applied sciences

MDPI

Article

Transfer Learning-Based Piezoelectric Actuators Feedforward Control with GRU-CNN

Yaqian Hu [1], Herong Jin [1,*], Xiangcheng Chu [2,*] and Yali Yi [1]

[1] School of Mechanical Engineering, Yanshan University, Qinhuangdao 066004, China
[2] State Key Laboratory of New Ceramics and Fine Processing, School of Materials Science and Engineering, Tsinghua University, Beijing 100084, China
* Correspondence: ysujhr@ysu.edu.cn (H.J.); chuxiangcheng@tsinghua.edu.cn (X.C.)

Abstract

To compensate for hysteresis, low damping vibration, and their coupling effects, this paper proposes a gated recurrent unit and convolutional neural network (GRU-CNN) model as a feedforward control model that maps desired displacement trajectories to driving voltages. The GRU-CNN integrates a gated recurrent unit (GRU) layer to capture long-term temporal dependencies, a multi-layer convolutional neural network (CNN) to extract local data features, and residual connections to mitigate information distortion. The GRU-CNN is then combined with transfer learning (TL) for feedforward control of cross-batch and cross-type piezoelectric actuators (PEAs), so as to reduce reliance on training datasets. The analysis focuses on the impacts of target PEA data volume and source-target similarity on transfer learning strategies. The GRU-CNN trained on PEA #1 achieves high control accuracy, with a mean absolute error (MAE) of 0.077, a root mean square error (RMSE) of 0.129, and a coefficient of determination (R^2) of 0.997. When transferred to cross-batch PEA #2 and cross-type PEA #3, the GRU-CNN feedforward controller still delivers favorable performance; R^2 values all exceed 0.98, representing at least a 27% improvement compared to training from scratch. These results indicate that the proposed transfer learning-based feedforward control method can effectively reduce retraining effort, suggesting its potential applicability to batch production scenarios.

Keywords: piezoelectric actuator; hysteresis; transfer learning; feedforward control

Check for updates

Academic Editor: Douglas O'Shaughnessy

Received: 31 December 2025
Revised: 20 January 2026
Accepted: 24 January 2026
Published: 27 January 2026

Copyright: © 2026 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.

1. Introduction

As a type of precision positioning device, piezoelectric actuators (PEAs) are widely used in high-precision manufacturing. Compared with traditional actuators, they exhibit superior performance, including high resolution, large output force, and fast response [1]. However, the hysteretic behavior of PEAs, which is characterized by time-variability, unsteadiness, and nonlinearity, can significantly degrade positioning accuracy. In particular, variations in the excitation frequency lead to changes in both displacement responses and hysteresis loop morphology. This phenomenon has been reported to cause inconsistent deformation in actuator components and consequent positioning instability [2]. Therefore, the development of effective control models is essential for improving the positioning performance of PEAs.

Historically, hysteresis models have been primarily derived from the physical properties of piezoelectric materials. These mechanism-based models describe hysteresis by exploiting physical quantities such as stress, strain, magnetic flux density, and electric

field strength, with the aim of capturing the intrinsic origins of hysteretic behavior. Representative models include the Jiles–Atherton [3], Maxwell [4], and Dahl [5]. However, the multi-physics nature of practical engineering systems introduces numerous unknown parameters, complex computational procedures, and challenges in measuring physical quantities. As a result, purely mechanism-based models often exhibit limited flexibility and accuracy when applied to hysteresis compensation. In response, researchers have increasingly turned to phenomenological models, which characterize hysteresis loops without explicitly modeling the underlying physical mechanisms. Classical phenomenological models include the Preisach [6], Prandtl–Ishilinskii [7], and Bouc–Wen [8]. Although these models offer improved flexibility, their accuracy can degrade when handling complex excitation signals. To better capture frequency-dependent hysteresis effects, dynamic extensions of classical models have been proposed. For example, Dong et al. [9] extended the input space by incorporating generalized gradients and input derivatives, but this approach requires complex parameter identification and manual feature extraction. More recently, neural-network-based models have attracted growing attention for hysteresis modeling and control. Serpico and Visone [10] demonstrated the feasibility of applying feed-forward neural networks (NNs) to represent rate-independent hysteresis behavior. Adly and Abd-El-Hafiz [11] proposed a hybrid model that combines the Preisach model with a neural network to improve modeling accuracy. Yan et al. [12] further observed that hysteresis exhibits long-term memory behavior, and thus advocated the use of long short-term memory (LSTM) networks to capture hysteretic nonlinearities over a wide range of operating frequencies.

Despite these advances, two critical challenges remain. Firstly, the coupling of hysteresis and low damping vibration in PEAs complicates the morphology of hysteresis loops, posing significant difficulties for accurate nonlinear control. Existing models, such as the Hammerstein model [13], typically simplify non-hysteretic dynamics into a second-order linear system, which may be insufficient to comprehensively capture all factors affecting PEA positioning accuracy. Although Cheng et al. [14] established two nonlinear autoregressive moving average with exogenous input (NARMAX) models to represent cascaded hysteresis and dynamic subsystems; their approach requires two distinct types of excitation signals for system identification, thereby increasing the data acquisition burden. More importantly, the coupling effect between hysteresis and low-damping vibration is not explicitly addressed in their framework. Secondly, in cross-batch and cross-type PEA scenarios, neural network-based models require sufficient experimental data from each individual actuator for effective training. This results in repeated and time-consuming training processes whenever a new actuator is introduced, which is impractical in batch production settings. Transfer learning (TL) provides a promising solution to this challenge. When two actuators exhibit strong correlations, TL enables the transfer of knowledge learned from a source actuator to a target actuator, thereby improving learning efficiency and reducing the need for training from scratch. For example, Shao et al. [15] used a pre-trained VGG-16 network and fine-tuned it using time–frequency graphs obtained via wavelet transform to achieve cross-domain fault diagnosis. Liu et al. [16] verified the feasibility of TL for fault diagnosis across different metal 3D printers. Li et al. [17] applied TL to new materials to alleviate low prediction accuracy in magnetic component power loss modeling. However, few studies, such as [18–20], have explored the application of transfer learning in PEA systems, and there has been no investigation into how different transfer-learning strategies or target-domain data volumes influence model performance in cross-batch and cross-type PEA scenarios.

To investigate the feasibility and effectiveness of transfer learning strategies for piezoelectric actuators, this study proposes a gated recurrent unit-convolutional neural network

(GRU-CNN) combined with transfer learning for feedforward control of PEA nonlinearities. The main contributions of this work are summarized as follows:

(1) Feedforward control of coupled nonlinearities. To address the coupling between hysteresis and low damping vibration, a GRU-CNN-based nonlinear feedforward controller is developed to improve displacement control accuracy. The proposed model integrates a gated recurrent unit (GRU) layer to capture long-term temporal dependencies and a multi-layer convolutional neural network (CNN) to extract local data features. In addition, residual connections are incorporated to mitigate information distortion and facilitate effective gradient propagation. Experimental results demonstrate that the proposed controller can achieve accurate compensation of hysteresis, low-damping vibration, and their coupling, with a mean absolute error (MAE) of 0.077, a root mean square error (RMSE) of 0.129, and a coefficient of determination (R^2) of 0.997.

(2) Analysis of transfer-learning strategies for cross-batch and cross-type PEAs. This study investigates the potential of transfer learning for cross-batch and cross-type PEAs by considering four representative transfer-learning strategies and analyzing the effects of target-domain data volume and source-target similarity on transfer performance. For the cross-batch PEA #2 with a limited target dataset of 125 samples, fine-tuning selected layers of the pre-trained model yields substantially improved performance compared with training from scratch, resulting in an 85% reduction in MAE, an 82% reduction in RMSE, and a 27% increase in R^2. For the cross-type PEA #3 under the same data constraint, fine-tuning the pre-trained model achieves an 83% reduction in MAE, an 81% reduction in RMSE, and a 46% increase in R^2 relative to training from scratch. These results highlight the effectiveness of transfer learning in reducing data requirements and retraining effort.

2. Piezoelectric Actuator System

2.1. Piezoelectric Actuator

The piezoelectric actuator is employed to provide precise uniaxial motion. It integrates two multi-layer stacked piezoelectric ceramics P-887.91 (Physik Instrumente Ceramic GmbH, Lederhose, Thuringia, Germany), which drive a lever-type compliant amplification mechanism to generate motion along the target axis, as shown in Figure 1a. When excited by an applied voltage ranging from 0 V to 150 V, the actuator achieves a maximum travel range of 2.21 mm. As shown in Figure 1b, PEAs #1 and #2 have identical mechanical structures but originate from different production batches. In contrast, PEA #3 differs structurally from the first two, as its shorter compliant amplification mechanism results in a reduced travel range. When a sinusoidal voltage with a frequency of 100 Hz and an amplitude of 105 V is applied, as illustrated in Figure 1c, the dynamic displacement responses of PEAs #1, #2, and #3 exhibit noticeable differences. Although PEAs #1 and #2 share the same structural design, variations in their response data are observed, which can be primarily attributed to assembly tolerances and preload variations.

2.2. Test Platform

The test platform for the PEAs is shown in Figure 2. To measure the displacement of each PEA, a head-spot type position sensor LK-H020 (Keyence Corporation, Osaka, Japan) is employed, featuring a measuring range of ±3 mm and a measuring accuracy of 0.02 μm (±0.02% F.S.). The sensor controller LK-G5001 (Keyence Corporation, Osaka, Japan) provides an analog output proportional to the measured displacement, with a voltage range of 0 V to 10 V. A power amplifier module with a fixed gain of 15 is used to drive the PEAs. The voltage curves are generated via a MATLAB (v2024a) program, with an amplitude range of 0 V to 10 V. Consequently, the output voltage of the power amplifier module ranges from 0 V to 150 V. All voltage and displacement signals are synchronously

acquired using a 16-bit data acquisition card NI USB-6218 (National Instruments, Austin, TX, USA), with the sampling frequency set to 10 k Hz.

