Article | Received 12 May 2026; Revised 20 July 2026; Accepted 1 August 2026; Published 31 August 2026
https://doi.org/10.55092/sc20260015

MixStyle-based adversarial domain generalization for cross-domain bridge damage localization

Xiaoxia Wei[1], Yizhou Lin[2,3,*], Hongwei Ma[2,3], Zhenhua Nie[1,4,5,*] and Jun Li[6]

[1] School of Mechanics and Construction Engineering, Jinan University, Guangzhou, China
[2] School of Environment and Civil Engineering, Dongguan University of Technology, Dongguan, China
[3] Guangdong Provincial Key Laboratory of Intelligent Disaster Prevention and Emergency Technologies for Urban Lifeline Engineering, Dongguan, China
[4] Key Laboratory of Marine Civil Engineering Materials and Structures, Ministry of Education of China, Guangzhou, China
[5] The Key Laboratory of Green Toughening and Safety Prevention and Control of Offshore Structures, Department of Education of Guangdong Province, Guangdong, China
[6] School of Civil and Mechanical Engineering, Curtin University, Perth, Australia

* Correspondence authors; E-mails: lin_yizhou@dgut.edu.cn (Y.L.); niezh@jnu.edu.cn (Z.N.).

Highlights:

- Establish a style-transfer mechanism to enhance source domain diversity.
- Implement adversarial training to improve cross-domain robustness.
- Predict structural damage accurately without using sample information from real structures.
- Demonstrate robustness in real-world environments and apply successfully to different damage scenarios.

Abstract: A critical challenge in structural health monitoring (SHM) is the extreme scarcity of labeled damage data from real-world bridges. Consequently, domain adaptation approaches that require target domain data with damage situation become nearly infeasible for real-world engineering applications. To address this limitation, this paper proposes a structural damage detection (SDD) method based on domain generalization, which constructs a highly generalizable damage recognition model by integrating style transfer and adversarial training. A latent domain with diverse styles is generated by mixing style features, such as mean and variance from different samples, followed by label generation through clustering. Adversarial training is then introduced to encourage the feature extractor to learn domain-invariant damage-sensitive features, thereby enabling the model to focus on essential features that are indicative of damage states. To validate the effectiveness of the proposed method, a simply supported beam model was established as the benchmark structure. Simulation results demonstrate that, compared to conventional damage identification algorithms, the proposed method exhibits superior damage localization performance on unseen target domain data. Furthermore, laboratory experiments confirm that the approach achieves excellent results in practical scenarios.

Keywords: style transfer; domain generalization; bridge damage localization; adversarial training; structural health monitoring

1.Introduction

As critical components of modern transportation infrastructure, bridges play a pivotal role in ensuring public safety, sustaining economic development, and maintaining social stability. Throughout their service life, these structures inevitably undergo progressive deterioration caused by multiple factors including material aging, environmental corrosion, increasing traffic loads, and extreme events such as earthquakes and floods. Such degradation may ultimately lead to sudden structural failures with potentially catastrophic consequences. SHM has emerged as an essential technological solution for early damage detection, identifying critical defects including cracks, corrosion, and bearing failures before they develop into catastrophic failures. As systematically outlined by Farrar [1], the fundamental objectives of SHM address four key questions: (1) Determining whether damage has occurred; (2) Locating the spatial position of damage; (3) Quantifying the severity of damage; (4) Predicting the remaining service life of the structure.

