j, Nm_{SC}+j+1, a_{I_{(i_k)}}C)$.

As a result, the average BLER at C in the high SINR regime, denoted as $\bar{\epsilon}_{SC}^{\infty}(P_S)$, is obtained as

$$\bar{\epsilon}_{SC}^{\infty}(P_S) = \frac{\tau\alpha_{SC}(a_{SC})^{Nm_{SC}}}{(Nm_{SC}+1)!}(\eta^{Nm_{SC}+1} - \psi^{Nm_{SC}+1}). \quad (42)$$

2) AVERAGE BLER OF THE SECOND PHASE

In the high transmit SINR regime, where $\bar{\gamma}_{CD_k}$ is relatively large, we have $F_{\gamma_{CD_k}(\gamma)} \to (a_{CD_k}\gamma)^{m_{CD_k}}/(Nm_{CD_k})!$, for $\gamma > 0$, Hence, the CDF of the SINR at the cluster head can be simplified as

$$F_{\Gamma_{CD_k}}(\gamma) \to \beta_{CD_k}\frac{(a_{CD_k}\gamma)^{m_{CD_k}}}{(m_{CD_k}+1)!}, \quad \gamma > 0, \quad (43)$$

where $\beta_{CD_k} \triangleq \sum_{i=1}^{\varrho(\mathcal{B})} \sum_{j=1}^{\tau_i(\mathcal{B})} \chi_{i,j}(\mathcal{B})\frac{(j+m_{CD_k}-1)!}{(j-1)!}(a_{I_{(i)}D_k})^{-m_{CD_k}}$.

Hence, the average BLER at D_k in the high SINR regime, denoted as $\bar{\epsilon}_{CD_k}^{\infty}(P_C)$, is obtained as

$$\bar{\epsilon}_{CD_k}^{\infty}(P_C) = \frac{\tau\beta_{CD_k}}{(m_{CD_k}+1)!}$$
$$\times (a_{CD_k})^{m_{CD_k}}(\eta^{m_{CD_k}+1} - \psi^{m_{CD_k}+1}). \quad (44)$$

3) AVERAGE e2e BLER

Accordingly, the average BLER of the system in the high SINR regime, denoted as $\bar{\epsilon}_{e2e}^{\infty}$, is given by

$$\bar{\epsilon}_{e2e}^{\infty} = 1 - (1 - \bar{\epsilon}_{SC}^{\infty}(P_S))\prod_{k=1}^{K}(1 - \bar{\epsilon}_{CD_k}^{\infty}(P_C)) \quad (45)$$

$$= 1 - \left\{1 - \frac{\tau\alpha_{SC}}{(Nm_{SC}+1)!}(a_{SC})^{Nm_{SC}}\right.$$
$$\times (\eta^{Nm_{SC}+1} - \psi^{Nm_{SC}+1})\left\}\prod_{k=1}^{K}\left\{1\right.$$
$$\left.- \frac{\tau\beta_{CD_k}}{(m_{CD_k}+1)!}(a_{CD_k})^{m_{CD_k}}(\eta^{m_{CD_k}+1} - \psi^{m_{CD_k}+1})\right\}. \quad (46)$$

4) ERROR FLOOR ANALYSIS

In this subsection, we determine the error floor of the average e2e BLER. The error floor analysis provides useful system design guidelines. There is a region in which performance flattens is called the error floor region. We consider the error floor to be the lower limit achieved by the average e2e BLER when either P_S or P_C tends towards infinity [53], while the counterpart (P_C or P_S, respectively) remains within a low-to-middle transmission regime. The error floor of the average e2e BLER when only P_S or $P_C \to \infty$ is obtained as

$$\bar{\epsilon}_{e2e}^{\infty} \to \begin{cases} \prod_{k=1}^{K} \bar{\epsilon}_{CD_k}, & P_S \to \infty, \\ \bar{\epsilon}_{SC}, & P_C \to \infty. \end{cases} \quad (47)$$

From (47), we draw the insights that by increasing the transmission power of the BS or CH alone does not lead to further improvement in the BLER. This is because the end-to-end instantaneous BLER is determined by the minimum BLER equation, which implies that the system's performance is limited by the weakest link, such as the user with the worst channel conditions.

5) DIVERSITY GAIN ANALYSIS

Unlike the error floor analysis, the diversity gain is calculated at $P_S = P_C \triangleq P \to \infty$. In other words, the diversity gain of the considering URLLC system can be defined as [54]

$$D_{e2e} = -\lim_{P}\frac{\log(\bar{\epsilon}_{e2e}(P))}{\log(P)} = -\lim_{P}\frac{\log(\bar{\epsilon}_{e2e}^{\infty}(P))}{\log(P)}$$
$$= \min\{Nm_{SC}, m_{CD_1}, \ldots, m_{CD_K}\}, \quad (48)$$

where $\bar{\epsilon}_{e2e}^{\infty}(P) = 1 - (1 - \bar{\epsilon}_{SC}^{\infty}(P))\prod_{k=1}^{K}(1 - \bar{\epsilon}_{CD_k}^{\infty}(P))$ is merely the average e2e BLER expressed as a function of P.

Remark 1: Specifically, when setting the transmit powers of the source (P_S) and the cluster head (P_C) equal ($P_S = P_C = P$), the average end-to-end BLER ($\bar{\epsilon}_{e2e}^{\infty}(P)$) can be modeled as a summation of terms $C_iP^{-b_i}$, where C_i and b_i are positive constants with respect to P. In this scenario, the diversity order of the system (D_{e2e}) is determined by the minimum value of b_i across all terms, denoted as $\min_i\{b_i\}$. As a result, when both BS and CH transmission powers are increased by r [dB] in the high transmit SINR regime, the average end-to-end BLER decreases by $r\min_i\{b_i\}$ [dB]. This relationship highlights the beneficial impact of increasing transmit powers on improving system performance, particularly in environments characterized by high SNR levels.

With such a communication network, locations of the S, C, and users are treated as random variables (RVs), which make

the performance analysis intractable. [55]. Thus, relying on the data driven techniques to develop a CNN model based on transfer learning that can be trained to predict the average overall BLER using pre-collected CSI and associated outage probability data. The multivariate confluent hypergeometric function having complex BLER expression renders the calculation of the system average overall BLER specified in (40) unfeasible for real-time application, as distances between nodes in the proposed system are now also RVs. We approach the challenge of determining the system average overall BLER as a non-linear regression problem in order to get over this hurdle. Specifically, we produce a data set that thoroughly describes the considered system. The created CNN can predict the average overall BLER accurately under different system settings.

To deal with unpredictable performance at ground users, S and C can leverage a CNN to predict the user performance and adjust parameters to satisfy the fairness and quality of service at each user. Motivated by the aforementioned performance analysis, this paper offers a methodology for assessing system performance and then proceeds to evaluate performance prediction through transfer learning [56].

IV. TRANSFER LEARNING-BASED PERFORMANCE PREDICTION

In our paper, we leverage ML for real-time performance prediction. We specifically use transfer learning, employing a pre-trained model initially trained on a larger source domain, to handle a target task on a limited dataset. This method applies the source model's knowledge to smaller datasets, facilitating efficient real-time predictions.

A. UNIVERSAL APPROXIMATION FUNCTION FOR PERFORMANCE PREDICTION

The average e2e BLER can be mathematically represented as a function $\Upsilon : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ represent the input and output spaces, respectively. Our goal is to develop

a transfer learning-based universal approximation function $\widetilde{\Upsilon} : \mathcal{X} \to \hat{\mathcal{Y}}$, where $\hat{\mathcal{Y}}$ is an accurate approximation of $\mathcal{Y}$ based on a limited dataset that does not fully capture the input-output relationship in Υ. Hence, the developed ML models utilize regression analysis to accurately characterize the continuous function Υ. These models are designed to exhibit high generalization capabilities, ensuring accurate predictions even with new inputs.

Here, regression analysis is a statistical technique used to estimate the relationship between a dependent variable y and an input feature vector, denoted as $\mathbf{x} = [x_1, \ldots, x_m]^\mathsf{T}$, where m denotes the number of features [57]. The goal of regression analysis is to determine an objective predictive function Υ that maps the feature space $\mathcal{X}$ to the output space $\mathcal{Y}$, i.e., $\Upsilon : \mathcal{X} \to \mathcal{Y}$, such that

$$y = \Upsilon(\mathbf{x}; \boldsymbol{\theta}), \tag{49}$$

where $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, and $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_m]^\mathsf{T}$ denotes the unknown parameters with θ_0 being the bias that allows for an offset in the prediction model. In this paper, we focus on non-linear Regression (NLR) models since the average e2e BLER is a non-linear function of input system parameters $\mathbf{x}$. In addition, using NLR models provide more flexible relationships between y and $\mathbf{x}$ using non-linear functions of the parameters and/or the independent variables. We denote the average e2e BLER in (51), as shown at the bottom of the next page, as $y = \Upsilon(\mathbf{x})$ with

$$\mathbf{x} = [\mathbf{p}_\mathsf{S}^\mathsf{T}, \mathbf{p}_{\mathsf{D}_1}^\mathsf{T}, \ldots, \mathbf{p}_{\mathsf{D}_K}^\mathsf{T}, \mathbf{p}_{I_1}^\mathsf{T}, \ldots, \mathbf{p}_{I_I}^\mathsf{T}, \mathbf{p}_\mathsf{C}^\mathsf{T}, F, M, m_\mathsf{SC},$$
$$m_{\mathsf{CD}_1}, \ldots, m_{\mathsf{CD}_K}, \ell_\mathsf{SC}, \ell_{\mathsf{CD}_1}, \ldots, \ell_{\mathsf{CD}_K}, N]^\mathsf{T}, \tag{50}$$

where $\mathbf{p}_\mathsf{X} \triangleq [x_\mathsf{X}, y_\mathsf{X}, z_\mathsf{X}]^\mathsf{T}$ denotes the 3D Cartesian coordinates of node X. It is noted that the total number of input parameters $L = 11 + 5K + 3I$. As a result, predicting the average e2e BLER of a URLLC system is a complex task that involves modeling the relationship between these factors and the average e2e BLER.

$$\Lambda_1 = \sum_{r=0}^{m} \binom{m}{r} \frac{(j+r-1)!}{(j-1)!} \frac{a_{I_{\langle i \rangle}\mathsf{C}}^j a_{\mathsf{SC}}^m}{m!} \sum_{r_1=0}^{m_1} \cdots \sum_{r_k=0}^{m_k} (\Lambda_2 + \Lambda_3) \prod_{k=1}^{K} \binom{m_k}{r_k} \frac{(j_k + r_k - 1)!}{(j_k - 1)!} \frac{a_{I_{\langle i_k \rangle}\mathsf{D}_k}^{j_k} a_{\mathsf{CD}_k}^{m_k}}{m_k!}, \tag{35}$$

$$\Lambda_2 = \sum_{k=1}^{K} \sum_{l_k=1}^{r_k+j_k} \frac{\chi_{k,l_k}}{a_{I_{\langle i_k \rangle}\mathsf{D}_k}^{l_k}} \left\{ \Lambda\left(\eta, m + m_\Sigma, a_\Sigma, \frac{a_{\mathsf{CD}_k}}{a_{I_{\langle i_k \rangle}\mathsf{D}_k}} \eta\right) - \Lambda\left(\psi, m + m_\Sigma, a_\Sigma, \frac{\psi a_{\mathsf{CD}_k}}{a_{I_{\langle i_k \rangle}\mathsf{D}_k}}\right) \right\}, \tag{36}$$

$$\chi_{k,l_k} = \frac{\left(\frac{a_{\mathsf{CD}_k}}{a_{I_{\langle i_k \rangle}\mathsf{D}_k}}\right)^{r_k+j_k-l_k}}{(r_k + j_k - l_k)!} \left[\frac{d^{r_k+j_k-l_k}}{ds^{r_k+j_k-l_k}} \left(\frac{a_{\mathsf{SC}}}{a_{I_{\langle i \rangle}\mathsf{C}}} \gamma + 1\right)^{r_k+j_k-l_k} \right]\Bigg|_{s \to -a_{\mathsf{CD}_k}/a_{I_{\langle i_k \rangle}\mathsf{D}_k}}, \tag{37}$$

$$\Lambda_3 = \sum_{l=1}^{r+j} \frac{\chi_l}{a_{I_{\langle i \rangle}\mathsf{C}}^l} \left\{ \Lambda\left(\eta, m + m_\Sigma, a_\Sigma, \frac{\eta a_{\mathsf{SC}}}{a_{I_{\langle i \rangle}\mathsf{C}}}\right) - \Lambda\left(\psi, m + m_\Sigma, a_\Sigma, \frac{\psi a_{\mathsf{SC}}}{a_{I_{\langle i \rangle}\mathsf{C}}}\right) \right\}, \tag{38}$$

$$\chi_l = \frac{\left(\frac{a_{\mathsf{SC}}}{a_{I_{\langle i \rangle}\mathsf{C}}}\right)^{r+j-l}}{(r+j-l)!} \left[\frac{d^{r+j-l}}{ds^{r+j-l}} \left(\prod_{k=1}^{K} (a_{\mathsf{CD}_k} \gamma + a_{I_{\langle i_k \rangle}\mathsf{D}_k})\right)^{r_k+j_k} \right]\Bigg|_{s \to -a_{\mathsf{SC}}/a_{I_{\langle i \rangle}\mathsf{C}}}. \tag{39}$$

B. DOMAIN ADAPTATION-BASED TRANSFER LEARNING SOLUTION

In domain adaptation-based transfer learning, a *domain* $\mathcal{D} = \{\mathcal{X}, p(\mathbf{x})\}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $p(\mathbf{x})$. A *task* $\mathcal{T} = \{\mathcal{Y}, f\}$ consists of an output space $\mathcal{Y}$ and an objective predictive function f which is not observed but can be learned from the training data. Here, the data consist of pairs $(\mathbf{x}, y)$, where $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$.