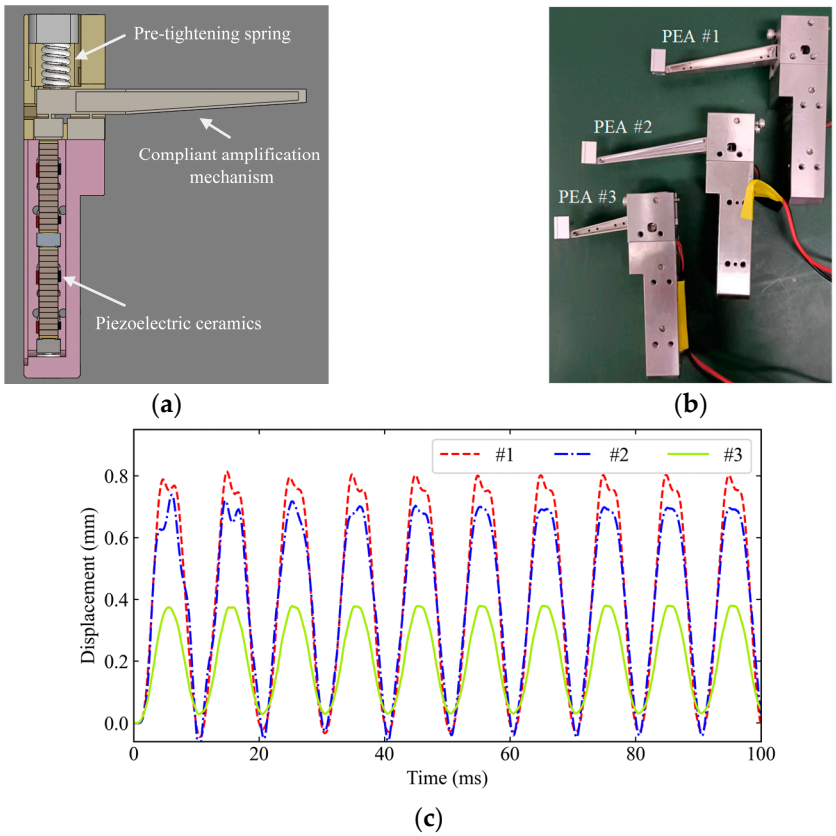


Figure 1. Piezoelectric actuators. (**a**) Mechanical structure; (**b**) physical prototype; (**c**) displacement responses of PEAs #1, #2, and #3 subjected to identical voltage excitation.

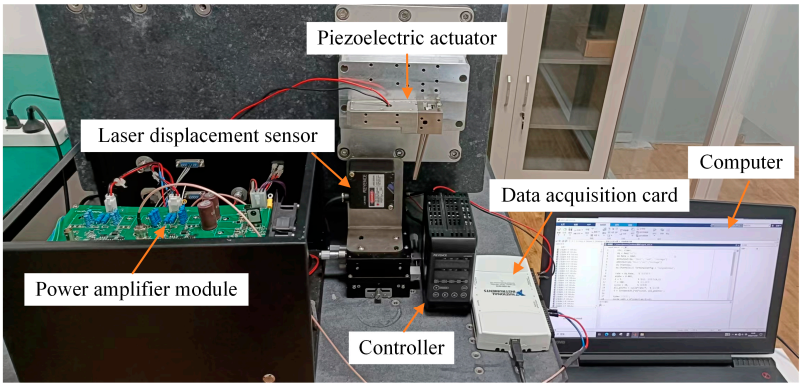
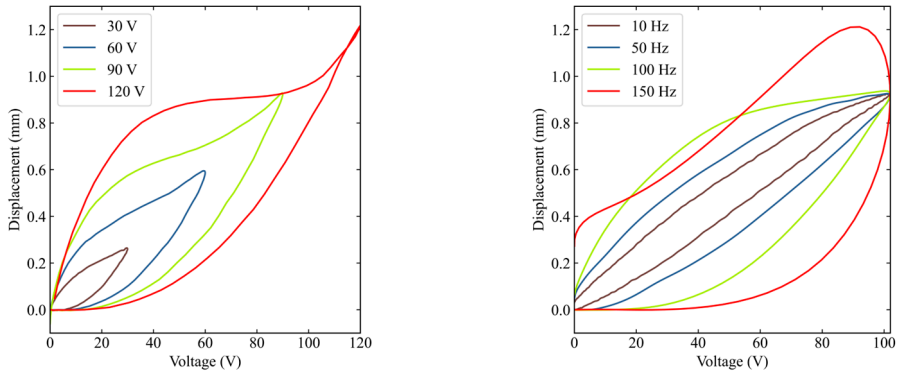


Figure 2. Piezoelectric actuator test platform.

2.3. Nonlinearity Analysis

The dynamic behavior of PEAs is influenced not only by hysteresis but also by system properties such as stiffness and damping [21]. Hysteresis refers to the delay of displacement response relative to voltage variation, which exhibits history dependence, nonlinearity, and non-monotonicity [22]. It has been demonstrated that hysteresis can cause 15–35% open-loop positioning errors [23]. When the excitation frequency is fixed at 100 Hz, Figure 3a shows that the width of the hysteresis loop increases to varying extents as the maximum amplitude of the voltage signal increases. Specifically, for sinusoidal voltages with amplitudes of 30 V, 60 V, 90 V, and 120 V, the corresponding hysteresis loop widths are 43.4%,

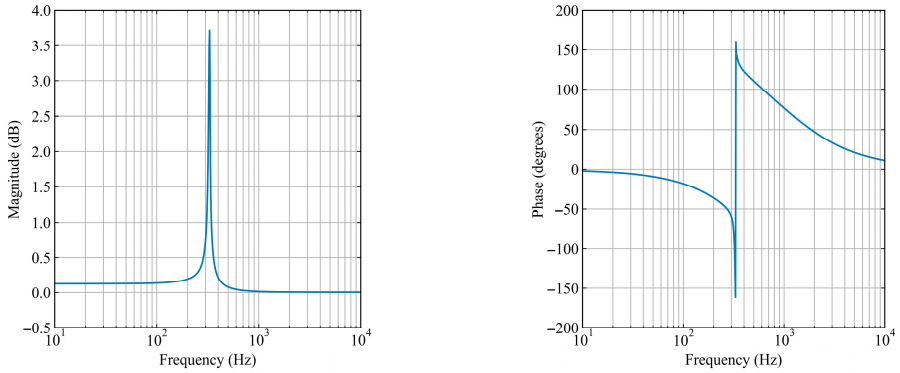
42.9%, 42.7%, and 47.5%, respectively, with displacement ranges of 0.31 mm, 0.70 mm, 1.13 mm, and 1.48 mm. In addition to amplitude dependence, the displacement behavior of PEAs is also strongly frequency-dependent. When the maximum voltage amplitude is fixed at 102 V and the excitation frequency varies, the positioning response changes accordingly. As shown in Figure 3b, with increasing signal frequency, the hysteresis loop of the PEA gradually expands, the hysteresis effect becomes more pronounced, and the displacement range varies. Specifically, for sinusoidal voltages with frequencies of 10 Hz, 50 Hz, 100 Hz, and 150 Hz, the hysteresis loop widths are 20.6%, 40.5%, 69.5%, and 71.8%, respectively, with corresponding displacement ranges of 0.94 mm, 0.95 mm, 1.04 mm, and 1.58 mm.



(**a**) (**b**)

Figure 3. Hysteresis characteristic of the piezoelectric actuator. (**a**) Amplitude correlation; (**b**) frequency correlation.

The influence of low damping vibration on the nonlinear hysteresis behavior of PEAs is minimal when operating at low excitation frequencies, particularly below 10 Hz. However, as the excitation frequency increases, hysteresis control becomes particularly challenging. Preliminary system identification indicates that PEA #1 has a damping ratio of 0.13 at its natural frequency of 2060 rad/s. The amplitude–frequency and phase–frequency characteristics of PEA #1 over a frequency range of 10–1000 Hz are presented in Figure 4. Near the frequency range of 300–400 Hz, the phase–frequency response exhibits a sharp decrease followed by a rapid increase, resulting in extremely poor actuator performance. In addition, as observed in Figure 3b, the shape of the hysteresis curve at 150 Hz differs from those at other frequencies. This deviation is attributed to the influence of low damping vibration, which becomes increasingly significant as the excitation frequency increases.



(**a**) (**b**)

Figure 4. Low damping vibration characteristic of the piezoelectric actuator. (**a**) Amplitude frequency response; (**b**) phase frequency response.

2.4. Datasets Description

The experimental data should include sufficient input–output information of the PEA, covering the entire frequency and amplitude range of interest. Considering the frequency-dependent and amplitude-dependent characteristics of PEAs, the voltage U is synthesized as a set of single-period signals, each defined by a unique pairs (f_i, A_i), where f_i and A_i denote the excitation frequency and voltage amplitude, respectively. The excitation frequencies are limited to below 200 Hz to account for service-life considerations and to reflect typical operating conditions in precision positioning applications. Firstly, 1000 points are uniformly randomly generated on the frequency–amplitude plane, where the frequency range is set to 10–200 Hz and the voltage amplitude range is maintained at 0–150 V. Then, the k-means clustering algorithm [24] is used to select appropriate (f_i, A_i) pairs from these 1000 points. Besides sinusoidal signals, the voltage set U also includes triangular signals, frequency-varying signals, amplitude-varying signals, and random signals. Finally, these voltages U and their corresponding displacements Y are collected from the test platform. Each sample is uniformly discretized into 1000 time points, forming a time series that represents the temporal evolution of the displacement–voltage pair. Additionally, a zero-phase low-pass filter is applied to reduce noise in the displacements Y. The four types of experimental samples used in this study are shown in Figure 5.

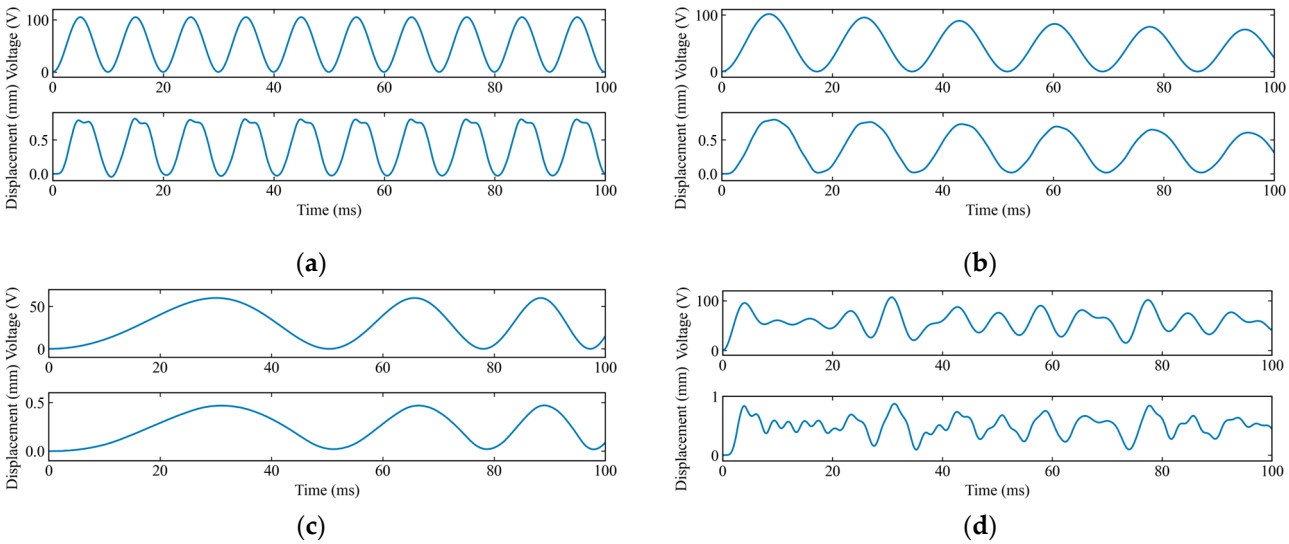


Figure 5. Experimental samples for the piezoelectric actuator. (**a**) Sinusoidal signals; (**b**) amplitude-varying signals; (**c**) frequency-varying signals; (**d**) mixed-frequency signals.

