

Article

Increasing 3D Printing Accuracy Through Convolutional Neural Network-Based Compensation for Geometric Deviations

Moustapha Jadayel  and Farbod Khameneifar * 

Department of Mechanical Engineering, Polytechnique Montréal, Montreal, QC H3T 1J4, Canada; moustapha.jadayel@polymtl.ca

* Correspondence: farbod.khameneifar@polymtl.ca

Abstract: As Additive Manufacturing (AM) evolves from prototyping to full-scale production, improving geometric accuracy becomes increasingly critical, especially for applications requiring high dimensional fidelity. This study proposes a machine learning-based approach to enhance the geometric accuracy of 3D printed parts produced by Fused Filament Fabrication (FFF), a widely used material extrusion process in which thermoplastic filament is heated and deposited layer by layer to form a part. Our method relies on a Convolutional Neural Network (CNN) trained to predict a systematic deviation field based on 3D scan data of a sacrificial print. These scans are acquired using a structured light 3D scanner, which provides detailed surface information on geometric deviations that arise during the printing process. The predicted deviation field is then inverted and applied to the digital model to generate a compensated geometry, which, when printed, offsets the errors observed in the original part. Experimental validation using a complex reference geometry shows that the proposed compensation method achieves an 88.5% reduction in mean absolute geometric deviation compared to the uncompensated print. This significant improvement underscores the CNN's ability to generalize across geometric features and capture systematic deformation patterns inherent to FFF. The results demonstrate the potential of combining 3D scanning and deep learning to enable adaptive, data-driven compensation strategies in AM. The method proposed in this paper contributes to reducing trial-and-error iterations, improving part quality, and facilitating the broader adoption of FFF for precision-demanding industrial applications.

Keywords: additive manufacturing; fused filament fabrication; convolutional neural network; geometric accuracy; 3D printing; geometric compensation; systematic deviations



Academic Editors: Kai Cheng, Angelos P. Markopoulos, Emmanouil-Lazaros Papazoglou and Panagiotis Karmiris-Obratański

Received: 1 April 2025

Revised: 24 April 2025

Accepted: 28 April 2025

Published: 1 May 2025

Citation: Jadayel, M.; Khameneifar, F. Increasing 3D Printing Accuracy Through Convolutional Neural Network-Based Compensation for Geometric Deviations. *Machines* **2025**, *13*, 382. <https://doi.org/10.3390/machines13050382>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fused Filament Fabrication (FFF), often referred to as Fused Deposition Modelling (FDM), involves the layer-by-layer extrusion of thermoplastic filament to create three-dimensional objects [1], making it an easily adoptable and accessible form of additive manufacturing. An evident advantage of the FFF process lies in its capacity to generate intricate geometries with printed support structures, eliminating the necessity for individual setups for each unique part. Such attributes render FFF particularly suitable for rapid prototyping or small-scale production scenarios where geometric accuracy is not the primary concern, and lead time is paramount [2]. Lately, FFF has increasingly found applications in larger-scale manufacturing environments. The accessibility and adaptability of AM machines have transformed them into indispensable tools within factory settings, where their capacity to produce intricate geometries on-demand is invaluable [3,4]. This

evolution reflects the shifting role of FFF and analogous AM techniques from temporary prototyping solutions to permanent components within the manufacturing landscape, offering novel avenues for innovation and efficiency.

To support this evolution, researchers have investigated the geometric accuracy of additively manufactured parts to assess the reliability of AM machines in producing geometrically correct parts. Additionally, efforts have been made to develop methodologies to improve the geometric accuracy of 3D-printed parts, enhancing FFF technology's overall capabilities and applicability. One method to indirectly improve the geometric accuracy of a 3D printed part is by dynamically adjusting the 3D printing process parameters, such as the deposition rate, movement speed, and nozzle temperature, in a closed feedback loop. By improving the stability of process parameters, the resulting printed part should remain close to its specified geometric accuracy, thus reducing non-repeatable errors. Dynamic parameter adjustments can additionally affect the mechanical capabilities of a manufactured part by ensuring that the optimal printing parameters are consistently maintained, thus resulting in more reliable mechanical properties. Ye et al. [5] proposed a deep forest approach to detect the effects of a process shift using the 3D scan measurement of a 3D printed part. Tamir et al. [6] used an in-situ camera to capture the newly printed layer and a combination of Machine Learning (ML) models and a fuzzy logic controller to adjust four printing parameters to remain at optimal values for the duration of the printing process. Brion and Pattinson [7] demonstrated a multi-head neural network, a neural network with multiple secondary neural networks with different objectives, able to determine deviations from optimal process parameters by analyzing images of deposited material in FFF 3D printers. Detecting parameter drift allowed automated process parameter adjustments to keep the printer working within expected conditions and improve geometric accuracy. Many other researchers have contributed to this field of study by analyzing different materials [8] or processes [9], by adding new parameters [10], or by implementing different methodologies to detect process parameter shifts [11]. Nath et al. [12] proposed a process design optimization framework to optimize FFF process parameters. The thermal history obtained from a simulation is used to predict the geometric inaccuracy of the printed part. The FFF process parameters are then optimized to minimize inaccuracy by affecting the thermal history obtained from the simulation.

Another exciting approach investigated by many researchers is to enhance the geometric accuracy of additively manufactured parts by analyzing and optimizing the 3D model before printing to identify predictable deviations and modify the print file accordingly, eliminating any errors or issues affecting the final print's geometric accuracy. Some researchers have proposed pre-compensating the 3D model of the desired part by running a Finite Element Analysis (FEA) of the printing process to predict the deviations caused by thermo-mechanical deformations caused by the printing process [13]. Chowdhury et al. [14,15] proposed the use of FEA simulations to build a training dataset for an Artificial Neural Network (ANN). The ANN would be faster and more generalizable to new situations without requiring the valuable time of experienced personnel to set up and run an FEA simulation. Francis and Bian [16] have demonstrated a neural network able to predict the distortion of a Laser-Based Additive Manufacturing machine by measuring the local heat transfer from the laser to the solidified part. This neural network retains a thermal history of the heat distribution and produces a prediction of deviation due to thermo-mechanical effects. Huang et al. [17] presented a method to improve the dimensional accuracy in Binder Jetting Additive Manufacturing (BJAM) by using an artificial neural network trained on FEA simulation results. This ANN predicts and compensates for shrinkage deformations occurring during the sintering process, significantly improving the dimensional accuracy of the final product.

Another method for enhancing the geometric accuracy of AM machines involves leveraging the geometric deviations observed in a sacrificial part to inform the fitting of an error model. Tong et al. [18] proposed a parametric error model the laser focal point positioning in a StereoLithography Apparatus (SLA) 3D printer. A kinematic path from the origin of the machine to the laser focal point is created by measuring the deviations of a printed part with a Coordinate Measuring Machine (CMM) to accumulate the rotational errors of predecessor axes to the current axis. Running axis errors are repeatable and are only related to the theoretical position of the laser focal point. In addition, a random error term is added to the parametric error model, representing the errors due to non-repeatable components. Parametric error analysis is further explored by Huang et al. [19], who computed a predictive statistical model best-fitting the deviation of an SLA 3D printed part. The statistical model is a closed-form equation with three components: the average deviation of the printed part, position-dependent deviations, and a high-frequency component added to the main trend. The authors also present the equation in polar form to clarify the relationship between the deviated and nominal points, reducing potential confusion. Both researchers aim to build a model that can fit the deviations produced by a 3D printer. The solution to the equation allows the authors to modify the CAD model accurately to overcome repeatable errors and minimize the impact of random errors in their model.

SLA 3D printers are often more accurate than FFF 3D printers [20] and can produce a more stable and repeatable dataset, ideal for parametric analyses such as those previously presented. On the other hand, FFF 3D printers have a more complex kinematic system due to the need to precisely coordinate multiple moving parts and manage filament extrusion, which introduces additional variables and potential sources of error and increases the complexity required for predictive statistical models. Wang et al. [21] produced a statistical prediction model based on Kriging spatial statistics [22], which is suitable due to the natural spatial correlations in the extruder positioning error. The prediction model is experimentally tested by modifying the CAD model by the inverse of the predicted deviation, demonstrating it as an effective in-plane deviation compensation method. Alternative geometric compensation schemes were proposed and compared experimentally by Cheng et al. [23], integrating process parameters into their statistical model to make it more generalizable.

To help determine the optimal compensation equation, an analytical framework was proposed by Huang [24] using quality measures, such as the Minimum Area Deviation and Minimum Volume Deviation. Then, to improve the geometric accuracy of freeform geometries, a generic and prescriptive methodology was established by Luan and Huang [25]. A convolution framework was also developed by Huang et al. [26] to predict 3D shape accuracy. This convolution model provides a layer-by-layer in-plane prediction of deformation, which can be used to compensate the initial geometry to eliminate the predicted deviations. Wang et al. proposed a formalized mathematical decomposition for fabricating non-smooth 3D printed parts, involving additive building of smooth base shapes followed by subtractive removal of extra material. Their model accounts for predicted deviations using location-dependent, process parameter, and shape-dependent covariates, with a zero-mean random field for spatial correlation and measurement error [27].