Structural Damage Detection (SDD) technology serves as the technical foundation for systems, providing essential methodologies for addressing critical damage assessment challenges. Current SDD approaches are broadly classified into local and global methods, with the latter category further subdivided into model-based and model-free techniques. The model-based approach employs a parametric modeling framework, beginning with the development of an accurate finite element model based on structural parameters. Subsequent comparison between empirical measurement data and simulated healthy-state model outputs enables quantitative damage identification [2]. The efficacy of this methodology fundamentally relies on establishing an intact-state model that faithfully represents the real-world structure. This process facilitates precise damage localization and severity assessment through detection of anomalous variations in localized stiffness parameters [3]. Recent research has advanced model-based approaches through several innovative methodologies. Significant contributions include: Simoen *et al.* [4] updated the vibration-based finite element model by using the non-probabilistic fuzzy method and the probabilistic Bayesian method for damage assessment. Xiong *et al.* [5] used the Maximum Likelihood Estimation (MLE) method to estimate the distribution characteristics of randomly varying parameters, and adopted the U-pool method [6] as the verification index to evaluate the accuracy of the updated model in the region of interest. Levine *et al.* [7] updated the finite element model using neural networks. Ren *et al.* [8] proposed a structural model update method based on the response surface Goller *et al.* [9] proposed the application of the Bayesian model update method based on modal attributes in dynamic systems. However, field implementation of these model-based approaches in bridge health monitoring presents substantial challenges. In-service bridges exhibit inherent structural complexity and are subject to environmental noise and modeling uncertainties. These factors collectively impact the effectiveness of model-updating techniques for damage detection. These operational constraints underscore the need for developing more robust SDD solutions that accommodate real-world monitoring conditions while maintaining diagnostic accuracy.

The model-free approach in structural damage detection primarily utilizes data-driven methodologies that correlate measurement data with damage characteristics through structural response analysis. Significant advancements in this area include Nie *et al.* [10] development of a vibration-based damage

identification method using phase space topology analysis. Nie *et al.* [11] further advanced the field by applying Functional Principal Component Analysis (FPCA) to operational vibration data, demonstrating its effectiveness through a real-world case study of a suspension bridge collision. Li *et al.* [12] contributed an innovative method correlating strain response covariance with modal parameters, validated through experimental studies on a seven-story steel frame structure. While effective with simulated data, these techniques face substantial implementation challenges when processing actual field measurements, including insufficient damage sensitivity, vulnerability to environmental noise, and significant performance gaps between controlled laboratory conditions and real-world structural monitoring scenarios. These practical limitations emphasize the ongoing need for methodological improvements to facilitate the transition from research to field-deployable structural health monitoring (SHM) solutions.

Deep learning has revolutionized data-driven approaches in structural health monitoring by enabling automatic feature extraction from vast amounts of monitoring data while effectively filtering out environmental influences that typically challenge traditional physical models. In the field of SHM, convolutional neural network (CNN) have emerged as the predominant deep learning tool for feature extraction, demonstrating particular effectiveness in vision-based damage detection applications [13,14]. Lin *et al.* [15] developed an advanced damage recognition technique employing deep CNN that directly processes raw sensor data to automatically identify features and precisely locate damage. This fundamental constraint currently impedes real-world implementation of supervised deep learning methods, despite their demonstrated theoretical advantages, thus driving the need for innovative approaches that can effectively utilize limited or unlabeled operational monitoring data.

Transfer learning methods offer a promising solution for structural damage detection by transferring knowledge from source domains (e.g., numerical simulation data) to target domains (actual monitoring data), enabling cross-domain damage identification without requiring annotated target domain data. Lin and Nie *et al.* [16] developed a cross-domain damage detection approach using domain adaptive transfer learning, where neural networks are trained simultaneously on both numerical simulation and real monitoring data without needing labeled damage data from actual structures. Building on this, Nie *et al.* [17] implemented an adversarial transfer learning framework that successfully bridges numerical simulations and real bridge structures, demonstrating superior damage localization accuracy compared to conventional methods by extracting domain-invariant features that reduce modeling errors and uncertainties. Further advancing the field, Valentina [18] introduced a bridge damage identification technique based on domain adaptation, which constructs a shared cross-domain feature space by thoroughly analyzing marked source domain data. Similarly, Li *et al.* [19] proposed the Partial Conditional Adversarial Domain Adaptation Network (PCADA), which employs adversarial training between feature generators and domain discriminators to obtain damage-sensitive yet domain-invariant representations, significantly enhancing cross-domain detection performance. While domain adaptation methods [20] have demonstrated effectiveness in structural damage identification, their reliance on target domain data for model fine-tuning presents significant practical limitations—particularly in scenarios involving new bridge constructions or post-disaster assessments where such data is typically unavailable.