Before model training, these experimental samples are normalized to the range [0, 1] to minimize the impact of variable scales on network training. The normalization formula is given by

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x denotes the experimental sample, x_{max} and x_{min} represent the maximum and minimum values of all experimental samples, respectively, and x^* is the normalized experimental sample.

In this study, PEA #1 serves as the source domain, while PEAs #2 and #3 serve as the target domains. The dataset size for PEA #1 is 15,000 samples, whereas the dataset sizes for PEAs #2 and #3 are set to 65, 125, 500, 2000, and 8000 samples, respectively. These varying dataset sizes correspond to different transfer learning scenarios, which are designed to simulate the performance of the transfer learning model under conditions of different data volume constraints. The performance differences between models are evaluated by

averaging the results of 10-fold cross-validation, which is performed on both the source and target domain datasets for comprehensive comparison. Meanwhile, 1200 untrained test samples are selected from each target domain to verify the generalization ability of the transfer learning models.

3. Methodology

The proposed transfer learning-based feedforward control framework for piezoelectric actuators consists of three main steps. Step 1 describes the data acquisition and preprocessing process for the PEA system. Step 2 presents the GRU-CNN feedforward model trained on the source domain, which learns the mapping from displacement sequences to the corresponding driving voltage. Step 3 introduces the evaluation and selection of transfer-learning strategies under different scenarios. An overview of the complete framework is illustrated in Figure 6.

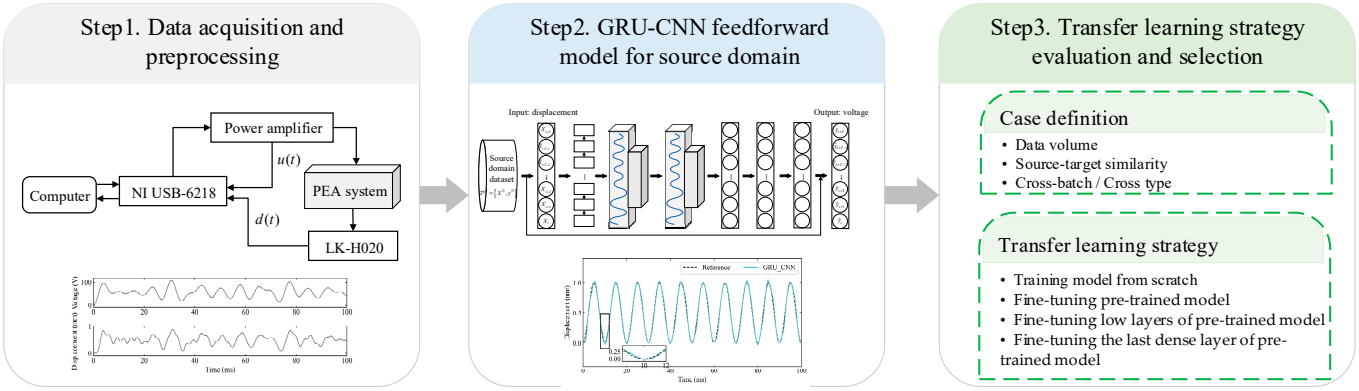


Figure 6. Overview of the proposed transfer learning-based feedforward control framework.

3.1. GRU-CNN

GRU, an improved variant of LSTM, integrates the input and forget gates of LSTM into a separate update gate. This modification reduces the number of training parameters while maintaining predictive performance comparable to LSTM [25]. Its structure is composed of a unit with a reset gate, an update gate, and a hidden state, enabling dynamic processing of input data at each time step and effectively mitigating the gradient vanishing problem. The principle of this unit is as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{3}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{4}$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \tag{5}$$

where x_t is the input at time step t, h_{t-1} and h_t are the previous and current hidden states, r_t and z_t denote to the reset gate and update gate, respectively, $\tilde{h}_t$ is the candidate hidden state, $\sigma(\cdot)$ is to the sigmoid activation function, $\tanh(\cdot)$ is the hyperbolic tangent activation function, and $\odot$ denotes the Hadamard product. W_r, W_z, W_h and U_r, U_z, U_h are the input and recurrent weight matrices, respectively, and b_r, b_z, b_h are bias vectors.

The multi-layer CNN structure is constructed by sequentially stacking one-dimensional convolution layers, activation layers, dropout layers, and residual connections. The convolution layer performs repeated filtering operations to generate feature maps, which characterize specific attributes associated with data points [26]. For this process, the input

is convolved with a set of weights (referred to as kernels), and the operation is mathematically expressed as

$$x_k^l = \sum_{i=1}^{N_l-1} conv1D\left(w_{ik}^{l-1}, s_i^{l-1}\right) + b_k^l \tag{6}$$

where x_k^l is the output of the k^{th} neuron at layer l, b_k^l is the bias of the k^{th} neuron at layer l, s_i^{l-1} represents the output of the i^{th} neuron at layer $l-1$, w_{ik}^{l-1} represents the kernel from the i^{th} neuron at layer $l-1$ to the k^{th} neuron at layer l, $conv1D(\cdot)$ refers to one-dimensional convolution with zero-padding. The outputs x_k^l and s_i^{l-1} have equal dimensions.

Subsequently, x_k^l is fed into the activation layer, where the rectified linear unit (ReLU) [27] is adopted as the activation function. ReLU sets negative inputs to zero and retains non-negative inputs, enabling the model to learn faster from training data and further alleviate the gradient vanishing problem. Its expression is given by

$$y_k^l = \max\left(0, x_k^l\right) \tag{7}$$

where y_k^l denotes the output of the activation layer.

Furthermore, dropout [28] is employed as a regularization technique to enhance the model's generalization performance on unseen data. It works by randomly deactivating a subset of neurons during training to avoid over-reliance on specific features.

The residual connection [29], also known as the jump connection, propagates the input through multiple layers and maximize the fusion of high-order features with low-order features. This design reduces information loss and prevents degradation of learning performance in deep CNNs. The expression of the residual connection is

$$o = \mathrm{F}(x) + C(x) \tag{8}$$

where $F(x)$ represents a series of transformations applied to the input x and $C(x)$ refers to a 1×1 convolution operation.

In this study, multiple samples are grouped into mini-batches and organized as three-dimensional tensors with the format (batch size, sequence length, feature dimension). When both the batch size and feature dimension are set to 1, the input sequence is denoted as $\{x_1, x_2, \ldots, x_t\}$, and the corresponding output sequence is $\{y_1, y_2, \ldots, y_t\}$. The GRU-CNN model aims to establish the mapping between the input and output sequences, which can be expressed as $(y_1, y_2, \ldots, y_t) = f(x_1, x_2, \ldots, x_t)$. The overall architecture of the GRU-CNN model is illustrated in Figure 7. It consists of an input layer, one GRU layer, two CNN layers, three dense layers, and an output layer. The input sequence is first processed by the GRU layer, which captures long-term historical information to model the long-term dependence and nonlinear memory behaviors of hysteresis. Next, the CNN layers apply kernel filtering to the entire output of the GRU layer to extract local trend features. Finally, the three dense layers perform feature integration and output the final predictions. The mean squared error (MSE) is adopted as the objective function to minimize the discrepancy between the actual output sequence $\{y_1, y_2, \ldots, y_t\}$ and the predicted output sequence $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_t\}$.

3.2. Transfer Learning Strategy

TL can generally be categorized into four basic paradigms: instance-based, feature-based, model-based and relational-based [30]. Since the hysteresis law for PEAs can be reflected in the structure and parameters of the model, and the learned knowledge resides in the weights and biases of different layers is inheritable, the model-based TL is used in this study. This knowledge is not only specific to the dataset for an actuator under a single

working condition, but can also be applied to hysteresis control for different PEAs under different working conditions.

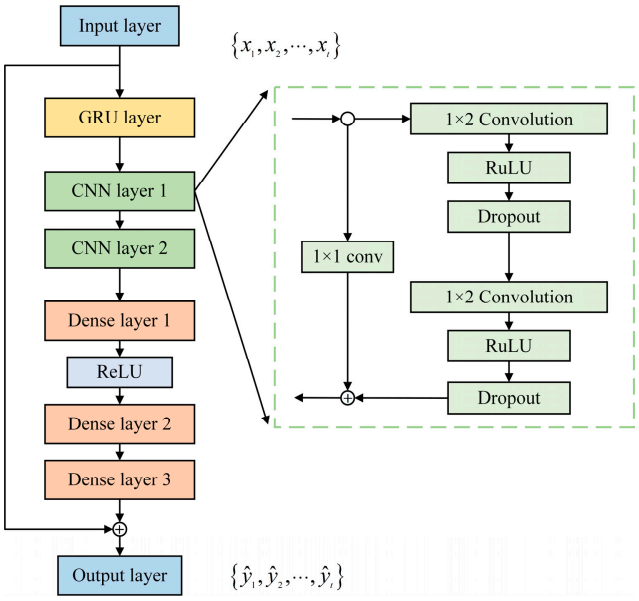


Figure 7. Framework of the GRU-CNN model.

To analyze the impact of different layer parameters processing on TL and verify that TL is an effective improvement, four training strategies are designed. The specific explanations for the four strategies are as follows:

1. Training model from scratch (Model 1). A GRU-CNN model is trained directly on the target PEA datasets.

2. Fine-tuning pre-trained model (Model 2). The pre-trained GRU-CNN model is fine-tuned with its entire structure and parameters fully retained.

3. Fine-tuning low layers of pre-trained model (Model 3). While retaining the structure and parameters of the pre-trained model, the parameters of the GRU and CNN layers are frozen, and the remaining network parameters are retrained using target PEA data.

4. Fine-tuning the last dense layer of pre-trained model (Model 4). On the basis of retaining the pre-trained model's structure and parameters, the parameters of the GRU layers, CNN layers, Dense layer 1, and Dense layer 2 are preserved, while only the parameters of Dense layer 3 are retrained.