Machine learning for AM has been a new trend, and researchers have been applying machine learning tools to improve AM performance. Wang et al. [27] did so by creating a learning framework for shape deviation modeling and learning of 3D geometries in AM by decomposing the AM process in an additive and a subtractive step to unify smooth and non-smooth shapes. A Bayesian methodology was employed by Sabbaghi et al. [28] to model deviations in a polygon and straight edges of freeform objects based only on the deviation measured on pentagons and cylinder, showing the capacity of the author's

model to generalize based on a limited dataset. Sabbaghi and Huang [29] refined this methodology by developing a new effect equivalence framework to enable the deviation model to transfer across AM processes.

Tsung et al. [30] explored transfer learning to share domain knowledge and combine multiple data sources for shape deviation prediction for 3D printed parts. The domain knowledge sharing would enable statistical models built for a specific geometry or printing process to be generalized to other applications. Ferreira et al. [31] presented a methodology for building a predictive shape deviation model for different 3D geometries by implementing a class of Bayesian Neural Networks to reduce the need for user inputs and efforts. Another Bayesian methodology was employed by Tao et al. [32] to identify and model anomalies in an unorganized 3D point cloud. This method can also discriminate outliers, allowing for higher accuracy in acquiring automated measurement data. Huang [33] decomposed the input geometry into stacks of 2D primitive geometry, which simplified the printing process and enabled them to establish an impulse response formulation and modeling framework to apply control theory to AM. Khanzadeh et al. [34] proposed the use of self-organizing maps on the scan data of 3D printed parts to define the geometrical quality of a 3D printed part when a large dataset is obtained from 3D scanning. Decker et al. [35] proposed a data-driven methodology of compensating the shape of a 3D printed part to increase the geometric accuracy by training a random forest model [36], a machine learning algorithm that constructs a multitude of decision trees during training and outputs the average prediction of the individual trees, and using it to predict the systematic deviations on a known geometry. An advantage of the proposed methodology in [35] is that it is not limited to planar deviations and can control all dimensions simultaneously.

Zhu et al. [37] demonstrated that a trained Convolutional Neural Network can reliably predict layer-by-layer, in-plane, and out-of-plane geometric deviations of simple geometries. The Convolutional Neural Network presented by the authors enables a user to predict deviation distribution in three dimensions. Li et al. [38] produced a trained Conditional Adversarial Network capable of predicting out-of-plane geometric inaccuracies of a printed part with the slice of a CAD model as input. The proposed Conditional Adversarial Network allows the authors to apply compensation to the CAD model before printing. McGregor et al. [39] separated the inference of geometric accuracy of 3D printed parts into two algorithms: a feature detection algorithm to classify geometry and a shape deviation generator specialized for each type of geometry. The two-algorithm approach limits the scope of the shape deviation predictor while allowing the algorithm to be broader. Recent advances in sensor-based and computational error detection methods, as highlighted by Mehta et al. [40], emphasize the integration of vision-based and fluctuation-based sensing techniques with model-driven correction strategies to improve 3D printing accuracy. Abdolahi et al. [41] demonstrated the effectiveness of machine learning-based predictive models in assessing the capability of the process and forecasting geometric deviations in additive manufacturing, achieving an accuracy of greater than 93% in predicting the form deviation. Their findings suggest that integrating predictive analytics, such as random forest and artificial neural networks, with CNN-based compensation models could improve geometric accuracy by proactively identifying deviations before printing, reducing the dependency on sacrificial parts.

Zhao et al. introduced a neural network-based approach using point cloud data, with three set abstraction levels and three interpolation layers before a multilayer perceptron, achieving an average error of 0.037 mm on the test dataset [42].

Some researchers presented in this study have reported using Artificial Neural Networks and their derivatives to solve the complex problem of geometric accuracy improvement of 3D printed parts. Indeed, Artificial Neural Networks have proven to be

a powerful tool for such complex tasks [43]. For instance, the tasks of image classification [44] and image processing [45] present significant challenges, but Convolutional Neural Networks (CNNs) have shown great potential and are showing continuously improving results [46]. CNNs make use of spatial correlations in the input data, usually images or videos, resulting from pixel arrangement and organization [47] and can detect patterns not necessarily discernible by a human [48]. Because 3D models, surface meshes, or point clouds do not have spatial correlations in their data structure, other types of ANN have been developed to work with them. For example, PointNet can classify 3D models in real-time with a novel input format to enable itself to use the CNN architecture [49]. Many other neural networks have been developed for model segmentation [50,51], object detection [52], and 3D model reconstruction [53].

Based on a review of the current literature, while several techniques have been developed to assess and improve the geometric accuracy of 3D-printed parts by analyzing deviations and fitting models to represent systematic deviations caused by the printer, a significant gap remains unaddressed. Specifically, no methods can produce a representation of systematic deviations caused by a printer by analyzing the deviation patterns present in the raw deviation data of a 3D printed part. One proposed approach is to print five sacrificial parts of the same geometry on the same machine and average their deviation vector field [54]. The average deviation vector field can then be utilized to compensate the nominal geometry and produce a part with significantly lower geometric deviations. However, printing multiple sacrificial parts can be wasteful and expensive in terms of cost and time.

We propose a new ML compensation method to compute the systematic deviation scalar field from a single sacrificial part to address this issue. Given enough training data, the proposed method can potentially compute the systematic error of any printed part from any FFF 3D printer without requiring additional training data. It also significantly reduces the number of sacrificial parts required after training and simplifies the overall process while maintaining or improving the quality of the final part. The proposed compensation CNN also differs from currently available methodologies by eliminating the geometric specificity inherent in other compensation approaches, which often require expert knowledge to produce the deviation model for a particular geometry. In addition, by pre-training the model on a large dataset, the proposed method has the potential to further improve the geometric accuracy of 3D printed parts by ensuring that the compensation for each point is treated independently. This approach allows the algorithm to effectively handle varying patterns of deviation across different sections of a printed part, maintaining high accuracy regardless of localized differences in the deviation field.

2. Proposed Method

The present study follows previous investigations that have demonstrated that morphing the nominal geometry by the inverse of the systematic deviations measured on the sacrificial parts can improve the geometric accuracy of subsequent printed parts [54]. However, the authors note that the raw deviation scalar field obtained from measuring a single sacrificial part E_1 comprises two distinct components: systematic deviation E_S and random errors E_{R1} , as shown in Equation (1). Although compensating the nominal geometry using both components of the deviation scalar field might eliminate some of the systematic deviations, it would also contain random errors, as is shown in Equation (2). In that equation, E_C and E_{RC} are the deviation scalar field and the random errors of the compensated part. The random errors E_{R1} and E_{RC} are theoretically independent and different in any two printed parts.

To address the accumulation of random errors issue, the authors propose the use of the average deviation scalar field by printing and measuring multiple parts. Theoretically, since the systematic deviations are repeated in all parts printed in the same machine with the same printing parameters, the systematic deviations should be represented in the average deviation scalar field. In contrast, the random errors should be significantly reduced, as they are not repeated.

The average deviation E_{ave} of n parts is computed by Equation (3) with E_i , the error of scanned part i . The systematic deviation E_S average remains constant, and the averaged random deviation components E_R are added. Equation (4) shows the deviation $E_{c,ave}$ of a part compensated with the average deviation scalar field. The systematic deviations E_S are still eliminated. However, the random error component is reduced by a factor $\frac{1}{n}$, improving the geometric accuracy of the compensated part.

$$E_1 = E_S \pm E_{R1} \quad (1)$$

where:

- E_1 : Total deviation of part 1 (mm)
- E_S : Systematic deviation inherent to the printing process (mm)
- E_{R1} : Random deviation specific to part 1 (mm)

$$\begin{aligned} E_C &= E_S \pm E_{RC} - E_1 \\ E_C &= E_S \pm E_{RC} - (E_S \pm E_{R1}) \\ E_C &= \pm(E_{RC} + E_{R1}) \end{aligned} \quad (2)$$

where:

- E_C : Residual deviation after compensation (mm)
- E_S : Systematic deviation inherent to the printing process (mm)
- E_{RC} : Random deviation of the compensated part (mm)
- E_1 : Total deviation of part 1 (mm)
- E_{R1} : Random deviation specific to part 1 (mm)

$$\begin{aligned} E_{ave} &= \frac{1}{n} \sum_{i=1}^n E_i \\ E_{ave} &= \frac{n \times E_S}{n} \pm \frac{1}{n} \sum_{i=1}^n E_{Ri} \\ E_{ave} &= E_S \pm \frac{1}{n} \sum_{i=1}^n E_{Ri} \end{aligned} \quad (3)$$

where:

- E_{ave} : Average deviation of n parts (mm)
- n : Number of scanned parts (unitless)
- E_i : Total deviation of part i (mm)
- E_S : Systematic deviation inherent to the printing process (mm)
- E_{Ri} : Random deviation of part i (mm)

$$\begin{aligned} E_{c,ave} &= E_S \pm E_{RC} - E_{ave} \\ E_{c,ave} &= E_S \pm E_{RC} - E_S \pm \frac{1}{n} \sum_{i=1}^n E_{Ri} \\ E_{c,ave} &= \pm E_{RC} \pm \frac{1}{n} \sum_{i=1}^n E_{Ri} \end{aligned} \quad (4)$$

where:

- $E_{c,ave}$: Residual deviation of the compensated part using average deviation (mm)
- E_S : Systematic deviation inherent to the printing process (mm)
- E_{RC} : Random deviation of the compensated part (mm)
- E_{ave} : Average deviation of n parts (mm)
- n : Number of scanned parts (unitless)
- E_{Ri} : Random deviation of part i (mm)

However, this approach also has some limitations. For example, it may be difficult or expensive to print and measure multiple parts in some cases, especially when dealing with complex or large-scale prints. The averaging process can also introduce additional errors or inaccuracies if the scan data are not perfectly aligned with the reference geometry represented by the part's nominal CAD model.