In this paper, we consider a pre-trained (source) model with one source domain $\mathcal{D}_s$ and one source target $\mathcal{T}_s$, and a fine-tuned (target) model with one target domain $\mathcal{D}_t$ and one target task $\mathcal{T}_s$. We can define $\mathcal{D}_s$, $\mathcal{T}_s$, $\mathcal{D}_t$, and $\mathcal{T}_t$ as [33]

$$\mathcal{D}_s \triangleq \{\mathcal{X}_s, p(\S_s)\}, \S_s \in \mathcal{X}_s, \mathcal{T}_s \triangleq \{\mathcal{Y}_s, f_s\},$$
$$f_s : \mathcal{X}_s \to \mathcal{Y}_s, y_s = f_s(\S_s) \in \mathcal{Y}_s, \quad (52)$$
$$\mathcal{D}_t \triangleq \{\mathcal{X}_t, p(\S_t)\}, \S_t \in \mathcal{X}_t, \quad \mathcal{T}_t \triangleq \{\mathcal{Y}_t, f_t\},$$
$$f_t : \mathcal{X}_t \to \mathcal{Y}_t, y_t = f_t(\S_t) \in \mathcal{Y}_t, \quad (53)$$

respectively, where $\mathcal{X}_s$, $\mathcal{Y}_s$, $\mathcal{X}_t$, and $\mathcal{Y}_t$ denote the feature space of source domain, output space of source domain, feature space of target domain, and output space of target domain, respectively. Similarly, $\mathbf{x}_s$ and $\mathbf{x}_t$ denote the input features vectors of the source and target domains, respectively. It is noted that the source model is designed with a complex architecture and high computational costs to accurately capture the average e2e BLER represented by the function Υ in (51). In this context, the source task is characterized as $f_s = \Upsilon$, and the output y_s is obtained by applying Υ to the input $\mathbf{x}_s$, where $\mathbf{x}_s$ corresponds to the system features $\mathbf{x}$ in (50). However, training the source model requires a substantial dataset and extensive training time. To overcome this challenge, we aim to develop a target model with less complex architecture by fine-tuning the source model with $\mathcal{T}_t = \mathcal{T}_s$. By opting for a fine-tuned target model on a small dataset, benefits such as faster training, reduced data requirements, and comparable accuracy to the source model can be achieved.

In practice, exact knowledge of $p(\mathbf{x}_s)$ and $p(\mathbf{x}_t)$ is unknown; instead, these probabilities are empirically estimated using the available dataset, which can vary depending on the dataset size. When the dataset size is sufficiently large, such as in the source model, the empirical probabilities $p(\mathbf{x}_s)$ can effectively represent the true probabilities. However, in cases where the dataset is small, the resulting empirical marginal probabilities in the target model may differ due

to the limited representation of the true probabilities, i.e., $\mathbf{p}(\mathbf{x}_t) \neq \sqrt{}(\mathbf{x}_s)$, where $\mathbf{x}_t, \S_s \in \mathcal{X}$. Considering the constraint of limited data available in the target domain, our objective is to design a target model that can be effectively trained and yield comparable accuracy to the source model.

Remark 2: Since $\mathcal{T}_s = \mathcal{T}_t$, $\mathcal{X}_s = \mathcal{X}_t$ and $p(\mathbf{x}_s) \neq p(\S_t)$, our problem is identified as a closed set domain adaptation in transductive transfer learning [33]. Specifically, some part of the dataset belong only to the source domain or to the target domain, while others belong to both datasets.

C. DATA

- The number of samples of our source dataset, denoted by n_{src}, is 1×10^5, while the number of samples of our target dataset, denoted by n_T, is 1×10^2.
- The target dataset contains a relatively smaller number of samples (10^2), as it represents a specific domain or task for which labeled data may be limited or expensive to obtain. In such scenarios, transfer learning offers a compelling advantage by leveraging knowledge acquired from the source domain to enhance learning and inference in the target domain. By fine-tuning the pretrained model with the smaller target dataset, we effectively transfer the learned representations and knowledge from the source domain to the target domain, thereby mitigating the challenges posed by limited labeled data availability.
- The choice of the number of samples in our source and target datasets is based on several factors. The source dataset comprises a large number of samples (10^5) to ensure a comprehensive coverage of diverse data instances, thus facilitating robust learning of underlying patterns and features. This abundance of data allows our model to capture a wide range of variations and complexities present in the source domain, which enhances its ability to effectively generalize and perform across different scenarios.

1) SOURCE DATASET

The source dataset is the dataset that is used to pre-train the original model. This dataset is typically large and diverse, and is used to learn a general feature representation of the data. All the features of our source dataset are represented in (50). In our case, for K users, the source dataset is in the form of a feature matrix and target vector.

$$y = \Upsilon\left([\mathbf{p}_S^\mathsf{T}, \mathbf{p}_{D_1}^\mathsf{T}, \ldots, \mathbf{p}_{D_K}^\mathsf{T}, \mathbf{p}_{I_1}^\mathsf{T}, \ldots, \mathbf{p}_{I_I}^\mathsf{T}, \mathbf{p}_C^\mathsf{T}, F, M, m_{\text{SC}}, m_{\text{CD}_1}, \ldots, m_{\text{CD}_K}, \ell_{\text{SC}}, \ell_{\text{CD}_1}, \ldots, \ell_{\text{CD}_K}, N]\right)$$

$$= 1 - \prod_{k=1}^{K} \left\{ 1 - \tau \sum_{m_k=0}^{Nm_{\text{CD}_k}-1} \sum_{i_k=1}^{\varrho(\boldsymbol{\mathcal{B}}_k)} \sum_{j_k=1}^{\tau_{ik}(\boldsymbol{\mathcal{B}}_k)} \sum_{r_k=0}^{m} \frac{\chi_{i_k,j_k}(\boldsymbol{\mathcal{B}}_k)}{r_k!} \left(\frac{m_{I_{\langle i_k\rangle}C}}{\bar{\gamma}_{I_{\langle i_k\rangle}}\ell_{I_{\langle i_k\rangle}C}}\right)^{j_k} \left(\frac{m_{\text{CD}_k}}{\bar{\gamma}_C \ell_{\text{CD}_k}}\right)^{r_k-1} \left[\Theta_1(r_k, \psi) - \Theta_1(r, \eta)\right] \right\}$$

$$\times \left\{ 1 - \tau \sum_{m=0}^{Nm_{\text{SC}}-1} \sum_{i=1}^{\varrho(\boldsymbol{\mathcal{A}})} \sum_{j=1}^{\tau_i(\boldsymbol{\mathcal{A}})} \sum_{r=0}^{m} \frac{\chi_{i,j}(\boldsymbol{\mathcal{A}})}{r!} \left(\frac{m_{I_{\langle i\rangle}C}}{\bar{\gamma}_{I_{\langle i\rangle}}\ell_{I_{\langle i\rangle}C}}\right)^{r-1} \left(\frac{m_{\text{SC}}}{\bar{\gamma}_S \ell_{\text{SC}}}\right)^{r-1} \left[\Theta(r, \psi) - \Theta(r, \eta)\right] \right\}, \quad \tau = \frac{\sqrt{M}}{\sqrt{2\pi(2^{2R}-1)}},$$
$$(51)$$

2) FEATURE MATRIX

In our paper, we consider the following feature matrix

$$\mathbf{X} = [\mathbf{x}_1^{\mathsf{T}}; \cdots ; \mathbf{x}_K^{\mathsf{T}}]^{\mathsf{T}}, \quad \mathbf{x}_k \triangleq [\mathbf{p}_{\mathrm{S}}^{\mathsf{T}}, \mathbf{p}_{\mathrm{D}_k}^{\mathsf{T}}, \mathbf{p}_{I_1}^{\mathsf{T}}, \dots, \mathbf{p}_{I_I}^{\mathsf{T}},$$
$$\mathbf{p}_{\mathrm{C}}^{\mathsf{T}}, F, M, m_{\mathrm{SC}}, m_{\mathrm{CD}_k}, \ell_{\mathrm{SC}}, \ell_{\mathrm{CD}_k}, N]^{\mathsf{T}}. \quad (54)$$

In addition, the feature set and target set of the source dataset are written as $\mathcal{X}_{\mathrm{s}} = \{\boldsymbol{\mathcal{X}}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(N)}\}$ and $\mathcal{Y}_{\mathrm{s}} = \{\bar{\epsilon}_{\mathrm{e2e}}^{(1)}, \bar{\epsilon}_{\mathrm{e2e}}^{(2)}, \dots, \bar{\epsilon}_{\mathrm{e2e}}^{(N)}\}$, respectively, where $\bar{\epsilon}_{\mathrm{e2e}}^{(n)} = \Upsilon(\mathbf{X}^{(n)})$ and n denotes the instance number. Hence, the structure of the dataset is $[\mathcal{X}_{\mathrm{s}}, \mathcal{Y}_{\mathrm{s}}]$, in which each data point is $[\mathbf{X}^{(n)}, \bar{\epsilon}_{\mathrm{e2e}}^{(n)}]$, where $\mathbf{X}^{(n)} \in \mathcal{X}_{\mathrm{s}}, \bar{\epsilon}_{\mathrm{e2e}}^{(n)} \in \mathcal{Y}_{\mathrm{s}}$, and N is the number of samples.

3) TARGET DATASET

The target dataset is the dataset for which the source model is adapted to perform a new task. This dataset is much smaller and more specific than the source dataset. The feature set and target set of the whole data are written as $\mathcal{X}_{\mathrm{t}} = [X_1', X_2', \dots, X_M']^{\mathsf{T}}$ and $\mathcal{Y}_{\mathrm{t}} = [\bar{\epsilon}_{\mathrm{e2e},1}, \bar{\epsilon}_{\mathrm{e2e},2}, \dots, \bar{\epsilon}_{\mathrm{e2e},M}]^{\mathsf{T}}$, respectively. Hence, the structure of the target dataset is $[\mathcal{X}_{\mathrm{t}}, \mathcal{Y}_{\mathrm{t}}]$. Here the important point to note is that the number of samples used in the source model dataset (N) is much greater than the number of samples used in the target model dataset (M) i.e., $M < N$.

D. THE PROPOSED CNN-BASED TL ALGORITHM

First, we design the source model $\mathcal{PT}(\mathbf{X}; \boldsymbol{\theta}^{\mathrm{src}})$ with $\mathcal{F}^{\mathrm{src}}$ being the set of all layers in the source model. Specifically, the source model is a CNN model designed for training on a large dataset to achieve a suitable robust performance in terms of accurate BLER predictions. This well-trained source model serves as our starting point. After training, layers $\mathcal{F}^{\mathrm{freeze}} \subseteq \mathcal{F}^{\mathrm{src}}$ are frozen to preserve these learned features.

Our second CNN model, i.e., the NFT target model, has fewer layers and less training data. We customize the source model for our task by adding a new fully-connected layer with appropriate output neurons, trained on our dataset. Since the source model has pre-trained features, we only train the newly added layers. We then fine-tune the source model by unfreezing certain layers in the source model, transferring its trained parameters to the non-fine-tuned (NFT) target model to better adapt to our task. After training and fine-tuning, the model's performance is evaluated on a validation set. If unsatisfactory, we modify the hyperparameters for improved performance and subsequently use this fine-tuned model for predictions. Details of the transfer learning algorithm are presented in Algorithm 1, and the descriptions of the source, NFT, and fine-tuned target models are explained in subsequent sections.