4. Experimental Results and Analysis

4.1. Validation of the GRU-CNN Model Trained on PEA #1

This section validates the effectiveness of the proposed GRU-CNN model through ablation and contrast experiments. The model is deployed as an offline-trained feedforward controller for compensating hysteresis and low-damping vibration of PEAs. The input to the model is the desired displacement trajectory, and the output is the driving voltage required to achieve this trajectory.

4.1.1. Ablation Experiment

To ensure robust performance across diverse data partitions, 10-fold cross-validation is implemented for all models. The performance discrepancies between models are evaluated by averaging the results of 10 validation trials. All models are trained using PyTorch 1.10.2-based deep learning packages on a computer equipped with one NVIDIA GTX 3080Ti GPU. A fixed random seed of 1024 is set to maintain consistent initialization parameters across

training sessions. The number of training epochs, the batch size, and the initial learning rate are set to 100, 100, and 0.001, respectively. The Adamax optimizer [31] is employed to update the model parameters, with exponential decay rates set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is dynamically adjusted according to the MSE on the validation set. If no improvement is observed for 10 epochs, the learning rate is reduced by multiplying it by 0.9.

An empirically guided hyperparameter optimization algorithm is adopted for the GRU-CNN model. The hyperparameter search ranges are defined based on prior studies and preliminary experiments, including a GRU hidden neuron number ranging from 15 to 55, CNN convolution kernel sizes ranging from 2 to 7, dropout rates ranging from 0.1 to 0.5, and dense layer neuron numbers ranging from 5 to 45. The final hyperparameter configuration is determined through cross-validation to balance control performance and model complexity, and the optimal settings yielding the best control performance are summarized in Table 1.

Table 1. Hyper-parameter setting of the GRU-CNN model.

Layer	Parameter	Input Dim	Output Dim
Input layer			100 × 1000 × 1 *
GRU layer	hidden neuron 35	100 × 1000 × 1	100 × 1000 × 35
CNN layer 1	kernel 2, stride 1, padding 1, channels 1000, dropout 0.1	100 × 1000 × 35	100 × 1000 × 35
CNN layer 2	kernel 2, stride 1, padding 1, channels 1000, dropout 0.1	100 × 1000 × 35	100 × 1000 × 35
Dense layer 1	neuron 25	100 × 1000 × 35	100 × 1000 × 25
Dense layer 2	neuron 25	100 × 1000 × 25	100 × 1000 × 25
Dense layer 3	neuron 1	100 × 1000 × 25	100 × 1000 × 1
1 × 1 conv	kernel 1, stride 1, padding 0 channels 1000	100 × 1000 × 35	100 × 1000 × 35
Output layer			100 × 1000 × 1

* 100 denotes the batch size, 1000 denotes the temporal length of each sample, and 1 denotes the feature dimension. The channels parameter in the CNN layers correspond to feature mappings along the temporal dimension.

Ablation experiments are conducted to validate the effectiveness of the GRU-CNN architecture for feedforward control of PEAs. Three comparison models are considered, including GRU, CNN and CNN-GRU. Model performance is evaluated using the MAE, RMSE, and R^2. These metrics are computed in the voltage domain and quantify the discrepancy between the driving voltage predicted by the feedforward model and the actual voltage applied to the PEA. Since the predicted voltage directly determines the PEA input, higher voltage prediction accuracy leads to improved displacement tracking performance. To ensure a fair comparison, the training dataset and test dataset used for the GRU, CNN and CNN-GRU are identical to those used for the GRU-CNN model. Train time is defined as the time required for a model to complete one training epoch on the training dataset, while test time refers to the inference time required to process the entire test dataset. GPU memory consumption represents the peak memory usage during model training. The results of the ablation experiments are summarized in Table 2.

Table 2. Ablation experiment results.

Models	GPU (MiB)	Train Time	Test Time	Evaluation Metrics		
				MAE	RMSE	R^2
GRU	1880	4.05 s	0.39 s	0.392 ± 0.030	0.619 ± 0.047	0.920 ± 0.010
CNN	2442	0.78 s	0.10 s	0.157 ± 0.010	0.237 ± 0.019	0.988 ± 0.002
CNN-GRU	2648	5.23 s	0.42 s	0.243 ± 0.111	0.358 ± 0.134	0.972 ± 0.024
GRU-CNN	2843	6.39 s	0.53 s	0.090 ± 0.004	0.147 ± 0.008	0.996 ± 0.001

As shown in Table 2, the GRU-CNN model achieves the best overall performance across all evaluation metrics, yielding the lowest MAE and RMSE as well as the highest R^2. This performance gain can be attributed to the complementary strengths of the GRU and CNN components. The GRU layer enhances the model's ability to capture long-term temporal dependencies in time-series data, while the CNN layers effectively extract local feature patterns. In contrast, the GRU model exhibits inferior performance, which can be attributed to its limited capability in capturing complex feature interactions. The CNN-GRU model performs less effectively than both the CNN and GRU-CNN models, possibly because convolutional feature extraction may disrupt temporal continuity that is critical for GRU-based sequence modeling. Although the CNN model demonstrates superior training and inference efficiency, indicating higher parameter efficiency, its prediction accuracy is lower than that of the proposed GRU-CNN model. Despite higher computational costs, including a GPU memory consumption of 2843 MiB, a train time of 6.39 s per epoch, and a test time of 0.53 s for the entire test set, the GRU-CNN model provides substantially improved prediction accuracy, demonstrating a favorable trade-off between computational cost and control performance.

4.1.2. Contrast Experiment

To further evaluate the effectiveness of the proposed GRU-CNN feedforward control model, a comparative study is conducted against several representative neural network architectures, including multilayer perceptron (MLP) [32], long short-term memory (LSTM) [33], temporal convolutional network (TCN) [34], LSTMseq2seq [35], and Transformer [36]. The architectures of these comparison models are illustrated in Figure 8, and their hyperparameters are determined through empirical tuning to ensure reasonable performance. Specifically, the MLP consists of three dense layers with dimensions of 2500–2500–1000, using the ReLU activation function throughout. The LSTM model comprises four stacked LSTM layers, each with a hidden state dimension of 100, followed by a dense output layer. The TCN is constructed using five residual blocks with dilation factors of 1, 2, 4, 8, and 16, respectively. The convolution kernel size is fixed at 7, and the channel dimensions follows a descending sequence of [128, 64, 32, 16, 8]. The LSTMseq2seq model includes both an encoder and a decoder, each composed of four LSTM layers with a hidden state dimension of 100, followed by a dense output layer. The Transformer model is implemented using a three-layer encoder with two attention heads, a hidden dimension of 150, and an embedding dimension of 50. For a fair comparison, all models except the Transformer are trained using the same training configuration, with 100 training epochs, a batch size of 100, and an initial learning rate of 0.001. Due to the higher optimization complexity of the attention mechanism, the Transformer is trained for 300 epochs while maintaining the same batch size and learning rate.

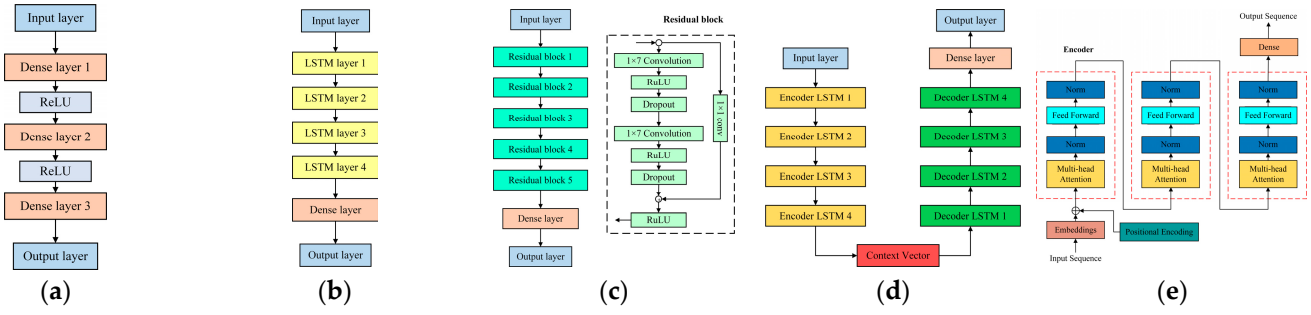


Figure 8. Structural makeup of models in contrast experiment. (**a**) MLP; (**b**) LSTM; (**c**) TCN; (**d**) LSTM-seq2seq; (**e**) Transformer.

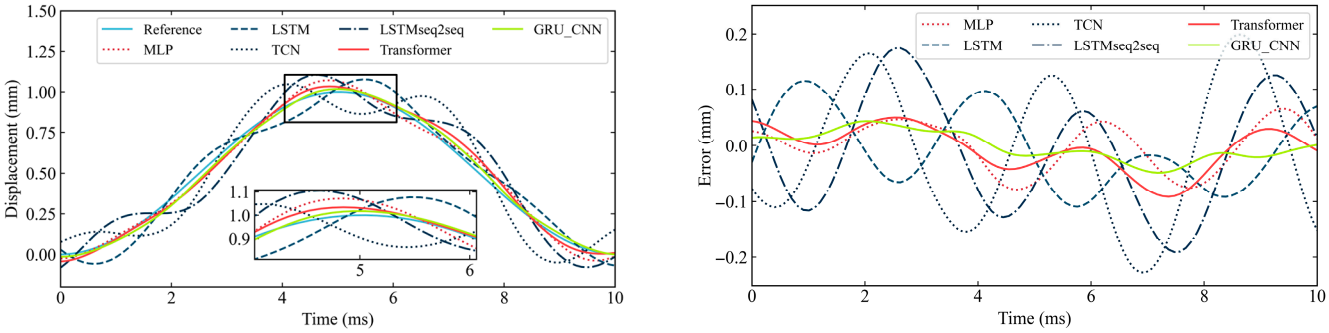
The quantitative comparison results are summarized in Table 3. Overall, the GRU-CNN model achieves superior control performance compared with the contrast models. Although the MLP exhibits lower GPU memory consumption as well as shorter train and test times, its purely fully connected structure lacks the ability to effectively model temporal dependencies, resulting in substantially higher MAE and RMSE values and a lower R^2. Compared with the LSTMseq2seq model, the GRU-CNN achieves significantly reduced train time while improving prediction accuracy by more than 40%. In the LSTMseq2seq framework, the encoder compresses the entire input sequence into a fixed-length context vector. When the input sequence length reaches 1000 points, this representation may become insufficient to preserve detailed temporal information, leading to reduced prediction accuracy. The Transformer model exhibits very high GPU memory consumption, reaching 11,313 MiB, which limits its suitability for practical PEA control applications, particularly in embedded or resource-constrained environments. In contrast, the GRU-CNN requires only 2843 MiB of GPU memory, approximately one-quarter of that required by the Transformer. For capturing the local nonlinear characteristics of PEAs, the localized feature extraction capability of CNN layers is more targeted than the global attention mechanism employed by the Transformer, which may introduce interference from less relevant information. Although the inclusion of CNN layers increases the computational cost of the GRU-CNN model in terms of train time and test times, the proposed model consistently achieves lower MAE, lower RMSE, and higher R^2 values than the other comparison methods. Overall, the GRU-CNN provides a favorable balance among control accuracy, computational efficiency, and hardware adaptability. These results indicate that the proposed model is effective in characterizing the nonlinear relationship between displacement and driving voltage in PEAs.