Due to the additive nature of the 3D printing process, layers at higher Z coordinates are necessarily printed later than layers with lower Z coordinates, creating a type of repetition in time and space following the build direction that can be leveraged for analyses. Following this hypothesis, a systematic deviation related to the extruder's position would be repeated from layer to layer. It would only be detectable by comparing the deviation at one point on a layer to another point on the same layer. In contrast, random errors occur only in a single layer. They should appear as horizontal streaks on the surface of the 3D printed part and in the deviation scalar field. See Figure 1 for an example of a part with random deviations presenting themselves as horizontal lines. It would then be possible to observe a neighborhood around a point of interest in a single scan and determine the proportion of systematic deviation in the measured error. However, the complexity of the scan data, with its multidimensional nature and unorganized structure, requires extensive data manipulation and complex methods such as machine learning to accurately define the systematic deviation scalar fields.



Figure 1. Random errors on the surface of a cylindrical 3D printed part occurring on a single layer. They often present themselves as horizontal streaks or bumps on the surface of the printed part.

We propose an algorithm to accurately identify the systematic deviations from a 3D scanned model of a single 3D printed part. This algorithm uses a Convolutional Neural Network on representative arrays of the neighboring deviations for each vertex. By training the compensation CNN on a dataset of scanned 3D printed parts, we aim to learn to compute the systematic deviation scalar field for a given 3D printed part. The

deviation scalar field can then compensate the nominal 3D model to improve the accuracy of subsequent prints. An approach based on computing the systematic deviation scalar field would not be limited to a single geometry or machine but only limited by the breadth of the training dataset.

Our algorithm is divided into two modules: (1) data preprocessing and (2) a compensation Convolutional Neural Network. The data preprocessing module requires a 3D scanned model of the printed part and a surface mesh of the reference geometry. The module will produce a series of arrays that describe, in grid array form, the deviation at and around each vertex of the reference geometry. The compensation Convolutional Neural Network module consists of the compensation CNN, which takes the arrays produced by the first module as inputs and produces a single value approximating the deviation amplitude at a vertex. This amplitude is then multiplied by the vertex's normal vector to create the compensation vector field.

Each algorithm module is discussed further in Sections 2.1 and 2.2, respectively.

2.1. Data Preprocessing

As discussed above, the unstructured and multidimensional data obtained by 3D scanning a 3D printed part are not conducive to analyses requiring positional information. Preprocessing the scan data in a generalized manner is necessary to assemble the deviations of a vertex neighborhood in an accessible configuration.

For each vertex v_i on the reference geometry's surface mesh, a planar grid tangent to that vertex is created. The orientation vector V_{ori} is initially aligned with the z axis of the reference geometry. If the angle between the normal vector n_{vi} and the $X - Y$ plane exceeds 45° , V_{ori} is reoriented towards the centroid of the reference geometry from the vertex's position. The perpendicular vector V_{per} is computed by taking the cross product of V_{ori} and n_{vi} and is used to align the rows of the grid. Subsequently, the orientation vector V_{ori} is recomputed by taking the cross product of V_{per} and n_{vi} , aligning the columns of the grid. Figure 2 illustrates three example situations and demonstrates the orientation of the orientation vector and perpendicular vectors.

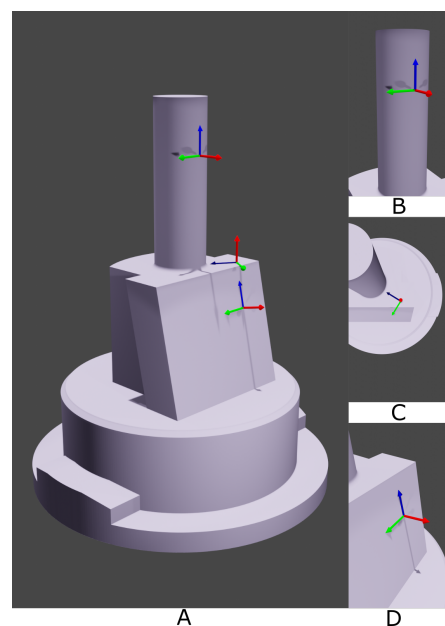


Figure 2. (A) Examples of vectors on the reference geometry's surface mesh. Red arrows are the normal vector (n_{vi}), green arrows are the perpendicular vectors (v_{per}), and blue arrows are the orientation vectors (v_{ori}). (B) Example of vectors on cylinder. (C) Example on a plane perpendicular to the Z-axis. (D) Example on a sloped surface.

The position of the center of each grid cell p_{uv} is defined by Equation (5), where u and v are the cell indices in the grid array, $shape$ is the number of cells per row and column, and D is the size of each grid cell. Parameter D is driven by the point density of the surface mesh, as it determines the grid's ability to capture the necessary detail of the surface. Each cell must contain at least one vertex from the mesh to ensure proper representation. A size D that is too small would result in insufficient point inclusion per cell, compromising the grid's representativeness of the surface. Conversely, a size D that is too large would include too many points within each cell, reducing the cells' independence and the grid's ability to accurately reflect local variations in the surface mesh. In contrast, the parameter $shape$ is driven by the size of the typical part that will be analyzed using the grid arrays. We aim for the largest possible size to ensure that the CNN has the best receptive field and contextual information. However, if $shape$ is too large, the grid arrays will contain many empty cells when analyzing the edges of the surface mesh. Therefore, $shape$ must be carefully chosen to balance the receptive field and contextual information provided to the CNN with the need to minimize empty values in the grid arrays. This study defines the parameters $shape$ as 20 cells per row and column and D as 0.2 mm.

$$p_{uv} = v_i - V_{per}D\left((u-1) - \frac{shape-1}{2}\right) - V_{ori}D\left((v-1) - \frac{shape-1}{2}\right) \quad (5)$$

where:

- p_{uv} : Position vector of the center of cell (u, v) (mm)
- v_i : Reference point (typically the center of the surface patch) (mm)
- V_{per} : Unit vector perpendicular to the surface patch (unitless)
- V_{ori} : Unit vector in the orientation direction of the patch (unitless)
- D : Grid cell size (mm)
- $shape$: Number of cells per row and column in the grid (unitless)
- u, v : Indices of the cell in the grid array (unitless)

Once the grid has been constructed, a ball query is applied to find all vertices of the reference mesh within a certain radius r_{ball} of each grid cell center p_{uv} using a K-d tree [55], a data structure used for organizing multidimensional data, facilitating efficient search operations by partitioning the space into smaller regions. This radius is related to the size D by this equation: $r_{ball} = \sqrt{2 * D^2}$. The deviation of each selected vertex is then averaged and stored in an accompanying array A_i . Vertices can be excluded from this average if the angle between the normal vector n_{vi} and the selected vertex's normal vector exceeds a threshold value. This study sets this threshold at 45° as a convenient angle threshold for simplifying computations while being an effective filter.

This process allows the creation of a set of grid arrays representing the neighborhood deviations of the printed part at each vertex of the reference geometry. The grid arrays A_i and the deviation at vertex v_i are then fed to the compensation CNN to determine the systematic deviation for vertex v_i . Figure 3 shows examples of this grid applied to different vertices of a mesh with various geometric features, with the deviation scalar field color-coded to show the deviation of the printed part from the reference geometry.

2.2. Compensation Convolutional Neural Network

In this section, we present the development of a compensation CNN designed to interpret the deviation scalar represented in the 2D grid arrays prepared in the previous section. These grid arrays serve as input data, akin to grayscale images with dimensions of 20×20 pixels. Given the small size of the inputs, we adopted an architecture inspired by the Visual Geometry Group (VGGNet) [56], renowned for its use of small convolution filters (3×3) but with significantly increased depth. During our implementation, we found

that a combination of 5×5 and 3×3 convolution filters yielded better performance after hyperparameter tuning. The final architecture of the compensation CNN, illustrated in Figure 4, incorporates Rectifier Linear Units (ReLUs) alongside each convolution filter to introduce non-linearity. Additionally, max-pooling layers were integrated to capture the most salient features while reducing computational complexity in subsequent layers. The number of channels C_1 , C_2 , C_3 , and C_4 were also optimized, and the best-performing numbers of channels were 9, 17, 27, and 64, respectively.