E. SOURCE CNN MODEL FOR SOURCE TASK

The source model, which consists of convolutional layers, pooling layers, fully connected layers, and output layers, is written as

$$\mathcal{PT}(\mathbf{X}; \boldsymbol{\theta}^{\mathrm{src}})$$
$$= [f_{\mathrm{fc}_6}^{(11)}(\boldsymbol{\theta}_{\mathrm{fc}_6}^{\mathrm{src}}) \circ f_{\mathrm{fc}_5}^{(10)}(\boldsymbol{\theta}_{\mathrm{fc}_5}^{\mathrm{src}}) \circ f_{\mathrm{fc}_4}^{(9)}(\boldsymbol{\theta}_{\mathrm{fc}_4}^{\mathrm{src}})$$

Algorithm 1 Proposed Adaptation Algorithm Based on Transfer Learning

1: **Initialize**: BATCH_SIZE $= S_{\mathrm{batch}}$, NUM_EPOCH $= 30$, SPLIT_SIZE $= 0.1$;
2: **Load**: Source dataset $\mathcal{S}_{\mathrm{s}} = \{(\boldsymbol{\mathcal{X}}_{\mathrm{s}}^{(i)}, y_{\mathrm{s}}^{(i)}) : \boldsymbol{\mathcal{X}}_{\mathrm{s}}^{(i)} \in \mathcal{X}_{\mathrm{s}}, y_{\mathrm{s}}^{(i)} \in \mathcal{Y}_{\mathrm{s}}, i \in [1, L_{\mathrm{s}}]\}$;
3: **Load**: Target dataset $\mathcal{S}_{\mathrm{t}} = \{(\boldsymbol{\mathcal{X}}_{\mathrm{t}}^{(i)}, y_{\mathrm{t}}^{(i)}) : \boldsymbol{\mathcal{X}}_{\mathrm{t}}^{(i)} \in \mathcal{X}_{\mathrm{t}}, y_{\mathrm{t}}^{(i)} \in \mathcal{Y}_{\mathrm{t}}, i \in [1, L_{\mathrm{t}}]\}$;
4: **Phase 1**: Source model training phase,
5:   **input** Source training dataset $\mathcal{S}_{\mathrm{s}}^{\mathrm{train}} \subset \mathcal{S}_{\mathrm{s}}$, where $|\mathcal{S}_{\mathrm{s}}^{\mathrm{train}}| = L_{\mathrm{s}}^{\mathrm{train}}$ and $\frac{L_{\mathrm{s}}^{\mathrm{train}}}{L_{\mathrm{s}}} = 1 - $ SPLIT_SIZE;
6:   **build** Source CNN model following structure in (55);
7:   $i \leftarrow 0$;
8:   **while** $i < $ NUM_EPOCH **do**
9:     $i \leftarrow i + 1; j \leftarrow 0; \boldsymbol{\theta}_{(i,j)}^{\mathrm{src}} \leftarrow \boldsymbol{\theta}_{(i-1,j)}^{\mathrm{src}}$;
10:     **while** $j < $ NO_BATCH **do**
11:       $j \leftarrow j + 1; \boldsymbol{\theta}_{(i,j)}^{\mathrm{src}} \leftarrow$ Optimizer$(\boldsymbol{\theta}_{(i,j-1)}^{\mathrm{src}}, \nabla \mathcal{L}_{\mathrm{batch}}(\mathcal{B}_{\mathrm{s}}^{(i,j)}; \boldsymbol{\theta}_{(i,j-1)}^{\mathrm{src}}))$;
12:   **return**: trained parameters $\tilde{\boldsymbol{\theta}}^{\mathrm{src}} \leftarrow \boldsymbol{\theta}_{(i,j)}^{\mathrm{src}}$ and pre-trained source model $\mathcal{PT}(\mathbf{X}; \tilde{\boldsymbol{\theta}}^{\mathrm{src}})$;
13: **Phase 2**: FT model training phase
14:   **input** Target training dataset $\mathcal{S}_{\mathrm{t}}^{\mathrm{train}} \subset \mathcal{S}_{\mathrm{t}}$, where $|\mathcal{S}_{\mathrm{t}}^{\mathrm{train}}| = L_{\mathrm{t}}^{\mathrm{train}}$ and $\frac{L_{\mathrm{t}}^{\mathrm{train}}}{L_{\mathrm{t}}} = 1 - $ SPLIT_SIZE;
15:   **build** FT CNN model with structure in (65);
16:   $i \leftarrow 0$;
17:   **while** $i < $ NUM_EPOCH **do**
18:     $i \leftarrow i + 1; j \leftarrow 0; \boldsymbol{\theta}_{(i,j)}^{\mathrm{ft}} \leftarrow \boldsymbol{\theta}_{(i-1,j)}^{\mathrm{ft}}$;
19:     **while** $j < $ NO_BATCH **do**
20:       $j \leftarrow j + 1$;
      $\boldsymbol{\theta}_{(i,j)}^{\mathrm{ft}} \leftarrow$ Optimizer$(\boldsymbol{\theta}_{(i,j-1)}^{\mathrm{ft}}, \nabla \mathcal{L}_{\mathrm{batch}}(\mathcal{B}_{\mathrm{t}}^{(i,j)}; \boldsymbol{\theta}_{(i,j-1)}^{\mathrm{ft}}))$;
21:   **return**: trained parameters $\tilde{\boldsymbol{\theta}}^{\mathrm{ft}} \leftarrow \boldsymbol{\theta}_{(i,j)}^{\mathrm{ft}}$ and fine-tuned model $\mathcal{FT}(\mathbf{X}; \tilde{\boldsymbol{\theta}}^{\mathrm{ft}})$;
22: **Phase 3**: Inference phase,
23:   **Input**: Test dataset $\mathcal{S}_{\mathrm{test}} = \{(\boldsymbol{\mathcal{X}}_{\mathrm{test}}^{(i)}, y_{\mathrm{test}}^{(i)}) : \boldsymbol{\mathcal{X}}_{\mathrm{test}}^{(i)} \in \mathcal{X}_{\mathrm{test}}, y_{\mathrm{test}}^{(i)} \in \mathcal{Y}_{\mathrm{test}}, i \in [1, L_{\mathrm{test}}]\}$;
24:   **Compute**: RMSE $= \sqrt{\frac{1}{L^{test}} \sum_{i=1}^{L_{\mathrm{test}}} (\hat{y}_{\mathrm{test}}^{(i)} - y_{\mathrm{test}}^{(i)})^2}$, where $\hat{y}_{\mathrm{test}}^{(i)} = \mathcal{FT}(\boldsymbol{\mathcal{X}}_{\mathrm{test}}^{(i)}; \tilde{\boldsymbol{\theta}}^{\mathrm{ft}})$;
25:   **if** RMSE $> $ criterion **then**
26:     **Go to** Phase 2;
27:   **else**
28:     **Output**: Trained FT model $\mathcal{FT}(\mathbf{X}; \tilde{\boldsymbol{\theta}}^{\mathrm{ft}})$;

$$\circ f_{\mathrm{fc}_3}^{(8)}(\boldsymbol{\theta}_{\mathrm{fc}_3}^{\mathrm{src}}) \circ f_{\mathrm{fc}_2}^{(7)}(\boldsymbol{\theta}_{\mathrm{fc}_2}^{\mathrm{src}}) \circ f_{\mathrm{fc}_1}^{(6)}(\boldsymbol{\theta}_{\mathrm{fc}_1}^{\mathrm{src}}) \circ f_{\mathrm{flat}}^{(5)}$$
$$\circ f_{\mathrm{p}_2}^{(4)} \circ f_{\mathrm{c}_2}^{(3)}(\boldsymbol{\theta}_{\mathrm{c}_2}^{\mathrm{src}}) \circ f_{\mathrm{p}_1}^{(2)} \circ f_{\mathrm{c}_1}^{(1)}(\boldsymbol{\theta}_{\mathrm{c}_1}^{\mathrm{src}})](\mathbf{X}), \quad (55)$$

where $(f_2 \circ f_1(\boldsymbol{\theta}))(\mathbf{x}) \triangleq f_2(f_1(\mathbf{x}; \boldsymbol{\theta}))$, $(f_2(\boldsymbol{\theta}) \circ f_1)(\mathbf{x}) \triangleq f_2(f_1(\mathbf{x}); \boldsymbol{\theta})$ and $(f_2(\boldsymbol{\theta}_2) \circ f_1(\boldsymbol{\theta}_1))(\mathbf{x}) \triangleq f_2(f_1(\mathbf{x}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2)$, $\boldsymbol{\theta}_{\mathrm{lay}}$ represents the weights of layer lay, $f_{\mathrm{lay}_t}^{(l)}(\boldsymbol{\theta}_{\mathrm{lay}_t})$ represent the operation at the layer lay $\in \{\mathrm{c}, \mathrm{p}, \mathrm{flat}, \mathrm{fc}\}$,

$t \geq 1$ denotes the index of the layer lay, and $\boldsymbol{\theta}^{\text{src}} = \{\boldsymbol{\theta}^{\text{src}}_{\text{C}_1}, \boldsymbol{\theta}^{\text{src}}_{\text{C}_2}, \ldots, \boldsymbol{\theta}^{\text{src}}_{\text{fC}_3}, \ldots\}$.

1) ARCHITECTURE

Our CNN is composed two main parts, a convolution feature extraction part and a regression part. Specifically, the feature extraction part consists of several convolutional (Conv2D) layers and max-pooling (MaxPooling2D) layers, followed by the regression part, which consists of multiple fully-connected (Dense) layers. To ensure proper handling of regression tasks, it is essential to position a fully connected layer before the regression layer in the network's output.

a: 2D CONVOLUTIONAL LAYER (CONV2D-T)

First, p_t padding is applied to the input $\mathbf{A}^{(l-1)}$ of size $H^{(l-1)} \times W^{(l-1)} \times C^{(l-1)}$ to increase its spatial dimension, which involves adding extra rows and columns of zeros around the input. Let $\mathbf{A}^{(l-1)}_t$ be the input after padding, the relationship between $\mathbf{A}^{(l-1)}_t$ and the input $\mathbf{A}^{(l-1)}$ is $[\mathbf{A}^{(l-1)}_t]_{i,j,c} = [\mathbf{A}^{(l-1)}]_{i-p_t,j-p_t,c}$ if $1 + p_t \leq i \leq H^{(l-1)} + p_t$ and $1 + p_t \leq j \leq W^{(l-1)} + p_t$, otherwise $[\mathbf{A}^{(l-1)}_t]_{i,j,c} = 0$. It is noted that $\mathbf{A}^{(0)} = \mathbf{X}$ and $\mathbf{A}^{(0)}_t$ is the padded matrix.

The next step consists of applying a CNN operation using a kernel that applies a convolutional operation to the input data. Given $C^{(l)}$ kernels, we define each l-th kernel as a matrix $[W^{(l)}_t]c$ with $P^{(l)}$ rows and $Q^{(l)}$ columns, where $[W^{(l)}_t]p,q,c = w^{(l)}_{p,q,c}$. Each kernel yields a feature map $[Z^{(l)}]_c$ of size $H^{(l)} \times W^{(l)}$, where $H^{(l)} = \frac{H^{(l-1)} + 2\,p_t - P^{(l)}}{s} + 1$ and $W^{(l)} = \frac{W^{(l-1)} + 2\,p_t - Q^{(l)}}{s} + 1$ with s being the stride. The feature map $[\mathbf{Z}^{(l)}_t]_c$ is presented as

$$[\mathbf{Z}^{(l)}]_c = b^{(l)}_c + [\mathbf{W}^{(l)}_t]_c * [\mathbf{A}^{(l-1)}_t]_c, \tag{56}$$

$$[\mathbf{Z}^{(l)}]_{i,j,c} = b^{(l)}_c + \sum_{u=1}^{P} \sum_{v=1}^{Q} w^{(l)}_{v,u,c} \bar{a}^{(l-1)}_{s(i-1)+u-p_t, s(j-1)+v-p_t, c}, \tag{57}$$

$\forall c \in [1, C^{(l)}]$, where $\bar{a}^{(l)}_{i,j,c} = [\mathbf{A}^{(l)}_t]_{i,j,c}$ and $\mathbf{b}^{(l)}_t = [b_1, b_2, \ldots, b_{C^{(l)}}]^{\mathsf{T}}$ is the bias of the t-th convolutional layer. After the convolution operation, an element-wise non-linear activation function $\phi^{(l)}_{\text{C}_t}$ is applied to introduce non-linearity into the network. The output of Conv2D-1 is $\mathbf{A}^{(l)} \triangleq f^{(l)}_{\text{C}_t}(\mathcal{A}^{(l-1)}; \boldsymbol{\theta}^{\text{src}}_{\text{C}_t})$, that is a 3D matrix of size $H^{(l)} \times W^{(l)} \times C^{(l)}$. The c-th channel of $\mathbf{A}^{(l)}$, denoted as $[\mathbf{A}^{(l)}]_c$, is formulated as

$$\mathbf{A}^{(l)} = \phi^{(l)}_{\text{C}_t}(\mathcal{Z}^{(l)}). \tag{58}$$

Herein, the parameters of the Conv2D-t layer are denoted as $\boldsymbol{\theta}^{\text{src}}_{\text{C}_t} = \{\lfloor^{(l)}_t, \mathbf{W}^{(l)}_t\}$.

b: MAX POOLING LAYER (MAXPOOLING2D-T)