Table 3. Contrast experiment results.

Models	GPU (MiB)	Train Time	Test Time	Evaluation Metrics		
				MAE	RMSE	R^2
MLP	1585	0.30 s	0.07 s	0.198 ± 0.023	0.274 ± 0.023	0.984 ± 0.003
LSTM	2761	15.05 s	0.98 s	0.221 ± 0.027	0.347 ± 0.028	0.975 ± 0.003
TCN	2792	2.99 s	0.48 s	0.220 ± 0.019	0.351 ± 0.027	0.975 ± 0.004
LSTMseq2seq	3996	120.23 s	4.77 s	0.215 ± 0.016	0.329 ± 0.022	0.978 ± 0.003
Transformer	11,313	14.44 s	0.80 s	0.154 ± 0.023	0.230 ± 0.025	0.989 ± 0.002
GRU-CNN	2843	6.39 s	0.53 s	0.090 ± 0.004	0.147 ± 0.008	0.996 ± 0.001

For a more detailed comparison, Figure 9a presents the 100 Hz sinusoidal reference trajectory together with the measured trajectories obtained using different comparison models. The corresponding control deviations between the reference and measured trajectories are shown in Figure 9b. The measured trajectory produced by the LSTM model exhibits noticeable fluctuations around the reference trajectory, whereas the GRU-CNN model reduces the mean absolute error (MAE) by more than 57% relative to the LSTM model. Overall, the measured trajectory generated by the GRU–CNN model shows closer agreement with the reference trajectory. The controlling deviation ranges for the MLP, LSTM, TCN, LSTMseq2seq, Transformer, and GRU-CNN models are [−0.08 mm, 0.07 mm], [−0.11 mm, 0.12 mm], [−0.23 mm, 0.05 mm], [−0.19 mm, 0.17 mm], [−0.09 mm, 0.09 mm], and [−0.05 mm, 0.04 mm], respectively. While the control deviation range of the GRU-CNN model is comparable to that of the Transformer model in certain regions, it exhibits smaller deviations in others. Although all comparison models are capable of reducing trajectory tracking errors to some extent, the GRU-CNN model consistently achieves lower devia-

tion. These results further indicate the effectiveness of the proposed GRU-CNN model for feedforward control.



(**a**) (**b**)

Figure 9. Controlling performance of different models for 100 Hz sinusoidal reference trajectory. (**a**) Trajectory details; (**b**) trajectory deviations.

4.1.3. Verification of Generalization and Robustness

Generalization and robustness are important criteria for evaluating the practical applicability and reliability of a control model. To assess the generalization capability of the proposed GRU-CNN model, the network trained on the dataset of PEA #1 is directly evaluated on data collected from a different production batch. Specifically, the test set of PEA #2, consisting of 1200 samples, is used as the cross-batch generalization validation dataset. The generalization performance of the GRU-CNN model and the comparison model is summarized in Table 4. On the PEA #2 test set, the GRU-CNN model achieves an MAE of 0.2093, an RMSE of 0.3745, and an R^2 of 0.9752, representing the best overall performance among the evaluated models in terms of lower error and higher goodness of control. In addition, the relative differences between these metrics and those obtained on the PEA #1 test set are less than 57%, 60%, and 2%, respectively. These results indicate that the proposed GRU-CNN model does not excessively depend on the specific data distribution of the training PEA. Instead, it appears to capture representative characteristics of PEA operating behavior, enabling it to maintain stable performance when applied to data from different production batches. This observation suggests that the proposed model exhibits cross-batch generalization capability.

Table 4. Generalization verification results.

Metrics	MLP	LSTM	TCN	LSTMseq2seq	Transformer	GRU-CNN
MAE	0.337 ± 0.052	0.336 ± 0.036	0.373 ± 0.009	0.346 ± 0.027	0.223 ± 0.037	0.209 ± 0.014
RMSE	0.517 ± 0.059	0.570 ± 0.032	0.628 ± 0.009	0.576 ± 0.029	0.384 ± 0.036	0.375 ± 0.027
R^2	0.953 ± 0.012	0.943 ± 0.005	0.930 ± 0.001	0.941 ± 0.004	0.974 ± 0.005	0.975 ± 0.005

In practical application scenarios, operational data of PEAs collected by sensors are inevitably affected by electromagnetic interference and environmental noise, resulting in random disturbances in the input data. To quantitatively evaluate the robustness of the proposed GRU-CNN model against input data perturbations, the test dataset of PEA #1 is used as the benchmark. Multiple noisy test datasets are generated by injecting Gaussian noise with different intensities into the benchmark data, with the aim of analyzing the relationship between noise intensity and model performance degradation. The noise intensities are set to 1%, 2%, and 5% of the standard deviation of the benchmark dataset, where noise levels of 1–2% represent mild interference scenarios and 5% represents a relatively strong interference scenario. The performance of the GRU-CNN model under

the benchmark condition and the three noise levels is summarized in Table 5. Under Gaussian noise interference with intensities of 1% and 2%, the degradation rates of the model's MAE remain within 1%, while the degradation rates of the RMSE do not exceed 2%. These results indicate that when the input data are subject to small perturbations, the proposed model is able to maintain stable prediction accuracy without significant performance degradation. However, under a higher noise intensity of 5%, the degradation rates of both MAE and RMSE exceed 5%, indicating a noticeable decline in prediction accuracy. This observation suggests that although the GRU-CNN model exhibits a certain degree of robustness to low-level noise interference, its performance is more sensitive under stronger noise disturbances.

Table 5. Robustness verification results of GRU-CNN model.

Metrics	Benchmark	1% Noisy	2% Noisy	5% Noisy
MAE	0.090 ± 0.008	0.090 ± 0.008	0.091 ± 0.008	0.100 ± 0.007
RMSE	0.147 ± 0.010	0.148 ± 0.011	0.149 ± 0.011	0.158 ± 0.010
R^2	0.996 ± 0.001	0.996 ± 0.001	0.995 ± 0.001	0.995 ± 0.001

4.2. The Pre-Trained GRU-CNN Model Transferred to Other PEAs

To reduce train time while maintaining displacement control accuracy, transfer learning is employed to adapt the structure and parameters of a GRU-CNN model pre-trained on PEA #1 to other actuators. For PEAs #2 and #3, six graded dataset sizes, namely 65, 125, 500, 2000, and 8000 samples, are considered to represent small, medium, and large data scenarios. The similarity between the source and target domains is quantified using the Proxy A-distance (PAD) [37]. The PAD measures domain discrepancy by training a classifier to distinguish samples from two domains and evaluating the discrepancy based on the classifier error rate on a held-out test set. In this study, the PAD is computed using the minimum test error of a linear domain classifier based on a support vector machine (SVM), with the regularization parameter selected from $\left[10^{-5}, 10^{4}\right]$. Larger PAD values indicate greater domain discrepancy, whereas smaller PAD values correspond to higher domain similarity. To further analyze the relationship between data volume and domain similarity, the PAD results for different domain pairs are summarized in Table 6. In addition, Figure 10 shows t-SNE visualization of the original data features for the three PEAs. It can be clearly seen that there exist obvious data distribution discrepancy among the actuators. Domain pairs with higher PAD values exhibit clearer separation in the embedding space, whereas those with lower PAD values show stronger overlap, indicating higher domain similarity.

Table 6. Case settings under different data volume and domain similarity.

Case	Volume	PAD (PEA #1 vs. PEA #2)	PAD (PEA #1 vs. PEA #3)
1	62	1.981	1.981
2	125	1.966	1.968
3	500	1.871	1.897
4	2000	1.835	1.894
5	2000	1.529	1.708
6	8000	1.108	1.553

Different transfer strategies are selected based on the data volume scale and source-target domain similarity. For each transfer scenario, the most suitable strategy is identified by comparing the performance of the four transfer-learning models when constructing the feedforward control model for the target PEA. All models are trained using the Adamax

optimizer for 100 epochs with a batch size of 100. Models 1–4 are initialized with an initial learning rate of 0.005. During training, the learning rate is automatically adjusted based on the model's performance on the validation set. If the validation performance does not improve for 10 epochs, the learning rate is reduced by a factor of 0.5. In addition, an early stopping mechanism is employed to prevent overfitting. The experimental results illustrating the effects of data volume and source–target domain similarity on the four transfer-learning strategies for PEAs #2 and #3 are summarized in Table 7.