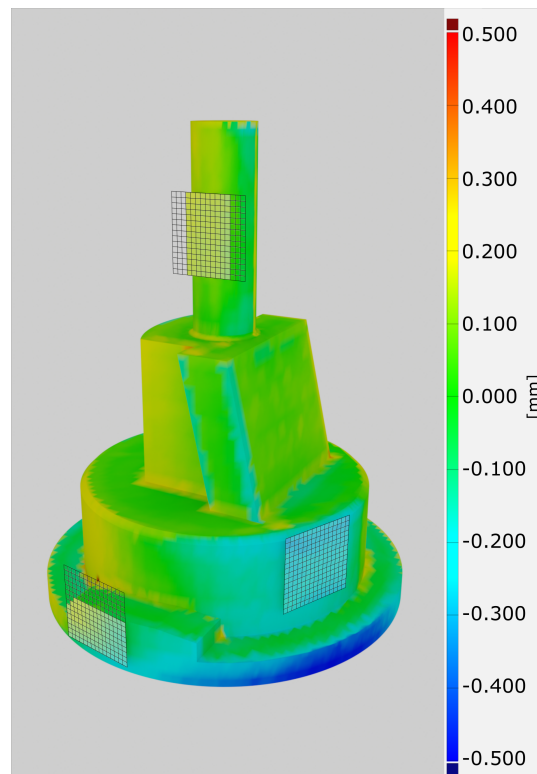


Figure 3. Illustration of grid arrays presented on a geometry with various geometric features.

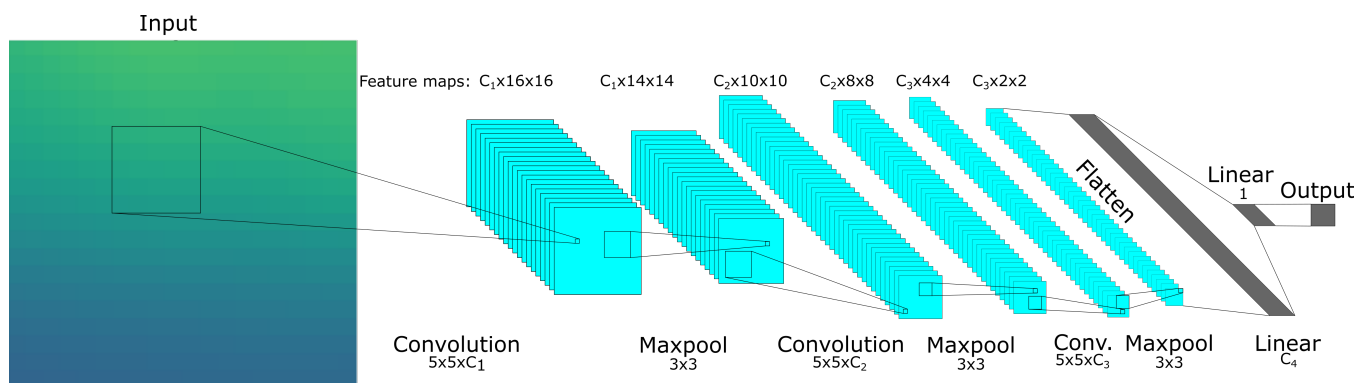


Figure 4. Architecture of the compensation Convolutional Neural Network used to determine the systematic deviations present in the input image. Each intermediary level is a feature map, where the result of the previous layer is represented.

Conventional CNN architectures such as VGGNet or the popular ResNet [57] are typically designed for classification tasks, wherein the network's objective is to categorize input data into predefined classes. To facilitate this task, these networks commonly employ a final fully connected layer with a size equal to the number of categories in the classification scheme. This layer consolidates the features extracted by the preceding convolutional layers and maps them to the respective class labels, enabling the network to make accurate

predictions. In our study, the objective of the compensation CNN differs from typical classification tasks; instead of categorizing input data into predefined classes, our CNN performs regression, where the goal is to predict a continuous value. We have designed our CNN with two final fully connected layers to accommodate this task. The first layer performs the necessary non-linear computations on the extracted features. In contrast, the second layer consolidates these computed features into a single scalar value, serving as the output of the regression task. This output is then applied to the nominal geometry following Equation (6):

$$v_{i,c} = v_i - R_i \times n_{vi} \quad (6)$$

where:

- v_i : Coordinate vector of the original vertex i on the reference geometry (mm)
- R_i : Output of the compensation CNN for vertex i , representing the systematic deviation (mm)
- n_{vi} : Normal vector at vertex i on the reference geometry (unitless)
- $v_{i,c}$: Coordinate vector of the compensated vertex i (mm)

This process effectively adjusts the 3D model to account for the systematic deviation of the printed part from the reference geometry, resulting in a compensated 3D model that can be printed with improved geometric accuracy and no systematic deviations.

3. Experimental Methodology

We now present the methodology employed to prepare the dataset used to train the compensation CNN, the training process itself, and the evaluation of the compensation CNN.

3.1. Reference Geometry

The geometry selected for training the compensation CNN had to meet specific requirements: it must be quick to print, easy to analyze mathematically, and suitable for comparing the proposed methodology to other 3D printing error compensation methods. Thus, a primary cylinder, a base, and a clocking feature were chosen. This configuration takes less than one hour to print, and the cylindrical portion of the point cloud can be analyzed mathematically to determine global deviation metrics such as RMS, mean, and standard deviation. Cylinders are also commonly used in other 3D printing error compensation methodologies, helping to reduce distortions induced by the heated build plate and minimizing misalignment errors. Figure 5 shows a rendered version of the chosen geometry.

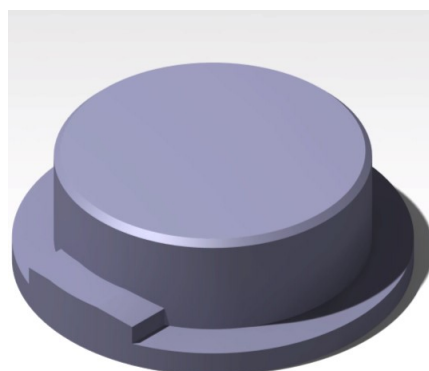


Figure 5. Render of the reference geometry used for training and evaluating the compensation CNN. The part measures 200 mm in diameter and 100 mm in height.

The nominal CAD model and the files used for the printing and the inspection were prepared using the CATIA V5-6R2018 (Dassault Systèmes, Vélizy-Villacoublay, France) CAD software. The tessellated file used to prepare the printing process was created with a maximum permissible chord error of 0.001 mm between the nominal geometry and the lines of any triangle created by the tessellation process and a maximum edge length of 0.1 mm to obtain a regularly meshed tessellated file. Consequently, the file contained 340,925 points and 681,846 triangles and should be a negligible source of geometric deviations for the experiments. We focused exclusively on non-horizontal surfaces for training and evaluating the compensation CNN. This decision is driven by 3D printing's layer-wise nature, where even minor compensations on horizontal surfaces will necessarily cause a discrete layer change.

3.2. Dataset Collection

To assemble the dataset, a combination of four Ultimaker 3 and four Ultimaker S5 (Ultimaker, Geldermalsen, The Netherlands) 3D printers, shown in Figure 6, were employed using 2.85 mm polylactic acid (PLA) filament from Polymaker (Polymaker, Shanghai, China). The training dataset comprised 12 sets of five 3D printed parts of the chosen reference geometry, totaling 60 3D printed parts. It is important to note that each Ultimaker 3 produced two sets of parts. However, the first sets were printed much earlier than the second sets and should be considered independent due to potential differences in printer calibration or environmental conditions. Each part was printed using the default “fast” printing parameters from the Cura 4.13.0 (Ultimaker, Geldermalsen, The Netherlands) slicing software. The key printing parameters are presented in Table 1 and are the same for both 3D printer models.