Maxpooling is a pooling operation that reduces the spatial dimension of the feature map while retaining the most important features. Let $\mathbf{A}^{(l)} = f^{(l)}_{p_t}(\mathcal{A}^{(l-1)})$ be the output of the t-th maxpooling layer, we have

$$[\mathbf{A}^{(l)}]_c = \max_{1 \leq u \leq P^{(l)}} \max_{1 \leq v \leq Q^{(l)}} \bar{a}^{(l-1)}_{s(i-1)+u, s(j-1)+v, c}, \tag{59}$$

where $P^{(l)}$ and $Q^{(l)}$ specify the kernel size and s is the stride. It is noted that given the input $\mathbf{A}^{(l-1)}$ of size $H^{(l-1)} \times W^{(l-1)} \times C^{(l-1)}$, the size of $\mathbf{A}^{(l)}$ of the MaxPooling2D-t layer is obtained with $H^{(l)} = \left\lfloor \frac{H^{(l-1)} - P^{(l)}}{s} \right\rfloor + 1$, $W^{(l)} = \left\lfloor \frac{W^{(l-1)} - Q^{(l)}}{s} \right\rfloor + 1$, and $C^{(l)} = C^{(l-1)}$.

c: FLATTEN LAYER (FLAT)

Before transforming our CNN into a multi-layer FC CNN, we flatten the input matrix $\mathbf{A}^{(l-1)}$ to the flatten output vector $\mathbf{a}^{(l)} \in \mathbb{R}^{d^{(l)}}$ where $d^{(l)} = H^{(l-1)} W^{(l-1)} C^{(l-1)}$.

d: FULLY CONNECTED (FC) LAYER (DENSE-T)

Followed by the flatten layer, FC layers compute a linear transformation followed by an activation function $\phi^{(l)}_{\text{fC}_t}$ on the preceding layer's flattened output. Let $\mathbf{a}^{(l-1)} \in \mathbb{R}^{d^{(l-1)}}$, the output vector $\mathbf{a}^{(l)} \triangleq f_{\text{fC}_t}(\dashv^{(l-1)}, \boldsymbol{\theta}^{\text{src}}_{\text{fC}_t})$ is formulated as [58]

$$\mathbf{a}^{(l)} = \phi^{(l)}_{\text{fC}_t}(\ddagger^{(l)}), \tag{60}$$

$$\mathbf{z}^{(l)} = \mathbf{b}^{(l)}_t + (\mathbf{W}^{(l)}_t)^{\mathsf{T}} \mathbf{a}^{(l-1)}, \tag{61}$$

where $\mathbf{W}^{(l)}_t \in \mathbb{R}^{d^{(l-1)} \times d^{(l)}}$ is the weight and $\mathbf{b}^{(l)}_t \in \mathbb{R}^{d^{(l)}}$ is the bias of the t-th fully connected layer. Herein, the parameters of the Dense-t layer are denoted as $\boldsymbol{\theta}^{\text{src}}_{\text{fC}_t} = \{\lfloor^{(l)}_t, \mathbf{W}^{(l)}_t\}$.

2) LOSS FUNCTIONS IN TRAINING A MODEL

We assess the performance of our model in terms of root mean square error (RMSE) function. The RMSE is used to determine the discrepancy between the predicted value and the actual output value across the entire test set. During the training phase, the loss function for each (i, j)-th mini-batch $\mathcal{B}^{(i,j)}$ of each i-th epoch is calculated. This loss function is formulated as

$$\mathcal{L}_{\text{batch}}(\mathcal{B}^{(i,j)}; \boldsymbol{\theta}^{\text{src}}) = \frac{1}{|\mathcal{B}^{(i,j)}|} \sum_{k=1}^{|\mathcal{B}^{(i,j)}|} (\bar{\epsilon}^{(k)}_{e2e} - \hat{\epsilon}^{(k)}_{e2e})^2, \tag{62}$$

where $\bar{\epsilon}^{(k)}_{e2e}$ and $\hat{\epsilon}^{(k)}_{e2e} = \mathcal{PT}(\mathbf{X}^{(k)}; \boldsymbol{\theta}^{\text{src}})$ for $\mathbf{X}^{(k)} \in \mathcal{B}^{(i,j)}$ are the true average e2e BLER and the predicted average e2e BLER, respectively. After training each i-th epoch, the loss function over all mini-batches is formulated as

$$\mathcal{L}_{\text{epoch}}(\mathcal{B}^{(i,1)}, \ldots, \mathcal{B}^{(i, N_{\text{batch}})}; \boldsymbol{\theta}^{\text{src}})$$
$$= \frac{1}{N_{\text{batch}}} \sum_{j=1}^{N_{\text{batch}}} \mathcal{L}_{\text{batch}}(\mathcal{B}^{(i,j)}; \boldsymbol{\theta}^{\text{src}}), \tag{63}$$

where N_{batch} denotes the number of mini-batches in the training dataset.

F. NFT *MODEL FOR TARGET TASK*

We utilize a reduced version of the original dataset in this part. This allows us to run the models more quickly, which is beneficial for applications with limited compute power. The mathematical representation of the NFT target model is formulated as

$$\mathcal{NFT}(\mathbf{X}; \boldsymbol{\theta}^{\mathtt{nft}}) = [f_{\mathtt{fc}_1}^{(4)}(\boldsymbol{\theta}_{\mathtt{fc}_1}^{\mathtt{nft}}) \circ f_{\mathtt{flat}}^{(3)}(\boldsymbol{\theta}_{\mathtt{flat}}^{\mathtt{nft}}) \circ f_{\mathtt{C}_1}^{(2)}(\boldsymbol{\theta}_{\mathtt{C}_1}^{\mathtt{nft}})$$
$$\circ f_{\mathtt{out}}^{(1)}(\boldsymbol{\theta}_{\mathtt{out}}^{\mathtt{nft}})](\mathbf{X}), \qquad (64)$$

where $\boldsymbol{\theta}^{\mathtt{nft}} \triangleq \{\boldsymbol{\theta}_{\mathtt{C}_1}^{\mathtt{nft}}, \boldsymbol{\theta}_{\mathtt{flat}}^{\mathtt{nft}}, \boldsymbol{\theta}_{\mathtt{fc}_1}^{\mathtt{nft}}, \boldsymbol{\theta}_{\mathtt{out}}^{\mathtt{nft}}\}$.

The training set for a NFT target model is a subset of the dataset used for the NFT target model that is used to train the model. We take a smaller subset of the training data by randomly selecting a subset of size M (where $M < N$) from the original dataset of size N. Let $\mathcal{D}_{\mathtt{nft}}^{\mathtt{train}}$ and $\mathcal{D}_{\mathtt{nft}}^{\mathtt{test}}$ denote the train and test sets for the NFT models. The training set $\mathcal{D}_{\mathtt{nft}}^{\mathtt{train}}$ can be written as $\mathcal{D}_{\mathtt{train}} = (\bar{x}_j, \bar{y}_j)_{j=1}^{n}$, $(x_j, y_j) \in \{\mathcal{D}_t, \mathcal{T}_t\}$, where n is the number of samples in the training set for the NFT target model. The loss function during the training phase of our NFT model, denoted as $\mathcal{L}_{\mathtt{batch}}(\mathcal{B}^{(i,j)}; \boldsymbol{\theta}^{\mathtt{nft}})$, is similar to (62) with $\hat{\epsilon}_{e2e}^{(k)} = \mathcal{NFT}(\mathbf{X}^{(k)}; \boldsymbol{\theta}^{\mathtt{nft}})$.

G. FINE-TUNED (FT) MODEL FOR THE TARGET TASK

We create a new model with the same architecture as the source model, but with the frozen layers excluded. This new model is used for fine-tuning. We begin by transferring the weights from the unfrozen layers in the pre-trained source model to the fine-tuned model, preserving the learned knowledge from the source dataset. Subsequently, we train this fine-tuned model using the fine-tuning dataset. The new fine-tuned target model can be formulated as

$$\mathcal{FT}(\mathbf{X}; \boldsymbol{\theta}^{\mathtt{ft}}) = [f_{\mathtt{out}}^{(13)} \circ \overbrace{f_{\mathtt{fc}_7}^{(12)}(\boldsymbol{\theta}_{\mathtt{fc}_7}^{\mathtt{ft}})}^{\text{new layer}}$$
$$\circ \overbrace{f_{\mathtt{fc}_6}^{(11)}(\boldsymbol{\theta}_{\mathtt{fc}_6}^{\mathtt{src}}) \circ \cdots \circ f_{\mathtt{fc}_t}^{(l_t)}(\boldsymbol{\theta}_{\mathtt{fc}_t}^{\mathtt{src}})}^{\text{trainable layers}}$$
$$\circ f_{\mathtt{freeze}}(\tilde{\boldsymbol{\theta}}_{\mathtt{freeze}}^{\mathtt{src}})](\mathbf{X}), \qquad (65)$$

where l_t and t are the indices of the last trainable FC layer, $f_{\mathtt{freeze}}$ presents the non-trainable (freezing) layers, $\tilde{\boldsymbol{\theta}}_{\mathtt{freeze}}^{\mathtt{src}}$ are the non-trainable parameters of the source model due to freezing layers, and $\boldsymbol{\theta}^{\mathtt{ft}} \triangleq \{\boldsymbol{\theta}_{\mathtt{fc}_7}^{\mathtt{ft}}, \boldsymbol{\theta}_{\mathtt{fc}_6}^{\mathtt{src}}, \ldots, \boldsymbol{\theta}_{\mathtt{fc}_t}^{\mathtt{src}}\}$ are the trainable parameters of the FT model.

The loss function in our FT model, denoted as $\mathcal{L}_{\mathtt{batch}}(\mathcal{B}^{(i,j)}; \boldsymbol{\theta}^{\mathtt{ft}})$, is similar to (62) with $\hat{\epsilon}_{e2e}^{(k)} = \mathcal{FT}(\mathbf{X}^{(k)}; \boldsymbol{\theta}^{\mathtt{ft}})$.

H. TESTING OF THE PROPOSED TL ALGORITHM AND ALL MODELS

1) PERFORMANCE COMPARISON OF SOURCE VERSUS NFT VERSUS FINE-TUNED TARGET MODEL

The RMSE performance comparison among source model with 5 hidden layers, the NFT target model with double the

number of hidden layers, and the fine-tuned target model (transfer learning model) with double the number of hidden layers of trainable parameters value is given in Table 4.

2) INFERENCE ON NEW DATA

Deep learning inference is the process of using a fully trained CNN to generate inferences (predictions) based on novel (unknown) data. For inference on new data, in our work, we load the qualified trained models $\mathcal{PT}(\boldsymbol{\theta}^{\mathtt{src}})$, $\mathcal{NFT}(\boldsymbol{\theta}^{\mathtt{nft}})$, and $\mathcal{FT}(\boldsymbol{\theta}^{\mathtt{ft}})$ and feed new inference data $\mathbf{x}' = [\mathbf{p}_{\mathtt{S}}'^{\top}, \mathbf{p}_{\mathtt{D}_1}'^{\top}, \ldots, \mathbf{p}_{\mathtt{D}_K}'^{\top}, \mathbf{p}_{I_1}'^{\top}, \ldots, \mathbf{p}_{I_I}'^{\top}, \mathbf{p}_{\mathtt{C}}'^{\top}, F', M', m'_{\mathtt{SC}}, m'_{\mathtt{CD}_1}, \ldots, m'_{\mathtt{CD}_K}, \ell'_{\mathtt{SC}}, \ell'_{\mathtt{CD}_1}, \ldots, \ell'_{\mathtt{CD}_K}, N]^{\top}$ into these trained networks. The output $\hat{\epsilon}_{e2e}^{(i)}$ is calculated through the fine-tuned trained model as follows $\bar{\epsilon}_{e2e}^{(i)} = \mathcal{FT}(\boldsymbol{\theta}^{\mathtt{ft}})(\S')$, and then compared $\hat{\epsilon}_{e2e}^{(i)}$ with analytical results.

3) EVALUATION OF THE TESTING

We can predict our output from the trained model as $\mathcal{FT}(\mathbf{X}^{test}; \boldsymbol{\theta}^{\mathtt{ft}}) \rightarrow \hat{\epsilon}_{e2e,test}^{(i)}$. To determine whether or not transfer learning works, we consider the RMSE in order to assess the fine-tuned model. The RMSE is a typical means of quantifying the quality of a model's fit in statistical modelling, particularly regression analysis. It is given as

$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{L^{test}} \left(\hat{\epsilon}_{e2e}^{(i)} - \bar{\epsilon}_{e2e,test}^{(i)}\right)^2}{L^{test}}}$ where, $L^{test} = L^{data} * 10\%$ and L^{data} denotes the total number of datapoints of the dataset. In addition, the RMSE of the transfer-model is significantly lower than that of the NFT model and is even comparable to that of the source model. The CNN structure of our proposed method is given in Fig. 2, where Eq. (55)) and Eq. (65) represent the output functions of the network.