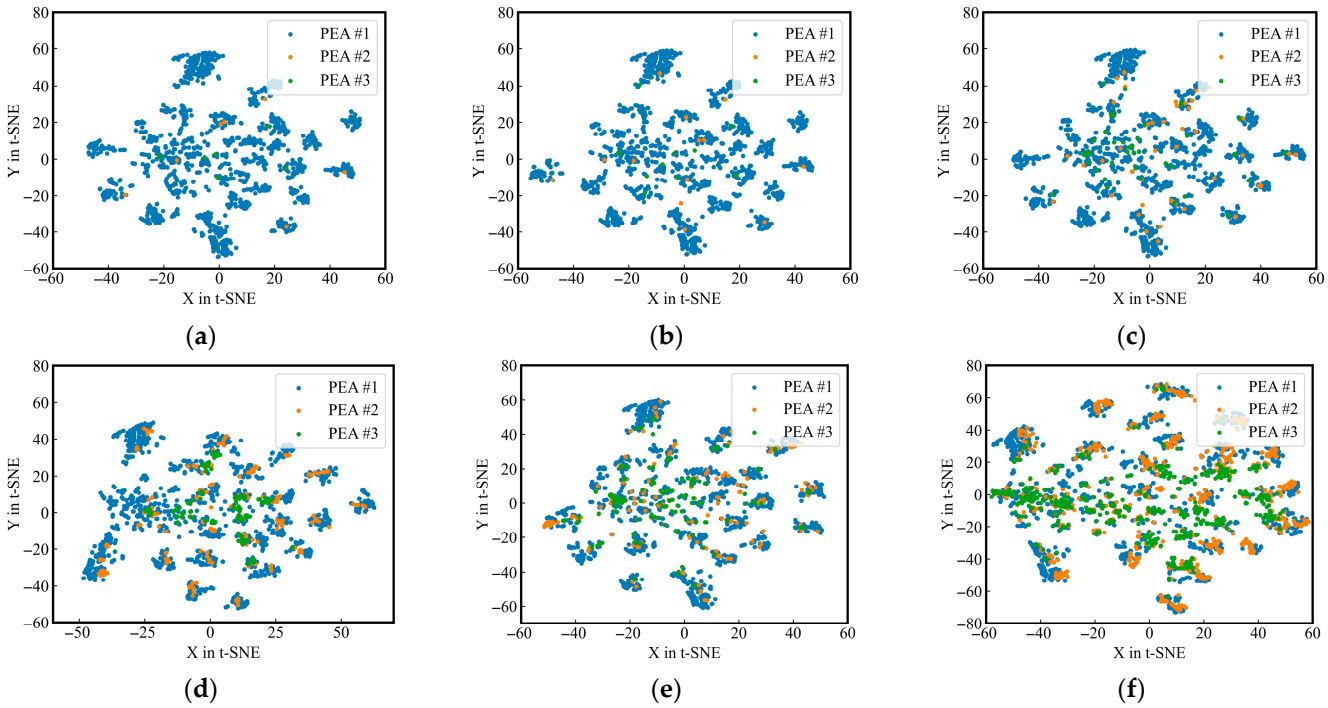


Figure 10. The t-SNE visualization of the original data features for three PEAs. (**a**) Scenario 1; (**b**) Scenario 2; (**c**) Scenario 3; (**d**) Scenario 4; (**e**) Scenario 5; (**f**) Scenario 6.

In terms of training efficiency, Model 1 and Model 2 exhibit comparable train times under the same data volume, as both require updating all network parameters. In contrast, Model 3 and Model 4 achieve shorter train times because parameters in key layers are frozen, reducing the number of trainable parameters. Regarding control accuracy, the performance of all models is closely influenced by both the size of the target dataset and the similarity between the source and target domains. Model 1, which is trained from scratch without leveraging pre-trained knowledge, shows the highest sensitivity to data volume. When the dataset size does not exceed 2000 samples, Model 1 performs the worst among all models, indicating insufficient learning of representative features. As the data volume increases to 8000 samples, its performance improves markedly. This observation highlights the benefit of transfer learning in reducing data dependence and training overhead in data-scarce scenarios.

For cross-batch transfer, Model 3 achieves the best performance when the dataset size is limited to 65–125 samples. For instance, when the target dataset contains 125 samples, Model 3 achieves an 85% reduction in MAE, an 82% reduction in RMSE, and a 27% increase in R^2 compared with Model 1. By freezing the GRU and CNN layers and only adapting the dense layers, Model 3 effectively mitigates overfitting and achieves better generalization. When the data volume increases to 500 samples, Model 2 slightly outperforms Model 3, suggesting that moderate data availability enables full fine-tuning to better capture inter-domain feature variations. Notably, when the data volume reaches 2000 samples and

the PAD value decreases, the performance of Models 2 and 3 deteriorates simultaneously, indicating that domain similarity plays a non-negligible role in transfer effectiveness. Overall, the data volume at which Model 2 surpasses Model 3 is not fixed but increases with the availability of target-domain data.

Table 7. Performance comparison of transfer learning strategies under different cases.

Case	Method	Train Time	Cross-Batch Transfer			Cross-Type Transfer		
			MAE	RMSE	R^2	MAE	RMSE	R^2
-	Pre-trained	6.39 s	0.209	0.375	0.975	1.279	1.903	0.412
1	Model 1	0.04 s	1.034 ± 0.074	1.363 ± 0.048	0.672 ± 0.024	1.449 ± 0.179	1.824 ± 0.166	0.455 ± 0.103
	Model 2	0.04 s	0.340 ± 0.031	0.529 ± 0.026	0.951 ± 0.005	0.399 ± 0.068	0.590 ± 0.084	0.942 ± 0.016
	Model 3	0.02 s	0.149 ± 0.015	0.234 ± 0.020	0.990 ± 0.002	0.273 ± 0.009	0.397 ± 0.016	0.974 ± 0.002
	Model 4	0.02 s	0.227 ± 0.018	0.334 ± 0.022	0.980 ± 0.003	0.465 ± 0.211	0.635 ± 0.300	0.921 ± 0.099
2	Model 1	0.08 s	0.910 ± 0.107	1.241 ± 0.124	0.726 ± 0.052	1.287 ± 0.035	1.697 ± 0.037	0.532 ± 0.021
	Model 2	0.08 s	0.229 ± 0.015	0.369 ± 0.020	0.976 ± 0.003	0.216 ± 0.019	0.326 ± 0.026	0.983 ± 0.003
	Model 3	0.03 s	0.138 ± 0.005	0.222 ± 0.007	0.991 ± 0.001	0.253 ± 0.005	0.374 ± 0.011	0.977 ± 0.001
	Model 4	0.03 s	0.233 ± 0.020	0.341 ± 0.028	0.979 ± 0.003	0.353 ± 0.010	0.523 ± 0.007	0.955 ± 0.001
3	Model 1	0.22 s	0.240 ± 0.054	0.369 ± 0.079	0.975 ± 0.012	1.108 ± 0.307	1.479 ± 0.392	0.622 ± 0.127
	Model 2	0.22 s	0.111 ± 0.005	0.169 ± 0.006	0.995 ± 0.0003	0.118 ± 0.012	0.169 ± 0.011	0.995 ± 0.001
	Model 3	0.08 s	0.125 ± 0.003	0.189 ± 0.003	0.994 ± 0.0002	0.227 ± 0.003	0.325 ± 0.006	0.983 ± 0.001
	Model 4	0.08 s	0.206 ± 0.002	0.297 ± 0.008	0.984 ± 0.001	0.330 ± 0.003	0.500 ± 0.004	0.959 ± 0.001
4	Model 1	0.86 s	0.855 ± 0.034	1.249 ± 0.056	0.724 ± 0.025	1.085 ± 0.013	1.606 ± 0.024	0.581 ± 0.012
	Model 2	0.86 s	0.750 ± 0.018	1.080 ± 0.026	0.794 ± 0.010	0.837 ± 0.028	1.218 ± 0.045	0.758 ± 0.018
	Model 3	0.30 s	0.222 ± 0.010	0.285 ± 0.009	0.986 ± 0.001	0.321 ± 0.011	0.432 ± 0.010	0.970 ± 0.001
	Model 4	0.30 s	0.341 ± 0.008	0.484 ± 0.007	0.959 ± 0.001	0.359 ± 0.004	0.511 ± 0.004	0.958 ± 0.001
5	Model 1	0.86 s	0.091 ± 0.008	0.135 ± 0.010	0.997 ± 0.0005	0.102 ± 0.007	0.150 ± 0.007	0.996 ± 0.0004
	Model 2	0.86 s	0.072 ± 0.002	0.109 ± 0.003	0.998 ± 0.0001	0.093 ± 0.002	0.143 ± 0.002	0.997 ± 0.0001
	Model 3	0.30 s	0.110 ± 0.004	0.173 ± 0.004	0.995 ± 0.0002	0.203 ± 0.004	0.295 ± 0.005	0.986 ± 0.0005
	Model 4	0.30 s	0.180 ± 0.003	0.267 ± 0.005	0.987 ± 0.0004	0.302 ± 0.002	0.459 ± 0.004	0.966 ± 0.001
6	Model 1	3.45 s	0.072 ± 0.006	0.105 ± 0.006	0.998 ± 0.0002	0.056 ± 0.005	0.088 ± 0.007	0.999 ± 0.0002
	Model 2	3.45 s	0.062 ± 0.004	0.095 ± 0.005	0.998 ± 0.0002	0.051 ± 0.004	0.084 ± 0.005	0.999 ± 0.0001
	Model 3	1.22 s	0.127 ± 0.006	0.182 ± 0.004	0.994 ± 0.0003	0.204 ± 0.008	0.302 ± 0.011	0.985 ± 0.001
	Model 4	1.22 s	0.178 ± 0.003	0.255 ± 0.002	0.989 ± 0.0002	0.297 ± 0.004	0.446 ± 0.004	0.968 ± 0.001

For cross-type transfer, the difficulty of adaptation increases due to larger structural difference between actuators. The pre-trained source model exhibits severe performance degradation when directly applied to PEA #3, indicating limited generalization capability without adaptation. Transfer learning therefore becomes essential in this scenario. At a dataset size of 65 samples, Model 3 again demonstrates superior performance, outperforming all other strategies by effectively balancing feature reuse and target-domain adaptation. As the data volume increases beyond this level, Model 2 generally achieves better accuracy, particularly when the source–target similarity is relatively high. At a dataset size of 125 samples, Model 2 achieves an 83% reduction in MAE, an 81% reduction in RMSE, and a 46% increase in R^2 compared with Model 1. At a dataset volume of 2000 samples, Model 2 performs better under higher similarity, whereas Model 3 remains more robust when similarity is low, revealing a nonlinear trade-off between data volume and domain similarity.

Compared with cross-batch transfer, cross-type transfer consistently results in higher MAE and RMSE values and lower R^2 values under the same data volume, reflecting the greater challenge posed by larger distribution discrepancies. Model 4, which fine-tunes only the final dense layer, performs adequately only when both data volume is small and domain similarity is high. In contrast, Model 3 provides a more robust option for small-data and low-similarity scenarios, while Model 2 offers the most stable performance across a

wide range of data volumes and similarity levels. Model 1 becomes competitive only when sufficient target-domain data are available.

4.3. Comparison with Different Transfer Learning Methods

To evaluate the effectiveness of the proposed GRU-CNN transfer learning approach, two transfer learning frameworks are selected as baseline models for comparative analysis. The long short-term memory domain adversarial neural network (LSTM-DANN) [38] is a domain-adversarial transfer learning method that has been widely adopted to address distribution shifts in time-series data. The second baseline is the deep transfer learning method in virtue of the architecture of CNN (DTr-CNN) [39], which seeks to simultaneously minimize prediction error and distribution discrepancy between the source and target domains. The total loss function of DTr-CNN consists of prediction loss, maximum mean discrepancy (MMD) loss, domain classification loss, and a regularization term. During training, the target-domain model is initialized using pre-trained source-domain parameters, after which model parameters are iteratively optimized by minimizing the total loss function with target-domain data.