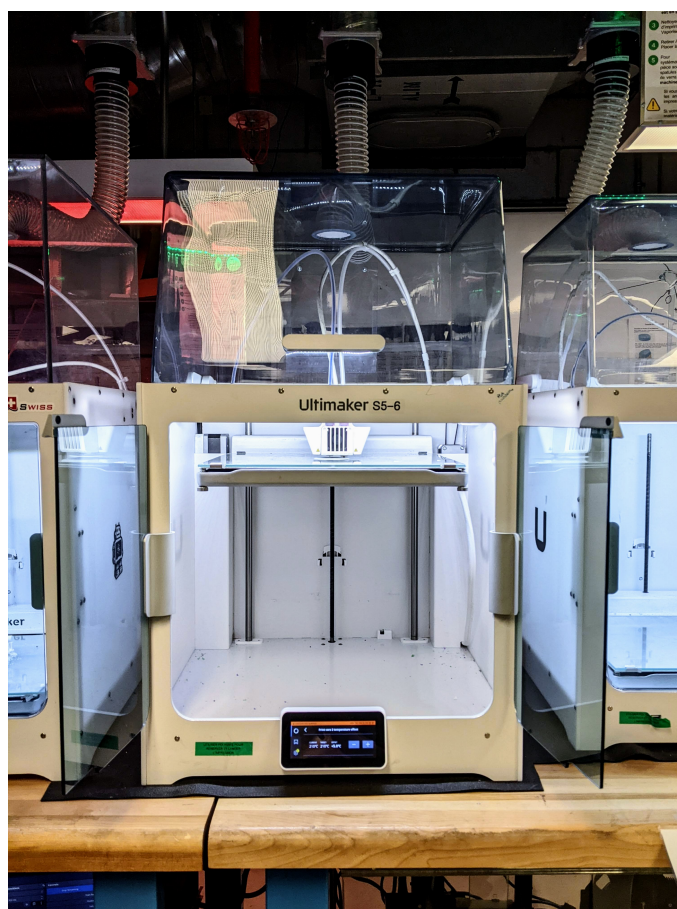
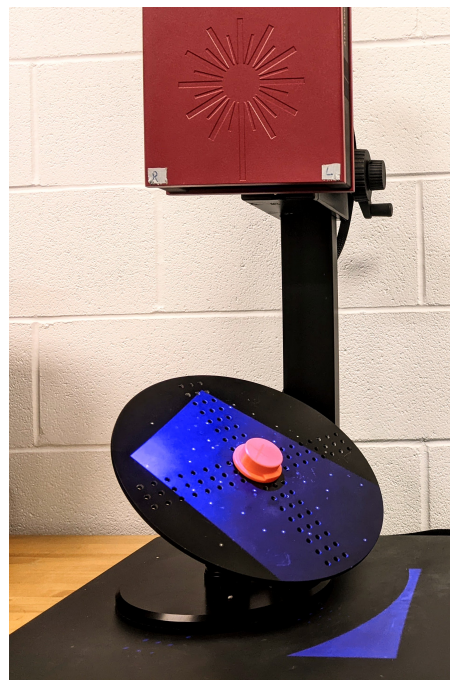


Figure 6. An Ultimaker S5 printing the chosen reference geometry.

Table 1. Printing parameters for the original and compensated parts on the Ultimaker 3 and Ultimaker S5.

Printing temperature	215 °C
Printing speed	70 mm/s
Cooling	100%
Support	None
Infill type	Triangles
Infill density	10%
Plate adhesion	None
Layer height	0.2 mm
Nozzle diameter	0.4 mm

The 3D scanner used to obtain the digitized version of the 3D printed part was an Atos Core 200 (GOM, Braunschweig, Germany) shown in Figure 7. This scanner was shown to attain a probe error in measuring form of 0.002 mm and a probe error in measuring size of 0.009 mm [58]. The GOM Scan 2018 software interpreted the images from the scanner to produce a tessellated 3D model representing the 3D printed parts using the high-resolution setting, resulting in a mesh of approximately 550,000 points at a density of roughly 30 $\frac{\text{points}}{\text{mm}^2}$. The alignment of the 3D scanned models to the nominal geometry was conducted using the GOM Inspect 2018 software.

**Figure 7.** Atos Core 200, 3D scanning the part.

3.3. Data Preprocessing and Compensation CNN Training

The preprocessing of the training dataset has been implemented according to the methodology presented in Section 2.1. This implementation was performed using the Python 3.9 programming language using external libraries, namely Trimesh 3.10.5 [59] for mesh processing and Libigl Python bindings 2.2.1 [60] for the computation of the deviation scalar field. To enhance this study's stability and repeatability, we opted to utilize the vertices of the CAD tessellated geometry as measurement points and the 3D scan data as the reference for the computation of the deviation scalar field. Consequently, this approach ensured that a corresponding deviation value can be identified for each coordinate within every deviation scalar field. The grid arrays were then prepared using the previously

computed deviation scalar field. The ground truths were also determined using the average deviation scalar field of each set of five printed parts. The output of the data preprocessing for the training dataset was 3,439,800 groups of input grid arrays, deviation values, and ground truths.

Following the preparation of the training dataset, we implemented the compensation CNN using the PyTorch 1.12.1 Machine Learning Framework [61] following the methodology presented in Section 2.2. A portion of the training dataset was set aside to create the validation dataset to validate the training process. The training dataset underwent backpropagation [62] to minimize the Mean Squared Error (MSE) [63] loss of the inference to the ground truths, while the validation dataset did not. By segregating these datasets, we mitigated the risk of overfitting [64], where the model would memorize training data patterns rather than learning meaningful relationships. Maintaining independence between the validation and training datasets ensured our CNN's robustness and generalization capability. An independent validation set is also an unbiased evaluator, providing insights into a model's performance on unseen data. We segregated whole sets of printed parts from unique printers to ensure the validation dataset's independence. This resulted in a split training and validation dataset of 2,579,850 and 859,950 inputs, respectively.

The training process was completed using an NVIDIA TITAN Xp (NVIDIA, Santa Clara, CA, USA), a Cuda enabled Graphics Processing Unit (GPU). Using the Weights & Biases 0.13.4 software [65], we tracked various training and validation parameters to determine when the compensation CNN performed optimally on data it had not been trained on. We also used the hyper-parameter tuning tools of the Weights & Biases software to enable the CNN to better generalize on unseen data. Also, the ADAM optimizer [66] was used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and an initial learning rate lr_0 of 1×10^{-3} with an exponential learning rate schedule [67] following Equation (7), where n_{epoch} is the current epoch and $lr_{n_{epoch}}$ is the value of the learning rate at that epoch.

$$lr_{n_{epoch}} = lr_0 \times 0.995^{n_{epoch}} \quad (7)$$

where:

- lr_0 : Initial learning rate (unitless)
- n_{epoch} : Current epoch number (unitless)
- $lr_{n_{epoch}}$: Learning rate at the current epoch (unitless)

3.4. Compensation CNN Evaluation

After successfully training the compensation CNN, it was necessary to evaluate its efficacy with brand-new data. In this study, we evaluated the CNN by using a compensation process with a newly printed part. The same reference geometry was chosen for this evaluation to better showcase the compensation CNN's efficacy. We used an Ultimaker 3 3D printer with a significant time difference between the data collection and this evaluation, making the 3D printed part independent from the training dataset. The 3D scanner was the Atos Core 200 presented earlier with its accompanying software: GOM Scan for scan interpretation and mesh creation, and GOM Inspect for mesh alignment and metrological analysis.

4. Results

4.1. Compensation CNN Training

We now present the results of the training process for the compensation CNN. This training process spanned 500 epochs and lasted 3 h. Figure 8 illustrates the progression of these losses throughout the training period. The presented loss metrics provided insights

into the performance of the CNN, indicating its ability to approximate the average deviation scalar field. Optimal performance was observed when the validation loss reached its minimum, highlighting the model's effectiveness. Notably, at epoch 306, the CNN achieved its lowest validation loss, suggesting more robust generalization capabilities. The MSE validation loss was $-5.24 \times 10^{-6} \text{ mm}^2$ and the MSE training loss was $-6.07 \times 10^{-6} \text{ mm}^2$. The MSE loss presented here is the difference between the MSE loss of deviation at the current vertex and the MSE loss of the compensation CNN's output. It is negative because the CNN output is closer to the ground truth than the input deviation. However, after this epoch, an increase in the validation loss was noted, indicative of potential overfitting.

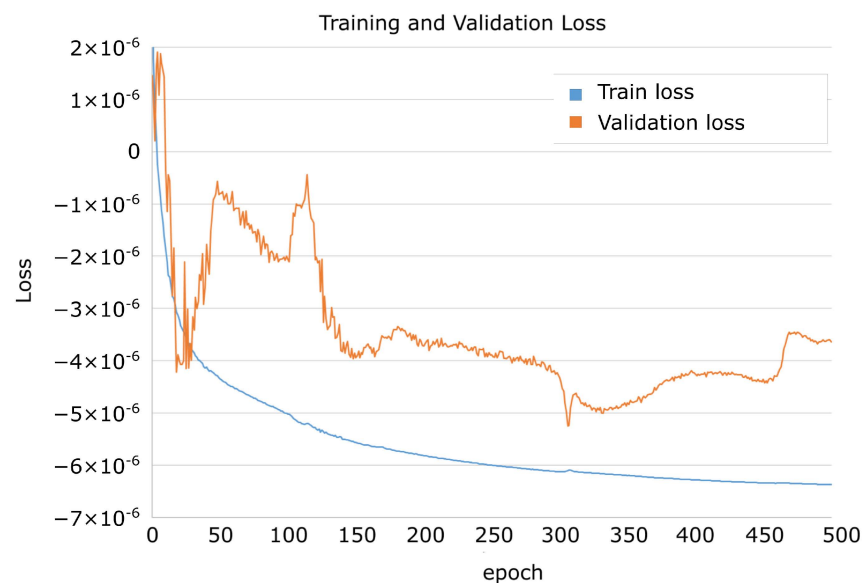


Figure 8. Training and validation loss of the Convolutional Neural Network.