I. COMPLEXITY ANALYSIS

For input $\mathbf{A}^{(l-1)}$ with $C^{(l-1)}$ channels, $C^{(l)}$ kernels with size $P^{(l)} \times Q^{(l-1)}$, and feature map of size $H^{(l)} \times W^{(l)}$, the time complexity is given in Tab. 2: Complexity ($\mathcal{O}$) of Algorithm 1.
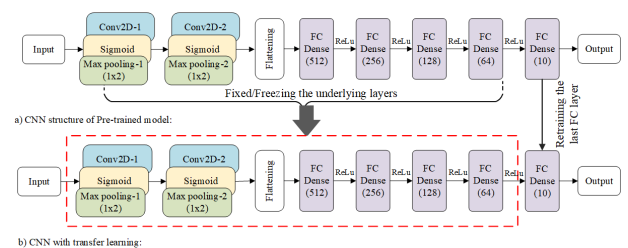

FIGURE 2. Proposed neural network structure a) CNN structure of pre-trained model, b) CNN with transfer learning.

V. RESULTS AND DISCUSSIONS

In this section, we present illustrative numerical results to show the performances of the proposed schemes in-terms of average e2e BLER. In order to evaluate the performance of our system model we first validate our system model

TABLE 2. Complexity ($\mathcal{O}$) of Algorithm 1.

Model	Training complexity
Source model $\mathcal{PT}(\mathbf{X}; \boldsymbol{\theta}^{S})$	$\mathcal{O}\left(1 + \max_{1 \leq l \leq 2}\left\{C^{(l)}H^{(l)}W^{(l)}P^{(l-1)}Q^{(l-1)}C^{(l-1)}\right\} + \max_{6 \leq l \leq 11}\left\{d^{(l-1)}d^{(l)}\right\}\right)$
Non fine tuned model $\mathcal{NFT}(\mathbf{X}; \boldsymbol{\theta}^{\text{nft}})$	$\mathcal{O}\left(1 + C^{(1)}H^{(1)}W^{(1)}P^{(0)}Q^{(0)}C^{(0)} + \max\left\{d^{(3)}d^{(2)}, d^{(4)}d^{(3)}\right\}\right)$

TABLE 3. Simulation parameters.

Parameter	Value
Blocklength (M)	300
Carrier frequency, f_c [GHz]	3
Nakagami-m shape parameters	[3,5]
Noise figure, NF [dBm]	10
No. ground users, K	3
No. information bits, F [bits]	100
No. Kernel per Conv-2D layer	4
No. neuron per Dense-t layer	$1024/2^{t-1}$
Transmit power at S, P_S [dBm]	[0,30]
Transmit power at C, P_C [dBm]	[0,30]
Thermal noise power density [dBm/Hz]	-174
No. antenna, N	4
No. interferer, L	3
Noise bandwidth, BW [MHz]	10
Kernel size	(2,3)

and verify the average e2e BLER of the system ($\bar{\epsilon}_{e2e}$). We investigate the effect of SNR, block-length, number of bits transmitted, and transmit power on the average overall BLER of the system. Finally, in this section, the comparison between simulation based prediction and analytical results are also discussed in this section. This section is further divided into following subsections.

A. AVERAGE BLOCK ERROR RATE

This subsection explains the average BLER as influenced by the distribution of transmission powers (P_S and P_C) and interference power distribution. We further incorporated Fig. 3 to facilitate comparison with a benchmark technique previously reported in [59].
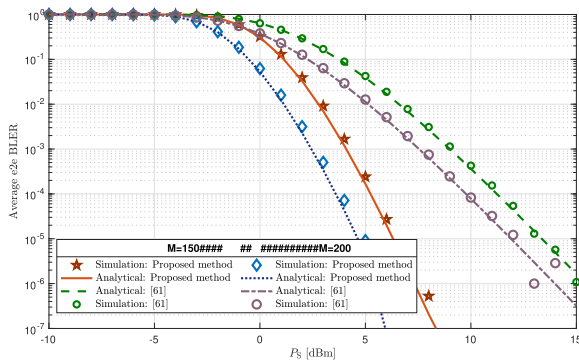
FIGURE 3. Performance comparison of proposed method and [59] using no interferer.

In Fig. 3, we present the results of the comparative analysis with [59]. With an increase of P_S, we observed a notable enhancement in the BLER of both proposed method and [59].

Specifically, Fig. 3 illustrates that, under no interferer, our proposed method showed superior BLER performance as compared to the method discussed in [59]. Furthermore, we found that, by increasing the value of M, a better BLER is achieved for both methods. This emphasizes the significance of optimizing M to achieve improved performance in wireless communication systems.
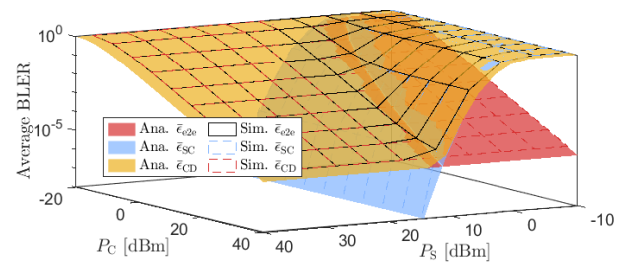

FIGURE 4. Average e2e BLER as a function of transmission powers with $M = 500$ symbol/block.

In Fig. 4, the average BLER of the first hop, second hop, and e2e BLER are observed under different values of P_S and P_C. First, it is observed that the analytical results matched well with the simulation results, which shows the high accuracy of the proposed approximation. It is observed that when both P_S and P_C increase, the average BLER of each hop and of the system also increase. This is because the system achieves diversity gain as $P_S \to \infty$ and $P_C \to \infty$. However, when fixing either P_S or P_C and varying P_C or P_S, respectively, we observe error floor levels. Specifically, the error floor at $P_C = 15$ dBm is shown in Fig. 4.

The insights obtained from Fig. 4 regarding the impact of power levels on BLER directly enlighten the analysis presented in Fig. 5, where we explore the broader implications of transmission parameters, such as block length, on network performance.
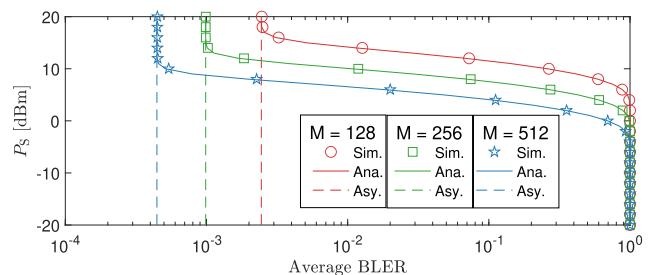

FIGURE 5. Source transmission power as a function of the average e2e BLER with different values of blocklengths.

In Fig. 5, we show the transmit power P_S versus the average e2e BLER with different blocklengths, i.e., $M = 128, 256$, and $M = 512$. The analytical, simulation, and asymptotic curves are shown in this figure when the number of information bits received by the D_k (F) $= 100$, and $P_C = 15$ dBm. Once more, the analytical curves match well with the simulation curves, which confirms the validity of our analysis. Furthermore, as shown in Fig. 5, for fixed channel conditions, when the blocklength M is increased, the average e2e BLER and the required P_S drop. Next, we observe the average e2e BLER when varying the blocklength. The insights gained from subsection V-A regarding the impact of power levels on BLER directly enlighten the analysis presented in subsection V-B where we explore the broader implications of transmission parameters such as block length, on network performance.

B. IMPACT OF BLOCKLENGTH

Here, we explore into the impact of M on the average BLER under varying values of P_S. Building upon this foundation from Fig. 5, where we utilized both analytical and simulation results to examine the impact of varying BS power levels P_S and M on BLER, helped us to delved into the influence of interference power distribution on the average block length in Fig. 6.
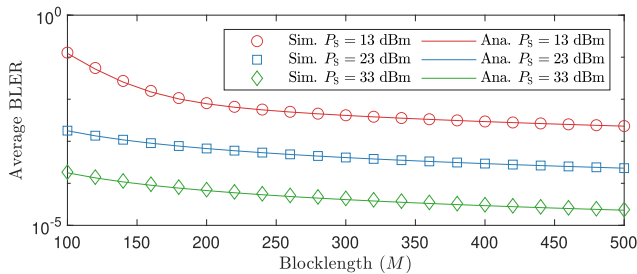
FIGURE 6. Impact of blocklength to the average BLER with different values of source transmission power.

In Fig. 6, we investigate the impact of the blocklength on the average e2e BLER with a maximum blocklength, M, set to 500 symbol/block and a source power, P_S, of 33 dBm. The observed trend shows a decrease in the average e2e BLER as the blocklength increases. Specifically, at a blocklength of $M = 100$ symbols/block, elevating the source power, P_S, from 13 dBm to 23 dBm induces an approximately two orders of magnitude decrease in the average e2e BLER. However, an additional 10 dBm increase in P_S results in only a single order of magnitude reduction in the average e2e BLER. This phenomenon is due to the presence of an error floor in the system at a fixed cluster head power, P_C, as detailed in the discussion of Fig. 5.

In Fig. 7, we examine the influence of M on the average e2e BLER, with a maximum block M set to 300 symbols per block and a source power P_{src} of 30 dBm. The observed trend indicates a decrease in the average e2eBLER with
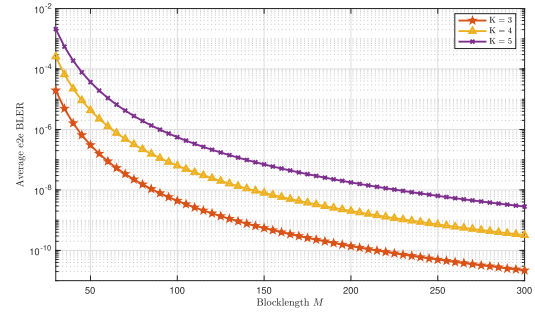
FIGURE 7. Impact of block length M and number of users K with constant P_{src}, P_C, and M.

an increase of M. However, we observe an increase in the average end-to-end block error rate is observed when the number of users k is increased, while P_{src}, P_C, and M remain constant. Fig. 7 provides valuable insights into how both block length and the number of users affect the reliability and delay of transmissions in the system.
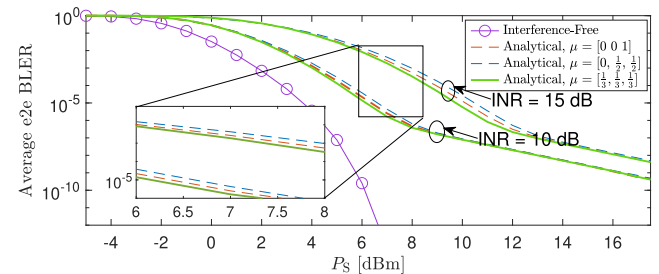
FIGURE 8. Impact of interference power distribution on the average blocklength, where $m_{I_iC} = m_{I_iD_k}$ for all $i \in [1, I]$ and $k \in [1, K]$.

In Fig. 8, the impact of the interference power distribution, denoted as μ, on the average e2e (system) BLER is discussed. It is noted that

$$\mu = \frac{1}{\sum_{i=1}^{I} P_{I_iC}\ell_{I_iC}}[P_{I_1C}\ell_{I_1C}, \ldots, P_{I_IC}\ell_{I_IC}]$$
$$= \frac{1}{\sum_{i=1}^{I} P_{I_iD_k}\ell_{I_iD_k}}[P_{I_iD_k}\ell_{I_iD_k}, \ldots, P_{I_ID_k}\ell_{I_ID_k}];$$
$$\forall k \in [1, K] \qquad (66)$$

The interference-free condition, i.e., $I = 0$ interferers, results in the lowest average e2e BLER. Interestingly, the system performance is not only impacted by the presence of interference but also by the variation of μ under the same INR values. Specifically, the lowest average e2e BLER is observed in an IID interference environment.

Building upon this foundation from Fig. 6, where we utilized both analytical and simulation results to examine the impact of varying BS power levels P_S and M on BLER, helped us to delved into the influence of interference power distribution on the average block length in Fig. 8.
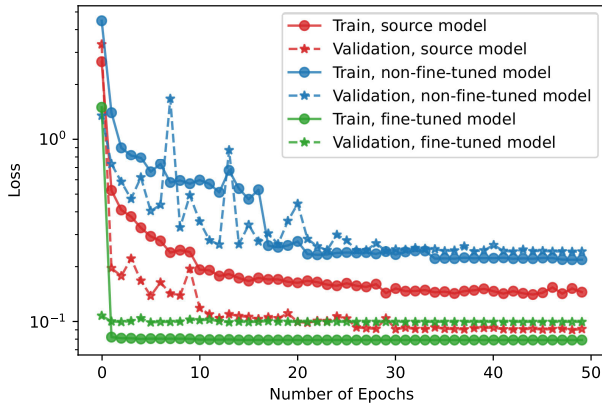
FIGURE 9. Training convergence of the source, non-fine-tune, and fine-tune models.