For LSTM-DANN, the feature extractor consists of two stacked LSTM layers, each with 64 hidden units. The regression predictor and the domain classifier are implemented using three and two dense layers, respectively, with each dense layer containing 100 neurons. The Adam optimizer is used for training, with a learning rate of 0.005, a batch size of 100, and a total of 100 training epochs. DTr-CNN adopts the same backbone architecture as the temporal convolutional network (TCN), comprising five one-dimensional convolutional layers with a fixed kernel size of 7. The number of convolutional kernels in these layers is sequentially set to 128, 64, 32, 16, and 8. The Adam optimizer is also employed for DTr-CNN, with a learning rate of 0.0005 and a convergence threshold of 1. The performance comparison among the proposed GRU-CNN model and the two baseline transfer learning methods is summarized in Table 8.

Table 8. Experimental results under two target domain scenarios.

PEAs	Models	Train Time	Evaluation Metrics		
			MAE	RMSE	R^2
Cross-batch	LSTM-DANN	2759 s	0.181 ± 0.052	0.264 ± 0.070	0.987 ± 0.007
	DTr-CNN	995 s	0.162 ± 0.014	0.242 ± 0.016	0.990 ± 0.001
	GRU-CNN	0.03 s	0.138 ± 0.005	0.222 ± 0.007	0.991 ± 0.001
Cross-type	LSTM-DANN	3449 s	0.264 ± 0.086	0.357 ± 0.101	0.978 ± 0.013
	DTr-CNN	1034 s	0.163 ± 0.014	0.224 ± 0.012	0.992 ± 0.001
	GRU-CNN	0.08 s	0.216 ± 0.019	0.326 ± 0.026	0.983 ± 0.003

In the cross-batch transfer scenario, where PEAs share identical mechanical structures, all three models achieve excellent control accuracy, with R^2 values exceeding 0.98. Among them, the proposed GRU-CNN achieves the most favorable overall performance in both accuracy and computational efficiency. Specifically, it yields the shortest train time, which is substantially lower than that of LSTM-DANN and DTr-CNN, while also yielding the lowest MAE and RMSE together with the highest R^2. This superior performance can be attributed to GRU-CNN's strong capability to capture long-term temporal dependencies and local nonlinear patterns associated with hysteresis and low-damping vibration, which remain largely transferable across different production batches. Under such low inter-domain discrepancy conditions, directly reusing and fine-tuning the pre-trained dense representations enables efficient knowledge transfer without requiring explicit domain alignment.

By contrast, in the cross-type transfer scenario, where PEAs differ in structural configurations, the performance differentiation among the three models becomes more pronounced. In this setting, DTr-CNN attains the highest prediction accuracy, while the proposed GRU-CNN follows closely with a competitive R^2 of 0.983. LSTM-DANN again exhibits the poorest overall performance, accompanied by the largest variance in prediction results. The inferior performance and instability of LSTM-DANN stem from its requirement to simultaneously optimize target-domain regression loss and domain classification loss. With only 125 samples available in the target domain, the model faces inherent difficulties in reliably estimating distribution discrepancies between the source and target domains, which limits the effectiveness of adversarial training via the gradient reversal layer. The slightly reduced accuracy of GRU-CNN in the cross-type scenario reflects the increased adaptation difficulty associated with larger inter-domain structural differences. In particular, temporal representations learned from the source PEA may encode inductive assumptions that are no longer fully valid for the target PEA, leading to suboptimal adaptation under limited target-domain data. Under these conditions, the DTr-CNN framework demonstrates superior robustness, as it explicitly minimizes distribution discrepancies between the source and target domains through joint optimization of prediction loss and domain alignment objectives. This explicit feature-alignment mechanism facilitates more effective adaptation to larger inter-domain variations, albeit at the cost of increased computational burden due to extensive reuse of source-domain data. In comparison, the TL-based GRU-CNN maintains higher computational efficiency by fine-tuning only the dense layers.

To further visualize the effects of the proposed GRU-CNN transfer learning approach, results from the cross-batch transfer and cross-type transfer scenarios are presented in Figures 11 and 12. The solid black line represents the actual driving voltage of the PEA, and the dotted red line represents the voltage predicted by each model. DTr-CNN exhibits significant prediction bias across the entire sequence, while the predictions of GRU-CNN transfer learning model closely match the actual voltage. The error ranges of each method are summarized in Figure 11d. Specifically, the error ranges of LSTM-DANN, DTr-CNN and GRU-CNN transfer learning model are [−6.08 V, 6.15 V], [−9.55 V, 12.82 V], and [−3.82 V, 4.13 V], respectively. The error range of GRU-CNN transfer learning model is smaller than those of LSTM-DANN and DTr-CNN. These results clearly indicate that the GRU-CNN transfer learning model yields the narrowest and most stable deviation range, reflecting its highest prediction accuracy. Figure 12 corresponds to the cross-type transfer scenario, and the visualization confirms the optimal performance of the GRU-CNN transfer learning model. It maintains close alignment with the actual voltage, whereas the other methods show larger errors. LSTM-DANN has an error range of [−10.03 V, 14.35 V], DTr-CNN has an error range of [−15.47 V, 13.69 V], and GRU-CNN transfer learning model has an error range of [−8.28 V, 5.58 V]. Even in the more challenging cross-type scenario with structural discrepancies, the GRU-CNN transfer learning model achieves the smallest error range and outperforms the other transfer learning methods.

4.4. Practical Control Verification

To evaluate the practical displacement control performance of the transfer learning-based GRU-CNN feedforward controller, experiments are conducted under multiple representative operating frequencies and different target-domain data volumes. The results for the cross-batch and cross-type transfer scenarios are summarized in Tables 9 and 10, respectively. Under the cross-batch transfer scenario, the displacement trajectory amplitude is set to 0.8 mm. The GRU-CNN controller achieves stable and accurate displacement tracking across all tested frequencies, indicating that transfer learning effectively enhances feedforward control accuracy under dynamic operating conditions. As the target-domain

data volume increases from 125 to 2000 samples, both MAE and RMSE generally decrease, particularly at 50 Hz and 100 Hz operating frequencies. The control errors remain small for all data volumes, with MAE values consistently below 0.02 mm. At higher frequencies, the control task becomes more challenging due to increased dynamic effects. For the cross-type transfer scenario, the displacement trajectory amplitude is set to 0.4 mm. Increasing the target-domain data volume from 125 to 500 and 2000 samples consistently reduces MAE and RMSE and improves R^2, confirming that additional target data facilitate more effective adaptation in cross-type transfer scenarios. Overall, the results in Tables 9 and 10 demonstrate that the transfer learning-based GRU–CNN feedforward controller can achieve reliable displacement control under a range of operating frequencies and data availability conditions. These findings support the practical applicability of the proposed method for displacement control tasks where collecting large-scale experimental data for each new actuator is impractical.

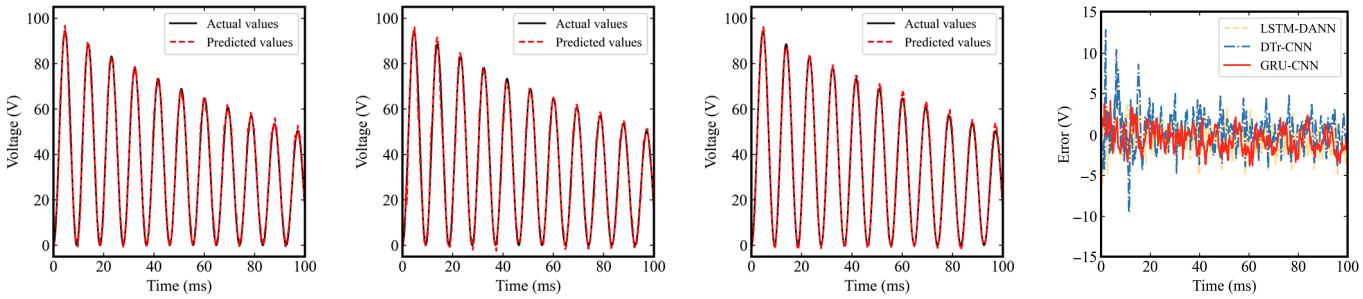


Figure 11. Performance disparities of transfer learning methods under the cross-batch scenario. (**a**) LSTM-DANN; (**b**) DTr-CNN; (**c**) GRU-CNN; (**d**) error.

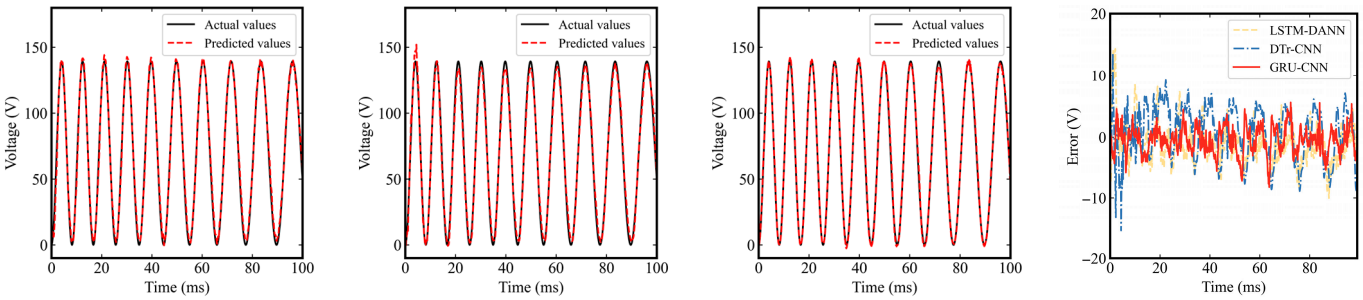


Figure 12. Performance disparities of transfer learning methods under the cross-type scenario. (**a**) LSTM-DANN; (**b**) DTr-CNN; (**c**) GRU-CNN; (**d**) error.

Table 9. Displacement control performance under cross-batch transfer.

Frequency (Hz)	MAE (mm)			RMSE (mm)			R^2		
	125	500	2000	125	500	2000	125	500	2000
10	0.014	0.017	0.017	0.018	0.024	0.023	0.996	0.993	0.993
50	0.031	0.032	0.023	0.037	0.039	0.028	0.983	0.981	0.990
100	0.043	0.029	0.026	0.054	0.037	0.033	0.964	0.983	0.987
150	0.041	0.047	0.046	0.051	0.057	0.058	0.968	0.960	0.959

Table 10. Displacement control performance under cross-type transfer.