To test the efficacy of the compensation CNN in isolating systematic deviations within a deviation scalar field, we can visually compare the input deviation scalar field with the inferred output of the compensation CNN against the ground truth. From Figure 9, we can observe that the inferred deviation scalar field is representative of the systematic deviations presented in the ground truth. It also eliminates other random deviations passed through the average deviation scalar field. In the subsequent section, we delve deeper into an analysis of a compensated 3D printed part utilizing the compensation CNN, aiming to evaluate the geometric improvements facilitated by this trained model.

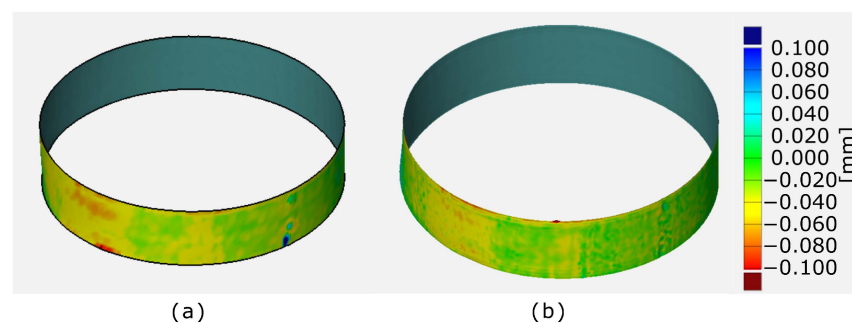


Figure 9. (a) Colormap of the difference between the 3D scan of the cylindrical portion of a 3D printed part. (b) Colormap of the difference between the CNN output using that 3D scan and the corresponding ground truth.

4.2. Result of Compensation Using the Convolutional Neural Network

To assess the effectiveness of the compensation CNN in identifying systematic deviations from a deviation scalar field, we compare the geometric accuracy of a compensated 3D printed part with the deviation scalar field of the sacrificial 3D printed part.

For this evaluation and to showcase the capability of the compensation CNN, we utilized the same geometry as previously presented on an Ultimaker 3, and ensured that sufficient time had elapsed between the collection of the training and validation datasets and this evaluation, resulting in significantly different deviation patterns due to the continued usage and maintenance of the 3D printer. Subsequently, this part was scanned using the Atos Core 200 3D scanner, following the same procedure as previously presented. The acquired 3D scan data was then aligned with the nominal geometry to extract the deviation scalar field and processed using the preprocessing methodology outlined in Section 2.1. The output of the compensation CNN was used to compensate the nominal geometry, which was subsequently printed using the same machine and printing parameters as the sacrificial part.

The sacrificial and compensated parts used for the evaluation are depicted in Figure 10, with their deviation scalar fields represented as colormaps. A visual inspection reveals that most deviations have been eliminated.

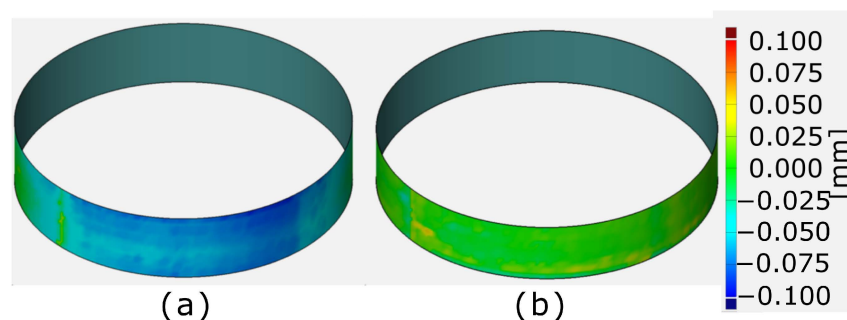


Figure 10. Deviation colormap of the cylindrical feature of the (a) sacrificial and (b) CNN compensated 3D printed part.

A more meticulous observation of the compensated deviation scalar field of the printed part reinforces our initial hypothesis. Assuming systematic deviations have been successfully eliminated, the remaining errors should stem from random sources. These remaining random errors are observable on the surface of the compensated 3D printed part as horizontal patterns of deviation. Hence, horizontal deviations could serve as indicators for identifying random errors.

To further study the efficacy of the compensation CNN to reduce the geometric deviation of the compensated 3D printed part, a numerical analysis is presented in Figure 11. It demonstrates that the CNN compensation method leads to substantial improvements in geometric accuracy, as evidenced by reductions in mean absolute error, standard deviation, and improvements in cylindricity. The reduction in the standard deviation of the deviation scalar field holds particular significance in this evaluation, as it provides insights into the distribution of the errors in the compensated part. In that sense, it serves as a proxy to evaluate the filtration of random errors from the sacrificial part of the compensation process. Reinforcing this, the standard deviation of the CNN-compensated 3D printed part is 0.0173 mm, which closely aligns with the standard deviation of 0.020 mm obtained by Jadayel and Khameneifar [54] for the compensated part using the average deviation scalar field of five parts. Considering that the 3D printer and the 3D scanner inherently introduce random errors into the process, it is essential to acknowledge that some random errors

cannot be eliminated. This similar result between the two experiments may suggest that we are approaching such a limit.

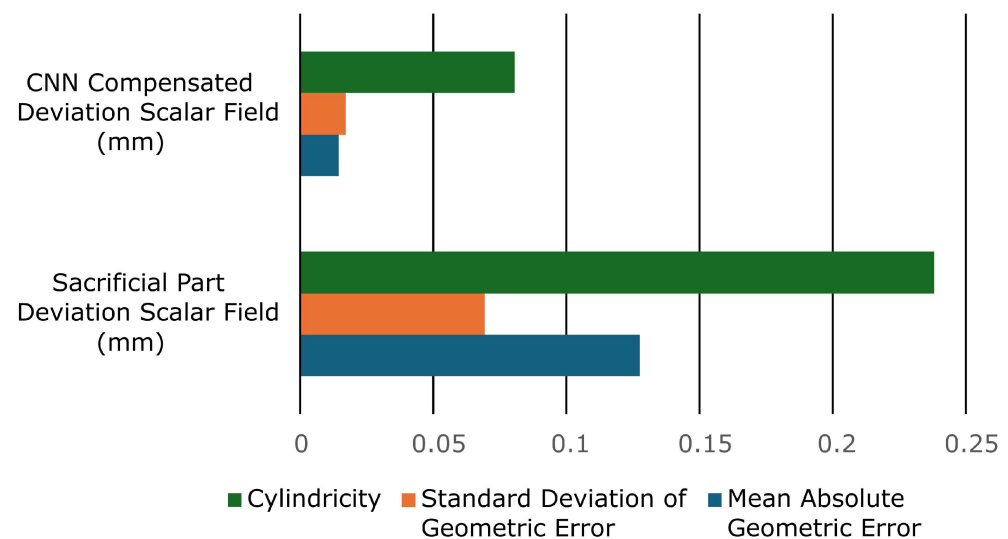


Figure 11. CNN compensation evaluation result.

5. Discussion

This study proposes a compensation methodology to reduce geometric deviations in 3D printed parts. This methodology consists of the preprocessing algorithm and the Convolutional Neural Network. The first introduces an innovative approach to interpreting the deviation scalar field, or color map, by encoding not only the scalar values but also capturing the patterns and spatial correlations of the deviations across the surface of the geometry in a computer-readable format, allowing the following algorithms access to insights previously inaccessible. In essence, it transforms disorganized information into structured data. While alternative methods for organizing information exist, such as generating a parametric function to depict surface deviations on geometry, they fail to account for the original dataset's dimensionality and discrete nature.

This study's proposed preprocessing method creates 2D grid arrays of the deviation neighborhood, representing the deviation values associated with the vertices neighboring each vertex of the nominal geometry within a certain distance. It organizes the deviations measured on the surface of the 3D printed part as a 2D projection organized in a grid array representing the measured data in 3D while being entirely interpretable to a machine learning algorithm. It is also more accurate than other methods of organizing data because there is no interpolation or extrapolation in the output; only measured data is presented. This ensures that the information provided is precise and reliable without introducing inaccuracies through additional computations. These 2D grid arrays also encapsulate the local deviations and the spatial hierarchical structure of the deviation scalar field, allowing for the detailed analysis of deviation patterns around a vertex.

The second algorithm presented in this work is specialized to detect patterns in an image or image-like data, in our case, a grid of values. After being trained on a large dataset, the compensation CNN is specially optimized to detect patterns of systematic deviations and approximate the proportion of the deviation that is random vs systematic. Thanks to the preprocessing step, the compensation CNN can leverage the spatial hierarchical structure of the deviations in the grid arrays without requiring knowledge of the underlying geometry. The proposed methodology presents distinct advantages, notably its generalized and machine-agnostic approach. Although the generalization is not thoroughly demonstrated in this work, NNs and CNNs have consistently shown a remarkable capacity

for generalization when trained on extensive datasets [68]. In addition, the success of the compensation CNN on a limited dataset for training and validation demonstrates the efficacy of the compensation methodology. While the proposed CNN-based compensation method demonstrates significant improvements, it is important to note that the current model has been trained and validated using a single reference geometry. Although the architecture and pre-processing methodology are designed to be geometry-agnostic, full validation across diverse part geometries and printer types is still ongoing. Further experimental work is needed to ensure robust generalization of the CNN across a wider range of real-world applications. Incorporating additional training data from multiple shapes and machines is a key objective of our future work to develop a fully generalized compensation tool.