C. CONVERGENCE AND PERFORMANCE OF TRANSFER LEARNING

This subsection addresses the convergence of training for the source, non-fine-tuned, and fine-tuned models, along with a comparative analysis between the true (analytical) curve and the predicted results derived from the proposed method.

In Fig. 9, we present the convergence speeds of the NFT model, the source model, and the fine-tuned model. The source model requires approximately 35 epochs to converge, while the NFT model fares better, converging at about 21 epochs. Remarkably, the fine-tuned model reaches convergence swiftly, within just 5 epochs. Furthermore, the fine-tuned model's training and validation accuracies closely align with the validation accuracy of the source model.
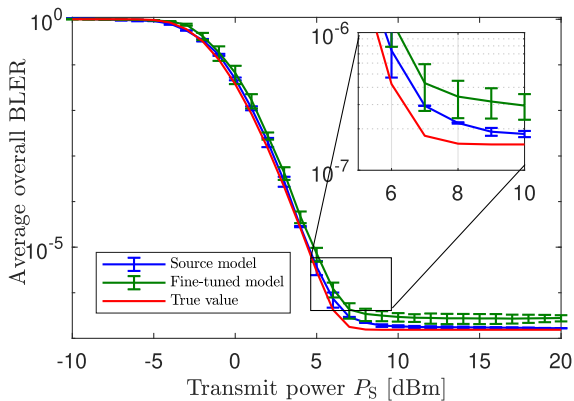

FIGURE 10. Impact of source's transmission power on the true and predicted BLER of the source and the fine-tuned models.

More specifically, in Fig. 10, the error bars representing the confidence intervals of the NFT, source, and fine-tuned models are clearly visible. Importantly, it is observed that as the transmission power increases, these confidence intervals (error bars) decrease, implying reduced variability and enhanced precision in the models' predictions. This reduction in the size of the confidence intervals as transmission power increases can be attributed to the fact that a higher

transmission power typically results in stronger signal quality, thereby reducing uncertainty in the data and leading to more consistent and reliable model predictions.

Fig. 9 and Fig. 10 offered insights into the training convergence and performance evaluation of our models. In Fig. 9, we thoroughly analyzed the training convergence of the source, non-fine-tuned, and fine-tuned models, providing a detailed understanding of their respective learning processes. Building upon this analysis, Fig. 10 delved into the impact of the source's transmission power on the true and predicted BLER of both the source and fine-tuned models.
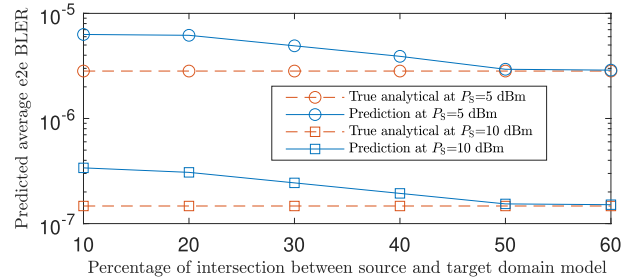

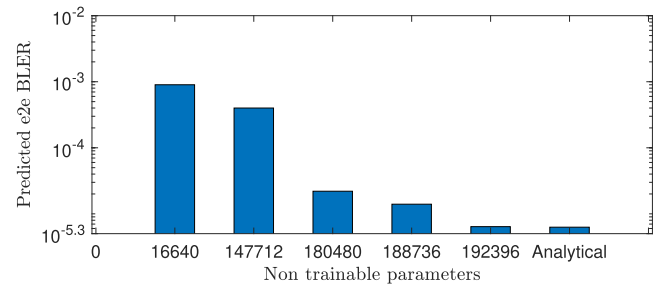**FIGURE 11.** Impact of adaptation domain on prediction accuracy.

FIGURE 12. Impact of transferred knowledge amount on prediction accuracy.

D. IMPACT OF DOMAIN ADAPTATION AND TRANSFERRED KNOWLEDGE

In subsection V-D, we explain the significance of the intersection between the source domain and target domain models for the overall prediction of our average BLER. Additionally, we examine the effect of transferred knowledge, specifically non-trainable parameters, on prediction accuracy.

The impact of the size of the intersection between source domain and target domain can be seen in Fig. 11. Some part of source domain dataset is used for target domain dataset, i.e., $\mathcal{D}_s \neq \mathcal{D}_t$ and $\mathcal{D}_s \cap \mathcal{D}_t \neq \emptyset$. It is noteworthy that increasing the mutuality from 10% to 60% enhances the average e2e BLER predictions with the fine-tuned target model, with 50% overlap yielding notably accurate e2e BLER $\bar{e}_{e2e}$.

In Fig. 12, the size of the non-trainable parameter implies the amount of transferred knowledge. The more the number of non-trainable parameters, the greater the transferred knowledge and hence, the better the performance of fine-tuned target model. Specifically, at 192,396 shared

TABLE 4. Test performance comparison (Source vs. NFT Target Model vs. Fine-tuned-based Transfer Learning Target Model).

Models	Trainable Parameters	Non-trainable Parameters	Total Parameters	Test RMSE
Source	731,457	64	731,521	0.0925
NFT	64,993	0	64,993	0.2040
FT	17.409	731,488	748,897	0.0791

parameters, we obtain a high prediction accuracy for the fine-tuned model. This behavior is due to the domain adaptation principle, where increasing shared knowledge, the more the target model can learn from the source, resulting in better prediction accuracy as it more effectively models the target domain's characteristics.

VI. CONCLUSION

This paper presents a performance analysis of an ultra-reliable low-latency communication system for clustered networks with short packet communication in the finite blocklength regime. The system utilizes a cluster head as a wireless relay to achieve ultra-reliable and low-latency communications between a base station and ground users in the presence of multiple interferers. The paper derives a closed-form expression for the overall block error rate of the system based on a theoretical model that considers the packet size, blocklength, and maximum achievable rate as the main factors affecting the system's performance. Additionally, the paper proposes a transfer learning approach that uses fine-tuned models with domain adaptation for real-time prediction of the system's performance. This approach has the potential to improve the prediction accuracy and facilitate the deployment of URLLC systems in practical applications where real-time performance prediction is crucial.

APPENDIX A
PROOF OF LEMMA 1

In order to evaluate the PDF of I_C, let $\gamma_{I_i C} = \bar{\gamma}_{I_i C} \ell_{I_i C} |h_{I_i C}|^2$. Since $|h_{I_i C}|^2$ follows a gamma distribution, the PDF and CDF of $\gamma_{I_i C}$ are obtained as

$$f_{\gamma_{I_i C}}(\gamma) = \frac{1}{(m_{I_i C} - 1)!} \left(\frac{m_{I_i C}}{\bar{\gamma}_{I_i} \ell_{I_i C}} \right)^{m_{I_i C}}$$
$$\times \gamma^{m_{I_i C} - 1} \exp\left(-\frac{m_{I_i C}}{\bar{\gamma}_{I_i} \ell_{I_i C}} \gamma \right), \quad (67)$$

$$F_{\gamma_{I_i C}}(\gamma) = 1 - \exp\left(-\frac{m_{I_i C}}{\bar{\gamma}_{I_i} \ell_{I_i C}} \gamma \right)^{m_{I_i D_k} - 1} \frac{1}{m!} \left(\frac{m_{I_i C}}{\bar{\gamma}_{I_i} \ell_{I_i C}} \gamma \right)^m, \quad (68)$$

where $\gamma > 0$. The characteristic function (c.f.) of $\gamma_{I_i C}$ is given by

$$\Phi_{\gamma_{I_i C}}(j\omega) = \mathbb{E}\left\{ e^{j\omega \gamma_{I_i C}} \right\}$$
$$= \frac{\left(\frac{m_{I_i C}}{\bar{\gamma}_{I_i} \ell_{I_i C}} \right)^{m_{I_i C}}}{(m_{I_i D_k} - 1)!} \int_0^\infty \gamma^{m_{I_i C} - 1} e^{-\left(\frac{m_{I_i C}}{\bar{\gamma}_{I_i} \ell_{I_i C}} - j\omega \right)\gamma} \quad (69)$$

Using the identity [Eq. (3.381.3), R1], the c.f. of $\gamma_{I_i D_k}$ can be derived in closed-form as $\Phi_{\gamma_{I_i C}} = \left(1 - j\omega/a_{I_i C} \right)^{-m_{I_i C}}$. Hence, the c.f. of I_C, denoted as $\Phi_{I_C}(j\omega) = \mathbb{E}\left\{ e^{j\omega I_C} \right\}$, is given by

$$\Phi_{I_C}(j\omega) = \mathbb{E}\left\{ \exp\left(j\omega \sum_{i \in \mathcal{I}} \gamma_{I_i C} \right) \right\}$$
$$= \prod_{i \in \mathcal{I}} \mathbb{E}\left\{ e^{j\omega \gamma_{I_i C}} \right\}$$
$$= \prod_{i \in \mathcal{I}} \Phi_{\gamma_{I_i C}}(j\omega)$$
$$= \prod_{i \in \mathcal{I}} \left(1 - j\omega/a_{I_{\langle i \rangle} C} \right)^{-m_{I_{\langle i \rangle} C}}, \quad (70)$$

where $\gamma_{I_i C}$ for $i \in \mathcal{I}$ are mutually independent.

Consider an $n \times n$ matrix $\boldsymbol{\Lambda}$ with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ in any order. Let $\varrho(\boldsymbol{\Lambda})$ be the number of distinct eigenvalues, $\lambda_{\langle 1 \rangle}, \lambda_{\langle 2 \rangle}, \ldots, \lambda_{\langle \varrho(\boldsymbol{\Lambda}) \rangle}$ be the distinct eigenvalues in decreasing order, and $\tau_i(\boldsymbol{\Lambda})$ be the multiplicity of $\lambda_{\langle i \rangle}$. Then, the $(i, j)^{th}$ characteristics coefficient $\chi_{i,j}(\boldsymbol{\Lambda})$, $i = 1, 2, \ldots, \varrho(\boldsymbol{\Lambda})$, $j = 1, 2, \ldots, \tau_i(\boldsymbol{\Lambda})$, is defined as a partial fraction expansion coefficient of $\det(I_n + \xi\boldsymbol{\Lambda})^{-1}$ such that

$$\det(I_n + \xi\boldsymbol{\Lambda})^{-1} = \prod_{i=1}^{\varrho(\boldsymbol{\Lambda})} \left(1 + \xi\lambda_{\langle i \rangle} \right)^{-\tau_i(\boldsymbol{\Lambda})}$$
$$= \sum_{i=1}^{\varrho(\boldsymbol{\Lambda})} \sum_{j=1}^{\tau_i(\boldsymbol{\Lambda})} \chi_{i,j}(\boldsymbol{\Lambda}) \left(1 + \xi\lambda_{\langle i \rangle} \right)^{-j}, \quad (71)$$

where ξ is a scalar constant such that $(I_n + \xi\boldsymbol{\Lambda})$ is non singular and $\chi_{i,j}(\boldsymbol{\Lambda})$ can be determined by

$$\chi_{i,j}(\boldsymbol{\Lambda})$$
$$= \frac{1}{(\tau_i(\boldsymbol{\Lambda}) - j)! \lambda_{\langle i \rangle}^{\tau_i(\boldsymbol{\Lambda}) - j}}$$
$$\times \left[\frac{d^{\tau_i(\boldsymbol{\Lambda}) - j}}{d\xi^{\tau_i(\boldsymbol{\Lambda}) - j}} (1 + \xi\lambda_{\langle i \rangle})^{\tau_i(\boldsymbol{\Lambda})} \det(\mathbf{I}_n + \xi\boldsymbol{\Lambda})^{-1} \right]\Bigg|_{s \to -\frac{1}{\lambda_{\langle i \rangle}}}, \quad (72)$$

where $\sum_{i=1}^{\varrho(\boldsymbol{\Lambda})} \sum_{j=1}^{\tau_i(\boldsymbol{\Lambda})} \chi_{i,j}(\boldsymbol{\Lambda}) = 1$. Using a partial fraction decomposition of $\Phi_{I_C}(j\omega)$ we obtain

$$\Phi_{I_C}(j\omega) = \sum_{i=1}^{\varrho(\boldsymbol{\mathcal{A}})} \sum_{j=1}^{\tau_i(\boldsymbol{\mathcal{A}})} \chi_{i,j}(\boldsymbol{\mathcal{A}}) \left(1 + \frac{j\omega}{a_{I_{\langle i \rangle} C}} \right)^{-j}. \quad (73)$$

The PDF of I_C, defined as $f_{I_C}(x) = \frac{1}{2\pi} \int_{-\infty}^\infty e^{-j\omega x} \Phi_{I_C}(j\omega) \, d\omega$, can be written as,

$$f_{I_C}(x) = \sum_{i=1}^{\varrho(\boldsymbol{\mathcal{A}})} \sum_{j=1}^{\tau_i(\boldsymbol{\mathcal{A}})} \chi_{i,j}(\boldsymbol{\mathcal{A}}) \frac{1}{2\pi} \int_{-\infty}^\infty e^{j\omega x} \left(1 + j\omega/a_{I_{\langle i \rangle} C} \right)^{-j} d\omega. \quad (74)$$

Hence we achieve the final expression of the PDF of I_C given in (5). Now, the CDF of I_C, defined as $F_{I_C}(x) = \int_0^x f_{I_C}(t)dt$, is obtained as

$$F_{I_C}(x) = 1 - \sum_{i=1}^{\varrho(\boldsymbol{\mathcal{A}})} \sum_{j=1}^{\tau_i(\boldsymbol{\mathcal{A}})} \chi_{i,j}(\boldsymbol{\mathcal{A}}) \frac{a_{I_{\langle i \rangle} C}^j}{(j-1)!} \int_x^\infty t^{j-1} e^{-t a_{I_{\langle i \rangle} C}} dt. \quad (75)$$

Using [47, Eq. (aaa)], we obtain (6). This completes the proof of Lemma 1.