The core of DG is to learn a model with strong generalization ability from multiple source domains, and the goal is to maintain stable performance on unknown target domains [21]. Fan *et al.* [22] made significant contributions through their Deep Mixed Domain Generalization Network (DMDGN), designed for cross-domain fault diagnosis under unseen operating conditions. Building on this foundation,

Wang *et al.* [23] developed an innovative three-stage knowledge distillation framework for domain generalization in structural damage detection. Further advancing the field, Ragab *et al.* [24] implemented a Conditional Contrastive Domain Generalization (CCDG) approach for steel rolling machinery fault diagnosis. Nie *et al.* [25] proposed a style transfer domain generalization deep learning model that integrates Phase-Consistent Normalization (PCN) and Batch-Instance Normalization (BIN) to address the generalization problem in damage detection caused by distribution shifts between finite element models and real structures. However, The PCN mixing process is one-off and occurs only at the input stage of the network, which limits the diversity and richness of the augmented data. However, Moreover, by merely filtering out noise to retain damage-related information in CNNs, this approach struggles to learn domain-invariant features from more complex data, ultimately restricting its generalization capability. These domain generalization approaches collectively represent important progress in overcoming the data availability limitations that constrain traditional domain adaptation methods, offering more practical solutions for real-world structural health monitoring applications where target domain data may be scarce or completely unavailable [26].

The individual components used in the proposed framework, including MixStyle, K-means clustering, gradient reversal, and adversarial training, have been established in previous studies. The contribution of this work lies in their combination for source-only bridge damage localization. MixStyle is applied to intermediate structural-response features, K-means provides auxiliary pseudo-domain labels, and the domain discriminator is trained jointly with the damage-localization classifier. Compared with the PCN-BIN strategy in Nie *et al.*, the proposed framework introduces feature-statistics mixing during feature extraction together with an explicit domain-discrimination objective. The relative performance of these approaches depends on the data and evaluation protocol.

We propose an innovative framework that integrates MixStyle data augmentation with adversarial training to achieve enhanced cross-domain generalization capabilities. The MixStyle augmentation technique plays a pivotal role in expanding the effective domain coverage space, facilitating more accurate simulation of potential target domain distributions while preserving essential source domain characteristics. Through carefully designed adversarial training, our approach enforces the learning of domain-invariant feature representations that demonstrate consistent robustness under diverse operational scenarios. A distinctive advantage of this framework is its ability to achieve reliable cross-domain damage detection performance while requiring only numerical simulation data (source domain) during training, completely eliminating the dependency on actual structural monitoring data (target domain).

The organization of this paper is structured as follows. Section 2 introduces the fundamental concepts underlying the proposed methodology and presents its detailed implementation framework. Section 3 validates the method's effectiveness through comprehensive numerical simulations. Experimental verification using a real simply-supported beam structure is conducted in Section 4 to demonstrate the practical applicability of the approach. Finally, Section 5 summarizes the key findings and conclusions of this study.

2. Methodology

The proposed domain-generalization approach involves three key steps (Figure 1). First, MixStyle-based feature augmentation is applied to source-domain response data to enhance distribution coverage. Second, K-means clustering generates auxiliary pseudo-domain labels, and adversarial training learns

features that are sensitive to physical damage labels but less sensitive to latent-domain statistics. Finally, a damage-location classifier is trained using the source-domain labels. During testing, the trained feature extractor and damage classifier directly process unseen target-domain response data; the domain discriminator is not used. In this study, the primary task is cross-domain damage localization. The classifier outputs the intact state or a specific damage-location class. The five damage-severity levels are used to increase data diversity and represent different degrees of localized stiffness reduction; they are not independent output classes. K-means labels are auxiliary pseudo-domain labels and are not damage labels.

The key innovations are as follows: (1) Enhanced data augmentation through MixStyle effectively simulates potential target domain distributions, overcoming data acquisition challenges in bridge monitoring scenarios; (2) Novel unsupervised adaptation combines clustering-generated pseudo-labels with adversarial training for robust cross-domain feature alignment; (3) Practical engineering solution requiring only numerical simulation data (source domain) while maintaining accuracy on real structures (target domain) with wider parameter variations, eliminating dependency on target domain data.