To obtain a generalized compensation CNN, popular image classification CNNs like AlexNet and VGGNet are pre-trained using datasets like ImageNet, holding more than 1.2 million images [44], and trained on specific datasets containing millions of images [56]. For our implementation, the precise number of independent data points required remains uncertain and would likely depend on factors such as the complexity of the CNN model and the desired level of performance. Given the resource limitations of this study, a thorough investigation into the optimal dataset size is not considered. However, this remains an important avenue for future research. These data points would consist of sets of 3D scanned 3D printed parts of different geometry produced from different machines using various sets of printing parameters. On the other hand, a production team that is required to scan all produced 3D printed parts for quality assurance purposes could already have all the necessary data to train the compensation CNN.

Assuming successful training leads to a generalized model, no additional training, work, or adjustment would be needed to obtain a compensated nominal geometry. Only a single sacrificial part per machine and printing parameter would be necessary to compensate the nominal model for a production run. In contrast, previous methodologies necessitated a more labor-intensive approach, especially with increasing geometric complexity. For instance, the parametric function required to model the deviations on a 3D free-form shape is more complex than the parametric function necessary to model the deviations on a primitive geometry. Another method of compensating a nominal geometry to reduce systematic deviations required five sacrificial parts for each machine and sets of printing parameters [54]. The method proposed in this paper reduces wasted time, material, and cost by 80%, requiring only a single sacrificial part per machine and printing parameters.

6. Conclusions

The proposed method involves training a compensation CNN to predict a deviation scalar field representing systematic errors in 3D printed parts. This is performed by analyzing grid arrays derived from the deviation field of a sacrificial print scanned with a structured light 3D scanner. The approach was successfully demonstrated and shown to enhance the geometric accuracy of subsequent prints by compensating the nominal geometry using the inverse of the predicted systematic deviations.

The compensated part used for validation showed an average geometric error reduction of 88.5% and a 75% reduction in standard deviation compared to the non-compensated part. By training the CNN to detect and quantify systematic patterns within the deviation field, we minimized the influence of random deviations present in the sacrificial print. This led to a 66% improvement in cylindricity of the compensated part.

These improvements suggest that lower-cost 3D printers can achieve results comparable to high-end machines, or that tighter tolerances can be met using the proposed

compensation method. While this study demonstrates the method's potential, future work should expand the dataset to include a wider range of geometries and printing platforms.

Future work would be required in the preprocessing algorithm to resolve the missing data issue when no valid vertex of the nominal geometry is present to occupy a cell in the grid array. As the CNN is a purely mathematical tool, a placeholder value has to be inserted in the cell, as it is impossible to input an empty cell in a grid array. This placeholder can still affect the perception of the CNN, even if pooling layers and masks were implemented.

Author Contributions: Conceptualization, M.J. and F.K.; methodology, M.J. and F.K.; software, M.J.; validation, M.J.; formal analysis, M.J. and F.K.; investigation, M.J.; resources, M.J. and F.K.; data curation, M.J.; writing—original draft preparation, M.J.; writing—review and editing, M.J. and F.K.; visualization, M.J.; supervision, F.K.; project administration, F.K.; funding acquisition, F.K. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant RGPIN-2017-06922.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

Abbreviations

The following abbreviations are used in this manuscript:

AM	Additive Manufacturing
PPE	Personal Protective Equipment
FDM	Fused Deposition Modelling
FFF	Fused Filament Fabrication
SLA	Stereolithography Apparatus
CAD	Computer Aided Design
FEA	Finite Element Analysis
ICP	Iterative Closest Point
STL	Standard Tessellation Language
GD&T	Geometric Dimensioning and Tolerancing
CMM	Coordinate Measuring Machine

References

1. Sachs, E.; Cima, M.; Williams, P.; Brancazio, D.; Cornie, J. Three dimensional printing: Rapid tooling and prototypes directly from a CAD model. *J. Eng. Ind.* **1992**, *114*, 481–488. [\[CrossRef\]](#)
2. Campbell, I.; Bourell, D.; Gibson, I. Additive manufacturing: Rapid prototyping comes of age. *Rapid Prototyp. J.* **2012**, *18*, 255–258. [\[CrossRef\]](#)
3. Colosimo, B.M.; Huang, Q.; Dasgupta, T.; Tsung, F. Opportunities and challenges of quality engineering for additive manufacturing. *J. Qual. Technol.* **2018**, *50*, 233–252. [\[CrossRef\]](#)
4. Mehrpouya, M.; Dehghanghadikolaei, A.; Fotovvati, B.; Vosooghnia, A.; Emamian, S.S.; Gisario, A. The potential of additive manufacturing in the smart factory industrial 4.0: A review. *Appl. Sci.* **2019**, *9*, 3865. [\[CrossRef\]](#)
5. Ye, Z.; Liu, C.; Tian, W.; Kan, C. A deep learning approach for the identification of small process shifts in additive manufacturing using 3D point clouds. *Procedia Manuf.* **2020**, *48*, 770–775. [\[CrossRef\]](#)
6. Tamir, T.S.; Xiong, G.; Fang, Q.; Yang, Y.; Shen, Z.; Zhou, M.; Jiang, J. Machine-learning-based monitoring and optimization of processing parameters in 3D printing. *Int. J. Comput. Integr. Manuf.* **2022**, *36*, 1362–1378. [\[CrossRef\]](#)
7. Brion, D.A.; Pattinson, S.W. Generalisable 3D printing error detection and correction via multi-head neural networks. *Nat. Commun.* **2022**, *13*, 4654. [\[CrossRef\]](#)
8. Chacón, J.; Caminero, M.A.; García-Plaza, E.; Núñez, P.J. Additive manufacturing of PLA structures using fused deposition modelling: Effect of process parameters on mechanical properties and their optimal selection. *Mater. Des.* **2017**, *124*, 143–157. [\[CrossRef\]](#)