APPENDIX B
PROOF OF LEMMA 3

In order to evaluate the PDF of I_{D_k}, let $\gamma_{I_iD_k} = \bar{\gamma}_{I_iD_k} \ell_{I_iD_k} |h_{I_iD_k}|^2$, since $|h_{I_iD_k}|^2$ follows a Gamma distribution, the PDF and CDF of $\gamma_{I_iD_k}$ are obtained as

$$f_{\gamma_{I_iD_k}}(\gamma) = \frac{(a_{I_iD_k})^{m_{I_iD_k}}}{(m_{I_iD_k}-1)!} \gamma^{m_{I_iD_k}-1} e^{-a_{I_iD_k}\gamma},$$

$$F_{\gamma_{I_iD_k}}(\gamma) = 1 - e^{-a_{I_iD_k}\gamma} \sum_{m=0}^{m_{I_iD_k}-1} \frac{(a_{I_iD_k})^m}{m!}, \qquad (76)$$

where $\gamma > 0$. Now, following the steps used in Lemma 1, the c.f. of I_{D_k} is derived as

$$\Phi_{I_{D_k}}(j\omega) = \sum_{i=1}^{\varrho(\mathcal{B}_k)} \sum_{j=1}^{\tau_i(\mathcal{B}_k)} \chi_{i,j}(\mathcal{B}_k)\left(1 + j\omega\mu_{I_{(i)}D_k}\right)^{-j}. \qquad (77)$$

Hence, the PDF of I_{D_k}, defined as $f_{I_{D_k}}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-j\omega x} \Phi_{I_{D_k}}(j\omega)d\omega$, and the CDF of I_{D_k}, defined as $F_{I_{D_k}}(x) = \int_0^x f_{I_{D_k}}dt$, are obtained as

$$f_{I_{D_k}}(x) = \sum_{i=1}^{\varrho(\mathcal{B}_k)} \sum_{j=1}^{\tau_i(\mathcal{B}_k)} \chi_{i,j}(\mathcal{B}_k)\frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega x}$$
$$\times \left(1 + j\omega\mu_{I_{(i)}D_k}\right)^{-j}d\omega, \qquad (78)$$

$$F_{I_{D_k}}(x) = 1 - \sum_{i=1}^{\varrho(\mathcal{B}_k)} \sum_{j=1}^{\tau_i(\mathcal{B}_k)} \frac{\chi_{i,j}(\mathcal{B}_k)}{(j-1)!}\Gamma(j, \mu_{I_{(i)}D_k}), \qquad (79)$$

respectively. After some mathematical manipulations, we obtain (15) and (16). This completes the proof of Lemma 3.

APPENDIX C
PROOF OF THEOREM 1

From (28), we have $Z(\Gamma_{SC}) \approx \frac{1}{2} - \tau(\Gamma_{SC} - \omega)$ for $\psi < \Gamma_{SC} < \eta$. Hence, the average BLER for the S-to-C, link i.e., $\bar{\epsilon}_{SC}$, can be written as

$$\bar{\epsilon}_{SC} \approx \int_0^{\infty} Z(\Gamma_{SC})f_{\Gamma_{SC}}(x)dx \overset{(a)}{=} \tau \int_{\psi}^{\eta} F_{\Gamma_{SC}}(x)dx, \qquad (80)$$

where (a) is due to the partial integration method. Plugging (11) into (80), we obtain

$$\bar{\epsilon}_{SC} = 1 - \tau \sum_{m=0}^{Nm_{SC}-1} \sum_{i=1}^{\varrho(\mathcal{A})} \sum_{j=1}^{\tau_i(\mathcal{A})} \frac{\chi_{i,j}(\mathcal{A})}{m!} \frac{a_{I_{(i)}C}}{(j-1)!}$$
$$\int_0^{\infty} x^{j-1}e^{-a_{I_{(i)}C}x}dx \int_{\psi}^{\eta} (a_{SC}\gamma)^m e^{-a_{SC}(x+1)\gamma}d\gamma. \qquad (81)$$

It is noted that the second integral can be derived in closed-form expression as

$$\int_{\psi}^{\eta} (\gamma)^m e^{-a_{SC}(x+1)\gamma}d\gamma$$

$$= \sum_{r=0}^{m} \frac{m!}{r!}(a_{SC}(x+1))^{r-1}$$
$$\left[\psi^r e^{-a_{SC}(x+1)\psi} - \eta^r e^{-a_{SC}(x+1)\eta}\right]. \qquad (82)$$

Hence, substituting the above results and after some mathematical manipulations, we obtain (30). This completes the proof of Theorem 1.

APPENDIX D
PROOF OF THEOREM 2

From (28), we have $Z(\Gamma_{CD_k}) \approx \frac{1}{2} - \tau(\Gamma_{CD_k} - \omega)$ for $\psi < \Gamma_{CD_k} < \eta$. Hence, the average BLER for the C-to-D$_k$ link, i.e., $\bar{\epsilon}_{CD_k}$, is derived as $\bar{\epsilon}_{CD_k} \approx \tau \int_{\psi}^{\eta} F_{\Gamma_{CD_k}}(x)dx$.

Now, from (17), we obtain

$$\bar{\epsilon}_{CD_k} = 1 - \tau \sum_{m=0}^{Nm_{CD_k}-1} \sum_{i_k=1}^{\varrho(\mathcal{B}_k)} \sum_{j_k=1}^{\tau_{ik}(\mathcal{B}_k)} \frac{\chi_{i_k,j_k}(\mathcal{B}_k)}{m_k!} \frac{d_{I_{(i_k)}C}^{j_k}}{(j_k-1)!}$$
$$\int_0^{\infty} x^{j_k-1}e^{-a_{I_{(i_k)}C}x}dx \int_{\psi}^{\eta} (a_{CD_k}\gamma)^{m_k}$$
$$\times e^{-a_{CD_k}(x+1)\gamma}d\gamma. \qquad (83)$$

where the second integral can be derived in closed-form as

$$\int_{\psi}^{\eta} (\gamma)^{m_k} e^{-a_{CD_k}(x+1)\gamma}d\gamma$$

$$= \sum_{r_k=0}^{m} \frac{m_k!}{r_k!} \frac{\psi^{r_k}e^{-a_{CD_k}(x+1)\psi} - \eta^{r_k}e^{-a_{CD_k}(x+1)\eta}}{(a_{CD_k}(x+1))^{1-r_k}}. \qquad (84)$$

Hence, substituting the above results and with the help of the Confluent hypergeometric function of the second kind, we obtain (32). This completes the proof of Theorem 2.

APPENDIX E
PROOF OF THEOREM 3

The average e2e BLER of the system can be approximated as $\bar{\epsilon}_{e2e} \approx \tau \int_{\psi}^{\eta} F_{\Gamma_{sys}}(\gamma)d\gamma$. Substituting $F_{\Gamma_{sys}}$ from (19), we obtain

$$\bar{\epsilon}_{e2e} \approx 1 - \tau \int_{\psi}^{\eta} 1 - \left[1 - F_{\Gamma_{SC}}(\gamma)\right] \prod_{k\in\mathcal{K}}\left[1 - F_{\Gamma_{CD_k}}(\gamma)\right]d\gamma. \qquad (85)$$

Substituting the CDF of Γ_{SC} and Γ_{CD_k} and after some mathematical manipulations, we obtain $\bar{\epsilon}_{e2e}$ expressed in terms of $\Lambda_2 + \Lambda_3$, where

$$\Lambda_2 + \Lambda_3$$
$$= \int_{\psi}^{\eta} \frac{e^{\gamma(a_{SC}+\sum_{k=1}^{K}a_{CD_k})}\gamma^{m+\sum_{k=1}^{K}m_k}}{(a_{SC}\gamma + a_{I_{(i)}C})^{r+j}\prod_{k=1}^{K}(a_{CD_k}\gamma + a_{I_{(i_k)}D_k})^{r_k+j_k}}d\gamma. \qquad (86)$$

Performing partial fraction decomposition yields Λ_2 and Λ_3 as shown in (36) and (38), where

$$\Lambda(x, a, b, c)$$
$$= \int_0^x \gamma^a e^{-b\gamma}(1 + c\gamma)^{-q}d\gamma$$

$$= \frac{1}{(q-1)!} \int_0^x \gamma^a G_{1,1}^{1,1} \left[{\scriptstyle 1-q \atop 0} \Big| c\gamma \right] G_{0,1}^{1,0} \left[{\scriptstyle - \atop 1} \Big| b\gamma \right] d\gamma. \quad (87)$$

We can rewrite $\Lambda(x, a, b, c)$ in terms of bivariate Meijiers's G-function as

$$G_{A,[C,E],B,[D,F]}^{p,q,s,r,t} \left[{x \atop y} \Bigg| {\mathbf{a}_p, \mathbf{a}_a \atop \mathbf{c}_q, \mathbf{c}_c; \mathbf{e}_s, \mathbf{e}_E \atop \mathbf{b}_B \atop \mathbf{d}_r, \mathbf{d}_D; \mathbf{f}_t, \mathbf{f}_F} \right] = \frac{1}{(2\pi i)^2} \int_{L_1} \int_{L_2}$$
$$\times \frac{\Gamma(1 - \mathbf{a}_p + p + \eta)\Gamma(\mathbf{c}_q + p)}{\Gamma(1 - \mathbf{c}_C - p)\Gamma(1 - \mathbf{d}_D + p)\Gamma(1 - \mathbf{e}_E - \eta)}$$
$$\times \frac{\Gamma(\mathbf{d}_r - p)\Gamma(\mathbf{e}_s + \eta)\Gamma(\mathbf{f}_t - \eta)}{\Gamma(1 - \mathbf{f}_F + \eta)\Gamma(\mathbf{a}_A - p - \eta)\Gamma(\mathbf{b}_B + p + \eta)} x^p y^\eta dp d\eta, \quad (88)$$

where $G[\cdot]$ is the Meijer G-function and $\Gamma(\cdot)$ denotes the Gamma function. The following identity holds

$$\int_0^x t^{\nu-1} G_{p,q}^{m,n} \left[{\boldsymbol{\alpha}_n, \boldsymbol{\alpha}_p \atop \boldsymbol{\beta}_m, \boldsymbol{\beta}_q} \Big| \mu t \right] G_{P,Q}^{M,N} \left[{\boldsymbol{\gamma}_N, \boldsymbol{\gamma}_P \atop \boldsymbol{\delta}_M, \boldsymbol{\delta}_Q} \Big| \omega t \right] dt = x^\nu$$
$$G_{1,[n,N],1,[p+q-n,P+Q-N]}^{1,n,N,m,M}$$
$$\times \left[{\mu x \atop \omega x} \Bigg| {1 - \nu \atop 1 - \boldsymbol{\alpha}_n; 1 - \boldsymbol{\gamma}_N \atop \nu + 1 \atop \boldsymbol{\beta}_m, \boldsymbol{\beta}_q, 1 - \boldsymbol{\alpha}_p; \boldsymbol{\delta}_M, \boldsymbol{\delta}_Q, 1 - \boldsymbol{\gamma}_P} \right]. \quad (89)$$

Hence, $\bar{\epsilon}_{e2e}$ can be expressed in terms of Meijer-G function of two variables [60], as given in (34).

REFERENCES

[1] M. Vaezi, A. Azari, S. R. Khosravirad, M. Shirvanimoghaddam, M. M. Azari, D. Chasaki, and P. Popovski, "Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 1117–1174, 2nd Quart., 2022.

[2] A. Aijaz, S. Zeadally, M. Shah, and A. Vasilakos, "Short packet communication in 5G: Challenges and research directions," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 186–192, Sep. 2018.

[3] S. Zhang, X. Chen, Y. Li, D. Jiang, Q. Zhang, and M. Guizani, "Advancements in short packet communication for 6G networks," *Wireless Commun.*, vol. 29, no. 1, pp. 136–143, 2022.