Frequency (Hz)	MAE (mm)			RMSE (mm)			R^2		
	125	500	2000	125	500	2000	125	500	2000
20	0.009	0.006	0.010	0.010	0.007	0.012	0.995	0.997	0.993
60	0.012	0.009	0.011	0.014	0.011	0.013	0.990	0.994	0.991
110	0.020	0.014	0.018	0.025	0.017	0.021	0.968	0.985	0.977
160	0.030	0.022	0.023	0.038	0.028	0.029	0.929	0.960	0.959

5. Conclusions

In this paper, a transfer learning-based GRU-CNN feedforward control model is proposed to reduce the dependence on extensive target-domain experimental data for PEAs. By integrating gated recurrent units for temporal dependency modeling, convolutional neural networks for local nonlinear feature extraction, and transfer learning strategies for cross-domain adaptation, the proposed approach addresses the coupled effects of hysteresis and low damping vibration in PEAs under cross-batch and cross-type scenarios. Compared with training from scratch, the proposed model consistently achieves lower prediction errors and improved generalization across different actuators. In the cross-batch transfer scenario with only 125 target-domain samples, the proposed approach achieves an 85% reduction in MAE, an 82% reduction in RMSE, and a 27% increase in R^2 compared with scratch training. In the cross-type transfer scenario under the same data constraint, it leads to an 83% reduction in MAE, an 81% reduction in RMSE, and a 46% increase in R^2. From a practical perspective, the proposed model demonstrates reliable displacement control performance across a range of operating frequencies and target-domain data volumes. However, the experimental validation involves only three PEAs, with a limited range of structural differences between the source and target domains. Future work will focus on extending the evaluation to a larger and more diverse actuator population to strengthen the generality of the conclusions. In addition, the present study focuses on feedforward control modeling and offline evaluation; real-time closed-loop implementation and end-to-end control delay analysis will be addressed in future research.

Author Contributions: Conceptualization, Y.H. and H.J.; methodology, Y.H. and Y.Y.; software, Y.H.; validation, Y.H.; resources, X.C.; data curation, Y.H. and Y.Y.; writing—original draft preparation, Y.H. and Y.Y.; writing—review and editing, H.J. and X.C.; visualization, Y.H.; funding acquisition, X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (Grant No.52475596), and the State Key Laboratory of New Ceramic Materials Tsinghua University (No. KF202420).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The author would like to thank Paihe Science and Technology Holding Co., Ltd. Beijing for the technical support and the provision of the PEAs.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Baziyad, A.G.; Ahmad, I.; Salamah, Y.B.; Alkuhayli, A. Robust Tracking Control of Piezo-Actuated Nanopositioning Stage Using Improved Inverse LSSVM Hysteresis Model and RST Controller. *Actuators* **2022**, *11*, 324. [CrossRef]

2. Yi, K.A.; Veillette, R.J. A Charge Controller for Linear Operation of a Piezoelectric Stack Actuator. *IEEE Trans. Control Syst. Technol.* **2005**, *13*, 517–526. [CrossRef]
3. Jiles, D.C.; Atherton, D.L. Theory of ferromagnetic hysteresis. *J. Magn. Magn. Mater.* **1986**, *61*, 48–60. [CrossRef]
4. Maxwell, J.C. A Dynamical Theory of the Electromagnetic Field. *Philos. Trans. R. Soc. London* **1865**, *155*, 459–512. [CrossRef]
5. Dahl, P.R. Solid Friction Damping of Mechanical Vibrations. *AIAA J.* **1976**, *14*, 1–8. [CrossRef]
6. Preisach, F. On the Magnetic Aftereffect. *IEEE Trans. Magn.* **2017**, *53*, 0700111. [CrossRef]
7. Prandtl, L. Ein Gedankenmodell zur kinetischen Theorie der festen Körper. *ZAMM Z. Für Angew. Math. Und Mech.* **1928**, *8*, 85–106. [CrossRef]
8. Wen, Y.-K. Method for Random Vibration of Hysteretic Systems. *J. Eng. Mech. Div.* **1976**, *102*, 249–263. [CrossRef]
9. Dong, R.; Tan, Y.; Chen, H.; Xie, Y. A Neural Networks Based Model for Rate-Dependent Hysteresis for Piezoceramic Actuators. *Sens. Actuators A Phys.* **2008**, *143*, 370–376. [CrossRef]
10. Serpico, C.; Visone, C. Magnetic Hysteresis Modeling via Feed-Forward Neural Networks. *IEEE Trans. Magn.* **1998**, *34*, 623–628. [CrossRef]
11. Adly, A.A.; Abd-El-Hafiz, S.K. Using Neural Networks in the Identification of Preisach-Type Hysteresis Models. *IEEE Trans. Magn.* **1998**, *34*, 629–635. [CrossRef]
12. Yan, J.; DiMeo, P.; Sun, L.; Du, X. LSTM-Based Model Predictive Control of Piezoelectric Motion Stages for High-Speed Autofocus. *IEEE Trans. Ind. Electron.* **2023**, *70*, 6209–6218. [CrossRef]
13. Wang, Z.; Zhang, Z.; Mao, J.; Zhou, K. A Hammerstein-Based Model for Rate-Dependent Hysteresis in Piezoelectric Actuator. In Proceedings of the 2012 24th Chinese Control and Decision Conference (CCDC), Taiyuan, China, 23–25 May 2012; pp. 1391–1396. [CrossRef]
14. Cheng, L.; Liu, W.; Hou, Z.-G.; Yu, J.; Tan, M. Neural-Network-Based Nonlinear Model Predictive Control for Piezoelectric Actuators. *IEEE Trans. Ind. Electron.* **2015**, *62*, 7717–7727. [CrossRef]
15. Shao, S.; McAleer, S.; Yan, R.; Baldi, P. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Trans. Ind. Inf.* **2019**, *15*, 2446–2455. [CrossRef]
16. Liu, S.; Stebner, A.P.; Kappes, B.B.; Zhang, X. Machine Learning for Knowledge Transfer across Multiple Metals Additive Manufacturing Printers. *Addit. Manuf.* **2021**, *39*, 101877. [CrossRef]
17. Li, Z.; Wang, L.; Liu, R.; Mirzadarani, R.; Luo, T.; Lyu, D.; Niasar, M.G.; Qin, Z. A Data-Driven Model for Power Loss Estimation of Magnetic Materials Based on Multi-Objective Optimization and Transfer Learning. *IEEE Open J. Power Electron.* **2024**, *5*, 605–617. [CrossRef]
18. Ye, X.; Ni, Y.-Q.; Ao, W.K.; Yuan, L. Modeling of the Hysteretic Behavior of Nonlinear Particle Damping by Fourier Neural Network with Transfer Learning. *Mech. Syst. Signal Process.* **2024**, *208*, 111006. [CrossRef]
19. Gu, Y.; Lu, X.; Xu, Y. A Deep Ensemble Learning-Driven Method for the Intelligent Construction of Structural Hysteresis Models. *Comput. Struct.* **2023**, *286*, 107106. [CrossRef]
20. Kebede, G.A.; Lo, S.-C.; Wang, F.-K.; Chou, J.-H. Transfer Learning-Based Deep Learning Models for Proton Exchange Membrane Fuel Remaining Useful Life Prediction. *Fuel* **2024**, *367*, 131461. [CrossRef]
21. Sabarianand, D.V.; Karthikeyan, P.; Muthuramalingam, T. A Review on Control Strategies for Compensation of Hysteresis and Creep on Piezoelectric Actuators Based Micro Systems. *Mech. Syst. Signal Process.* **2020**, *140*, 106634. [CrossRef]
22. Zhong, J.; Nishida, R.; Shinshi, T. A Digital Charge Control Strategy for Reducing the Hysteresis in Piezoelectric Actuators: Analysis, Design, and Implementation. *Precis. Eng.* **2021**, *67*, 370–382. [CrossRef]
23. Jiang, Z.; Li, Y.; Wang, Q. Modeling of the Dynamic Hysteresis in DEAP Actuator Using an Empirical Mode Decomposition Based Long-Short Term Memory Network. *J. Intell. Mater. Syst. Struct.* **2021**, *32*, 2108–2123. [CrossRef]
24. Patil, M.S.; Charuku, B.; Ren, J. Long Short-Term Memory Neural Network-Based System Identification and Augmented Predictive Control of Piezoelectric Actuators for Precise Trajectory Tracking. *IFAC-PapersOnLine* **2021**, *54*, 38–45. [CrossRef]
25. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling 2014. In Proceedings of the NIPS 2014 Deep Learning and Representation Learning Workshop, Montreal, QC, Canada, 13–14 December 2014; pp. 1–9. [CrossRef]
26. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D Convolutional Neural Networks and Applications: A Survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [CrossRef]
27. Sudjianto, A.; Knauth, W.; Singh, R.; Yang, Z.; Zhang, A. Unwrapping The Black Box of Deep ReLU Networks: Interpretability, Diagnostics, and Simplification. *arXiv* **2020**, arXiv:2011.04041. [CrossRef]
28. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
29. Zhao, M.; Zhong, S.; Fu, X.; Tang, B.; Pecht, M. Deep Residual Shrinkage Networks for Fault Diagnosis. *IEEE Trans. Ind. Inf.* **2020**, *16*, 4681–4690. [CrossRef]

30. Li, J.; Lu, J.; Chen, C.; Ma, J.; Liao, X. Tool Wear State Prediction Based on Feature-Based Transfer Learning. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, 3283–3301. [CrossRef]
31. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15. [CrossRef]
32. Shiblee, M.; Kalra, P.K.; Chandra, B. Time Series Prediction with Multilayer Perceptron (MLP): A New Generalized Error Based Approach. In *Advances in Neuro-Information Processing*; Köppen, M., Kasabov, N., Coghill, G., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5507, pp. 37–44.
33. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the NIPS 2014 Advances in Neural Information Processing Systems 27, Montreal, QC, Canada, 8–13 December 2014; pp. 1–9. [CrossRef]
34. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271. [CrossRef]
35. Yin, R.; Ren, J. Sequence-to-Sequence LSTM-Based Dynamic System Identification of Piezo-Electric Actuators. In *2023 American Control Conference (ACC)*; IEEE: San Diego, CA, USA, 2023; pp. 673–678. [CrossRef]
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1–11. [CrossRef]
37. Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; Vaughan, J.W. A Theory of Learning from Different Domains. *Mach. Learn.* **2010**, *79*, 151–175. [CrossRef]
38. Fang, X.; Gong, G.; Li, G.; Chun, L.; Peng, P.; Li, W. A General Multi-Source Ensemble Transfer Learning Framework Integrate of LSTM-DANN and Similarity Metric for Building Energy Prediction. *Energy Build.* **2021**, *252*, 111435. [CrossRef]
39. Ye, R.; Dai, Q. Implementing Transfer Learning across Different Datasets for Time Series Forecasting. *Pattern Recognit.* **2021**, *109*, 107617. [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.