9. Liu, S.; Zhao, P.; Wu, S.; Zhang, C.; Fu, J.; Chen, Z. A pellet 3D printer: Device design and process parameters optimization. *Adv. Polym. Technol.* **2019**, *2019*, 1–8. [\[CrossRef\]](#)
10. Elkaseer, A.; Schneider, S.; Scholz, S.G. Experiment-based process modeling and optimization for high-quality and resource-efficient FFF 3D printing. *Appl. Sci.* **2020**, *10*, 2899. [\[CrossRef\]](#)
11. Deswal, S.; Narang, R.; Chhabra, D. Modeling and parametric optimization of FDM 3D printing process using hybrid techniques for enhancing dimensional preciseness. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2019**, *13*, 1197–1214. [\[CrossRef\]](#)
12. Nath, P.; Olson, J.D.; Mahadevan, S.; Lee, Y.T.T. Optimization of fused filament fabrication process parameters under uncertainty to maximize part geometry accuracy. *Addit. Manuf.* **2020**, *35*, 101331. [\[CrossRef\]](#)
13. Sarraga, R.F. Modifying CAD/CAM surfaces according to displacements prescribed at a finite set of points. *Comput.-Aided Des.* **2004**, *36*, 343–349. [\[CrossRef\]](#)
14. Chowdhury, S.; Mhapsekar, K.; Anand, S. Part Build Orientation Optimization and Neural Network-Based Geometry Compensation for Additive Manufacturing Process. *J. Manuf. Sci. Eng.* **2018**, *140*, 31009. [\[CrossRef\]](#)
15. Chowdhury, S.; Anand, S. Artificial neural network based geometric compensation for thermal deformation in additive manufacturing processes. In Proceedings of the International Manufacturing Science and Engineering Conference, Blacksburg, VA, USA, 27 June–1 July 2016; Volume 49910, p. V003T08A006.
16. Francis, J.; Bian, L. Deep Learning for Distortion Prediction in Laser-Based Additive Manufacturing using Big Data. *Manuf. Lett.* **2019**, *20*, 10–14. [\[CrossRef\]](#)
17. Huang, Y.; Wang, T.; Li, W.; Xu, Y.; Ji, H.; Zhang, H.; Liu, H.; Yang, Y. Geometric compensation of sintering deformation in binder jetting additive manufacturing based on artificial neural network. *Prog. Addit. Manuf.* **2024**, 1–17. [\[CrossRef\]](#)
18. Tong, K.; Amine Lehtihet, E.; Joshi, S. Parametric error modeling and software error compensation for rapid prototyping. *Rapid Prototyp. J.* **2003**, *9*, 301–313. [\[CrossRef\]](#)
19. Huang, Q.; Nouri, H.; Xu, K.; Chen, Y.; Sosina, S.; Dasgupta, T. Statistical Predictive Modeling and Compensation of Geometric Deviations of Three-Dimensional Printed Products. *J. Manuf. Sci. Eng.* **2014**, *136*, 61008. [\[CrossRef\]](#)
20. Shah, P.; Racasan, R.; Bills, P. Comparison of different additive manufacturing methods using computed tomography. *Case Stud. Nondestruct. Test. Eval.* **2016**, *6*, 69–78. [\[CrossRef\]](#)
21. Wang, A.; Song, S.; Huang, Q.; Tsung, F. In-Plane Shape-Deviation Modeling and Compensation for Fused Deposition Modeling Processes. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 968–976. [\[CrossRef\]](#)
22. Cressie, N. Spatial prediction and ordinary kriging. *Math. Geol.* **1988**, *20*, 405–421. [\[CrossRef\]](#)
23. Cheng, L.; Wang, A.; Tsung, F. A prediction and compensation scheme for in-plane shape deviation of additive manufacturing with information on process parameters. *IIE Trans.* **2018**, *50*, 394–406. [\[CrossRef\]](#)
24. Huang, Q. An Analytical Foundation for Optimal Compensation of Three-Dimensional Shape Deformation in Additive Manufacturing. *J. Manuf. Sci. Eng.* **2016**, *138*, 061010. [\[CrossRef\]](#)
25. Luan, H.; Huang, Q. Prescriptive Modeling and Compensation of In-Plane Shape Deformation for 3D Printed Freeform Products. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 73–82. [\[CrossRef\]](#)
26. Huang, Q.; Wang, Y.; Lyu, M.; Lin, W. Shape Deviation Generator—A Convolution Framework for Learning and Predicting 3D Printing Shape Accuracy. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1486–1500. [\[CrossRef\]](#)
27. Wang, Y.; Ruiz, C.; Huang, Q. Learning and Predicting Shape Deviations of Smooth and Non-Smooth 3D Geometries Through Mathematical Decomposition of Additive Manufacturing. *IEEE Trans. Autom. Sci. Eng.* **2023**, *20*, 1527–1538. [\[CrossRef\]](#)
28. Sabbaghi, A.; Huang, Q.; Dasgupta, T. Bayesian model building from small samples of disparate data for capturing in-plane deviation in additive manufacturing. *Technometrics* **2018**, *60*, 532–544. [\[CrossRef\]](#)
29. Sabbaghi, A.; Huang, Q. Model transfer across additive manufacturing processes via mean effect equivalence of lurking variables. *Ann. Appl. Stat.* **2018**, *12*, 2409–2429. [\[CrossRef\]](#)
30. Tsung, F.; Zhang, K.; Cheng, L.; Song, Z. Statistical transfer learning: A review and some extensions to statistical process control. *Qual. Eng.* **2018**, *30*, 115–128. [\[CrossRef\]](#)
31. Ferreira, R.d.S.B.; Sabbaghi, A.; Huang, Q. Automated geometric shape deviation modeling for additive manufacturing systems via Bayesian neural networks. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 584–598. [\[CrossRef\]](#)
32. Tao, C.; Du, J.; Chang, T.S. Anomaly detection for fabricated artifact by using unstructured 3D point cloud data. *IIE Trans.* **2023**, *55*, 1174–1186. [\[CrossRef\]](#)
33. Huang, Q. An impulse response formulation for small-sample learning and control of additive manufacturing quality. *IIE Trans.* **2022**, *55*, 926–939. [\[CrossRef\]](#)
34. Khanzadeh, M.; Rao, P.; Jafari-Marandi, R.; Smith, B.K.; Tschopp, M.A.; Bian, L. Quantifying geometric accuracy with unsupervised machine learning: Using self-organizing map on fused filament fabrication additive manufacturing parts. *J. Manuf. Sci. Eng.* **2018**, *140*, 31011. [\[CrossRef\]](#)
35. Decker, N.; Lyu, M.; Wang, Y.; Huang, Q. Geometric Accuracy Prediction and Improvement for Additive Manufacturing Using Triangular Mesh Shape Data. *J. Manuf. Sci. Eng.* **2021**, *143*, 61006. [\[CrossRef\]](#)

36. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
37. Zhu, Z.; Ferreira, K.; Anwer, N.; Mathieu, L.; Guo, K.; Qiao, L. Convolutional Neural Network for geometric deviation prediction in Additive Manufacturing. *Procedia CIRP* **2020**, *91*, 534–539. [\[CrossRef\]](#)
38. Li, L.; McGuan, R.; Isaac, R.; Kavehpour, P.; Candler, R. Improving precision of material extrusion 3D printing by in-situ monitoring & predicting 3D geometric deviation using conditional adversarial networks. *Addit. Manuf.* **2021**, *38*, 101695. [\[CrossRef\]](#)
39. McGregor, D.J.; Bimrose, M.V.; Shao, C.; Tawfick, S.; King, W.P. Using machine learning to predict dimensions and qualify diverse part designs across multiple additive machines and materials. *Addit. Manuf.* **2022**, *55*, 102848. [\[CrossRef\]](#)
40. Mehta, P.; Mujawar, M.A.; Lafrance, S.; Bernadin, S.; Ewing, D.; Bhansali, S. Editors' Choice—Review—Sensor-Based and Computational Methods for Error Detection and Correction in 3D Printing. *ECS Sensors Plus* **2024**, *3*, 030602. [\[CrossRef\]](#)
41. Abdolahi, A.; Soroush, H.; Khodaygan, S. Process capability analysis of additive manufacturing process: A machine learning-based predictive model. *Rapid Prototyp. J.* **2025**, *31*, 724–741. [\[CrossRef\]](#)
42. Zhao, M.; Xiong, G.; Wang, W.; Fang, Q.; Shen, Z.; Wan, L.; Zhu, F. A Point-Based Neural Network for Real-Scenario Deformation Prediction in Additive Manufacturing. In Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), Mexico City, Mexico, 20–24 August 2022; pp. 1656–1661. [\[CrossRef\]](#)
43. Hopfield, J.J. Artificial neural networks. *IEEE Circuits Devices Mag.* **1988**, *4*, 3–10. [\[CrossRef\]](#)
44. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
45. Pu, Y.; Gan, Z.; Henao, R.; Yuan, X.; Li, C.; Stevens, A.; Carin, L. Variational autoencoder for deep learning of images, labels and captions. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 2360–2368.
46. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaria, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [\[CrossRef\]](#)
47. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#)
48. Geirhos, R.; Temme, C.R.; Rauber, J.; Schütt, H.H.; Bethge, M.; Wichmann, F.A. Generalisation in humans and deep neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7549–7561.
49. Garcia-Garcia, A.; Gomez-Donoso, F.; Garcia-Rodriguez, J.; Orts-Escolano, S.; Cazorla, M.; Azorin-Lopez, J. Pointnet: A 3D convolutional neural network for real-time object class recognition. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 1578–1584.
50. Hanocka, R.; Hertz, A.; Fish, N.; Giryes, R.; Fleishman, S.; Cohen-Or, D. Meshcnn: A network with an edge. *ACM Trans. Graph. (TOG)* **2019**, *38*, 1–12. [\[CrossRef\]](#)
51. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5105–5114.
52. Song, S.; Lichtenberg, S.P.; Xiao, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 567–576.
53. Tan, Q.; Gao, L.; Lai, Y.K.; Xia, S. Variational autoencoders for deforming 3D mesh models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5841–5850.
54. Jadayel, M.; Khameneifar, F. Improving Geometric Accuracy of 3D Printed Parts Using 3D Metrology Feedback and Mesh Morphing. *J. Manuf. Mater. Process.* **2020**, *4*, 112. [\[CrossRef\]](#)
55. Bentley, J.L. Multidimensional binary search trees used for associative searching. *Commun. ACM* **1975**, *18*, 509–517. [\[CrossRef\]](#)
56. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
58. GOM. GOM Acceptance Test, Certificate No. 110826_CP20-170-60086. Acceptance/Reverification Base on VDI/VDE 2634, Part 3. Available online: https://www.zebicon.com/fileadmin/user_upload/2_Maaleudstyr/9_Certifikater/ATOS_Core_200_SN160300/2019-04-02_Acceptance_test_ATOS_Core_MV200_SN160300.pdf (accessed on 12 November 2020).
59. Dawson-Haggerty. Trimesh, 2019. Available online: <https://trimesh.org/> (accessed on 20 March 2022)
60. Jacobson, A.; Panozzo, D. Libigl: A Simple C++ Geometry Processing Library. 2018. Available online: <https://libigl.github.io/> (accessed on 17 September 2022).
61. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: San Francisco, CA, USA, 2019; pp. 8024–8035.
62. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)

63. Wang, Q.; Ma, Y.; Zhao, K.; Tian, Y. A Comprehensive Survey of Loss Functions in Machine Learning. *Ann. Data Sci.* **2022**, *9*, 187–212. [CrossRef]
64. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
65. Biewald, L. Experiment Tracking with Weights and Biases. 2020. Available online: <https://wandb.ai/site> (accessed on 20 March 2022).
66. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
67. Darken, C.; Chang, J.; Moody, J. Learning rate schedules for faster stochastic gradient search. In Proceedings of the Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop, Helsingør, Denmark, 31 August–2 September 1992; pp. 3–12. [CrossRef]
68. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv* **2017**, arXiv:1611.03530. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.