[4] Y. Prathyusha and T.-L. Sheu, "Coordinated resource allocations for eMBB and URLLC in 5G communication networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8717–8728, Aug. 2022.

[5] R. Chen, C. Li, S. Yan, R. Malaney, and J. Yuan, "Physical layer security for ultra-reliable and low-latency communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 6–11, Oct. 2019.

[6] Y. Ju, H. Wang, Q. Pei, and H.-M. Wang, "Physical layer security in millimeter wave DF relay systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 12, pp. 5719–5733, Dec. 2019.

[7] H. Chen, R. Abbas, P. Cheng, M. Shirvanimoghaddam, W. Hardjawana, W. Bao, Y. Li, and B. Vucetic, "Ultra-reliable low latency cellular networks: Use cases, challenges and approaches," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 119–125, Dec. 2018.

[8] S. Pala, M. Katwe, K. Singh, B. Clerckx, and C.-P. Li, "Spectral-efficient RIS-aided RSMA URLLC: Toward mobile broadband reliable low latency communication (mBRLLC) system," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3507–3524, Apr. 2024.

[9] J. Chen, L. Zhang, Y.-C. Liang, X. Kang, and R. Zhang, "Resource allocation for wireless-powered IoT networks with short packet communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1447–1461, Feb. 2019.

[10] M. Tao, "Effects of non-identical Rayleigh fading on differential unitary space-time modulation," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 1359–1369, May 2009.

[11] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[12] L. Xu, X. Zhou, Y. Tao, X. Yu, M. Yu, and F. Khan, "AF relaying secrecy performance prediction for 6G mobile communication networks in Industry 5.0," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5485–5493, Aug. 2022.

[13] A. E. Kalør, G. Durisi, S. Coleri, S. Parkvall, W. Yu, A. Mueller, and P. Popovski, "Wireless 6G connectivity for massive number of devices and critical services," 2024, *arXiv:2401.01127*.

[14] L. Xu, J. Wang, X. Li, F. Cai, Y. Tao, and T. A. Gulliver, "Performance analysis and prediction for mobile Internet-of-Things (IoT) networks: A CNN approach," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13355–13366, Sep. 2021.

[15] T. K. Oikonomou, N. G. Evgenidis, D. G. Nixarlidis, D. Tyrovolas, S. A. Tegos, P. D. Diamantoulakis, P. G. Sarigiannidis, and G. K. Karagiannidis, "CNN-based automatic modulation classification under phase imperfections," *IEEE Wireless Commun. Lett.*, vol. 13, no. 5, pp. 1508–1512, May 2024.

[16] Y. Liu, Y. Deng, A. Nallanathan, and J. Yuan, "Machine learning for 6G enhanced ultra-reliable and low-latency services," *IEEE Wireless Commun.*, vol. 30, no. 2, pp. 48–54, Apr. 2023.

[17] K. Glinskiy, A. Krasilov, E. Khorov, and A. Kureev, "Performance of ML-based channel prediction algorithms for URLLC: Channel model matters," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, Jul. 2023, pp. 306–311.

[18] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato, O. Dobre, and H. V. Poor, "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022.

[19] Y. Cai, W. Li, X. Meng, W. Zheng, C. Chen, and Z. Liang, "Adaptive contrastive learning based network latency prediction in 5G URLLC scenarios," *Comput. Netw.*, vol. 240, Feb. 2024, Art. no. 110185.

[20] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, Jan. 2020.

[21] Z. Yue, "Ts2vec: Towards universal representation of time series," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 8, 2022, pp. 8980–8987.

[22] H. Wu, J. Xu, J. Wang, and M. Long, "AutoFormer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 22419–22430.

[23] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, May 2021, pp. 11106–11115.

[24] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[25] S. Hussain Bouk, B. Omoniwa, and S. Shetty, "Predicting downlink retransmissions in 5G networks using deep learning," in *Proc. IEEE 21st Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2024, pp. 1056–1057.

[26] N. Hoang Tu, T.-D. Hoang, and K. Lee, "Short-packet URLLCs for MIMO underlay cognitive multihop relaying with imperfect CSI," *IEEE Access*, vol. 11, pp. 81672–81689, 2023.

[27] T.-V. Nguyen, T. Huynh-The, and V. Nguyen Quoc Bao, "Performance analysis and deep learning evaluation of URLLC full-duplex energy harvesting IoT networks over Nakagami-m fading channels," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Jul. 2023, pp. 309–313.

[28] C. Liu, S. Li, W. Yuan, X. Liu, and D. Wing Kwan Ng, "Predictive precoder design for OTFS-enabled URLLC: A deep learning approach," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 7, pp. 2245–2260, Jul. 2023.

[29] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Trans. Artif. Intell.*, vol. 1, no. 2, pp. 151–166, Oct. 2020.

[30] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4396–4415, Apr. 2023.

[31] M. Wang, Y. Lin, Q. Tian, and G. Si, "Transfer learning promotes 6G wireless communications: Recent advances and future challenges," *IEEE Trans. Rel.*, vol. 70, no. 2, pp. 790–807, Jun. 2021.

[32] R. W. L. Coutinho and A. Boukerche, "Transfer learning for disruptive 5G-enabled industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4000–4007, Jun. 2022.

[33] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[34] X. Liu, W. Yu, F. Liang, D. Griffith, and N. Golmie, "Toward deep transfer learning in industrial Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12163–12175, Aug. 2021.

[35] Q. Peng, A. Gilman, N. Vasconcelos, P. C. Cosman, and L. B. Milstein, "Robust deep sensing through transfer learning in cognitive radio," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 38–41, Jan. 2020.

[36] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *China Commun.*, vol. 17, no. 2, pp. 138–148, Feb. 2020.

[37] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[39] Y. Prathyusha and T.-L. Sheu, "Resource allocations for coexisting eMBB and URLLC services in multi-UAV aided communication networks for cellular offloading," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 6658–6671, May 2024.

[40] *Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects (Release 9)*, document 3GPP-TR-36.814 V9.0.0, Mar. 2010.

[41] P. Z. Peebles, *Probability, Random Variables and Random Signal Principles*, 4th ed. New York, NY, USA: McGraw-Hill, 2000.

[42] A. Rao and M.-S. Alouini, "Performance of cooperative spectrum sensing over non-identical fading environments," *IEEE Trans. Commun.*, vol. 59, no. 12, pp. 3249–3253, Dec. 2011.

[43] T. Nechiporenko, K. T. Phan, C. Tellambura, and H. H. Nguyen, "On the capacity of Rayleigh fading cooperative systems under adaptive transmission," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1626–1631, Apr. 2009.

[44] J. Zhang, H. Du, Q. Sun, B. Ai, and D. W. K. Ng, "Physical layer security enhancement with reconfigurable intelligent surface-aided networks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3480–3495, 2021.

[45] Y. Ai, F. A. P. deFigueiredo, L. Kong, M. Cheffena, S. Chatzinotas, and B. Ottersten, "Secure vehicular communications through reconfigurable intelligent surfaces," *IEEE Trans. Veh. Technol.*, vol. 70, no. 7, pp. 7272–7276, Jul. 2021.

[46] H. Ibrahim, H. Tabassum, and U. T. Nguyen, "Exact coverage analysis of intelligent reflecting surfaces with Nakagami-*m* channels," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 1072–1076, Jan. 2021.

[47] I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products*, 7th ed. New York, NY, USA: Academic, 2007.

[48] H. B. Celebi, A. Pitarokoilis, and M. Skoglund, "A multi-objective optimization framework for URLLC with decoding complexity constraints," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2786–2798, Apr. 2022.

[49] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[50] A. Agarwal, A. K. Jagannatham, and L. Hanzo, "Finite blocklength non-orthogonal cooperative communication relying on SWIPT-enabled energy harvesting relays," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3326–3341, Jun. 2020.

[51] N. H. Tu and K. Lee, "Performance analysis and optimization of multihop MIMO relay networks in short-packet communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4549–4562, Jun. 2022.

[52] T. Li, H. Zhang, J. Qiao, and J. Tian, "Robust beamforming design with finite blocklength for URLLC," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 2604–2608, Feb. 2023.

[53] F. Kara and H. Kaya, "Error probability analysis of NOMA-based diamond relaying network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2280–2285, Feb. 2020.

[54] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.

[55] T. N. Do, G. Kaddoum, T. L. Nguyen, D. B. da Costa, and Z. J. Haas, "Aerial reconfigurable intelligent surface-aided wireless communication systems," 2021, *arXiv:2106.05380*.

[56] B. Vu Minh, A.-T. Le, C.-B. Le, S. Q. Nguyen, V.-D. Phan, T. N. Nguyen, and M. Voznak, "Performance prediction in UAV-terrestrial networks with hardware noise," *IEEE Access*, vol. 11, pp. 117562–117575, 2023.

[57] S. Chatterjee and J. S. Simonoff, *Handbook of Regression Analysis*. Hoboken, NJ, USA: Wiley, 2013.

[58] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[59] L. Yuan, N. Yang, F. Fang, and Z. Ding, "Performance analysis of UAV-assisted short-packet cooperative communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4471–4476, Apr. 2022.

[60] A. M. Mathai and R. K. Saxena, *The H-Function With Applications in Statistics and Other Disciplines*. Hoboken, NJ, USA: Wiley, 1978.

MUJTABA GHOUS received the B.S. degree in electrical engineering from the University of Engineering and Technology (UET), Taxila, Pakistan, in 2014, and the M.S. degree in electrical engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2017. He is currently pursuing the Ph.D. degree with the Resilient Machine Learning Institute (ReMI), ÉTS, Montreal, Canada. He is a member of the ReMI, ÉTS. His research interests include machine learning and wireless communications.

THANH LUAN NGUYEN (Graduate Student Member, IEEE) received the B.S. degree (Hons.) from Ho Chi Minh City University of Technology and Education, Vietnam, in 2016. He is currently pursuing the degree with the École de Technologie Supérieure (ÉTS), Université du Québec, Montreal, Canada. He has authored and co-authored over ten ISI-indexed journals with over 200 citations. His research interests include wireless communication channels, energy harvesting, reconfigurable intelligent surfaces, and other emerging wireless technologies.

TRI NHU DO (Member, IEEE) was born in Da Nang, Vietnam. He received the bachelor's degree in electronics and telecommunications engineering from the Posts and Telecommunications Institute of Technology, Vietnam, in 2012, and the master's and Ph.D. degrees in electronics and computer engineering from Hongik University, South Korea, in 2015 and 2018, respectively. From 2017 to 2018, he was a Teaching Associate with the Department of Software and Communications Engineering, Hongik University. In 2019, he held the position of Research Associate with the Department of Computer Science, The University of Texas at Dallas, TX, USA. He was a Postdoctoral Research Fellow with the Resilient Machine Learning Institute (ReMI), École de Technologie Supérieure (ÉTS), Université du Québec, Montreal, QC, Canada, from 2020 to September 2023. He is currently a Tenure-Track Assistant Professor with the Electrical Engineering Department, Polytechnique Montréal, Montreal. His primary research interests include wireless communications and AI/ML.

GEORGES KADDOUM (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the École Nationale Supérieure de Techniques Avancées (ENSTA Bretagne), Brest, France, the M.S. degree in telecommunications and signal processing(circuits, systems, and signal processing) from the Université de Bretagne Occidentale and Telecom Bretagne (ENSTB), Brest, in 2005, and the Ph.D. degree (Hons.) in signal processing and telecommunications from the National Institute of Applied Sciences (INSA), University of Toulouse, Toulouse, France, in 2009. Since 2010, he has been a Scientific Consultant in the field of space and wireless telecommunications for several U.S. and Canadian companies. In 2014, he was awarded the ÉTS Research Chair in physical-layer security for wireless networks. He is currently a Professor and a Tier 2 Canada Research Chair with the École de Technologie Supérieure (ÉTS), Université du Québec, Montreal, QC, Canada, and a Faculty Fellow with the Cyber Security Systems and Applied AI Research Center, Lebanese American University. He has published over 300 journal articles, conference papers, two chapters in books, and has eight pending patents. His recent research activities cover wireless communication networks, tactical communications, resource allocations, and security. He received the Best Papers Award from the 2014 IEEE International Conference on Wireless and Mobile Computing, Networking, Communications (WIMOB), with three coauthors, and from the 2017 IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), with four coauthors. Moreover, he received IEEE Transactions on Communications Exemplary Reviewer Award, in 2015, 2017, and 2019. In addition, he received the Research Excellence Award from the Université du Québec, in 2018. In 2019, he received the Research Excellence Award from ÉTS in recognition of his outstanding research outcomes. He is serving as an Area Editor for IEEE Transactions on Machine Learning in Communications and Networking and an Associate Editor for IEEE Transactions on Information Forensics and Security, IEEE Transactions on Communications, and IEEE Communications Letters.

○ ○ ○