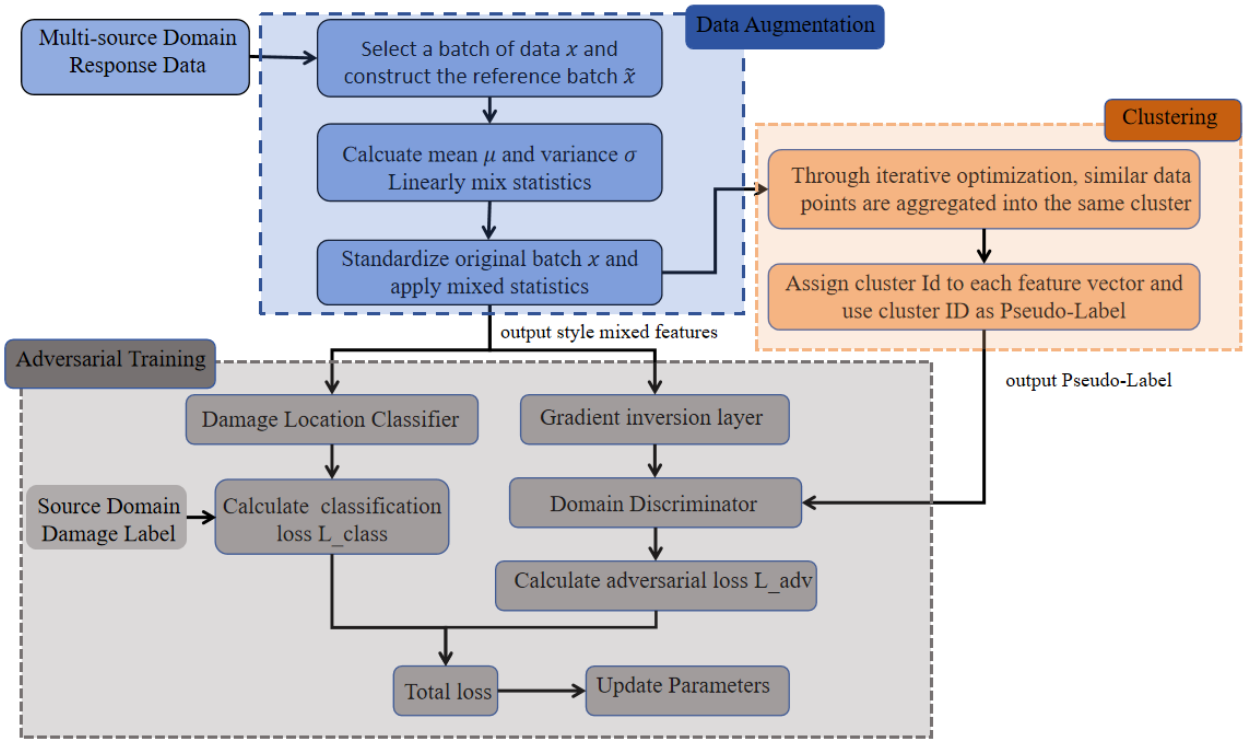


Figure 1. The three key steps of the proposed domain generalization method.

2.1. Style transfer

Style transfer [27] refers to a technique that transfers the texture characteristics of a style image to a content image while preserving the structural integrity of the original content. The batch normalization (BN) technique, introduced by Ioffe *et al.* [28], has demonstrated substantial improvements in the training efficiency of feed-forward neural networks. During neural network training, continuous weight updates cause shifts in the input distribution of subsequent layers, often resulting in gradient saturation and consequently slower model convergence. BN addresses this issue by normalizing features to follow a distribution with zero mean and unit variance [29], thereby stabilizing the learning process and accelerating training.

Let $X \in R^{N*C*H*W}$ be a batch of tensors, N, C, H, and W represent the dimensions of batch, channel, height, and width, respectively, and BN is defined as

$$BN(x) = \gamma(\frac{x-\mu(x)}{\sigma(x)}) + \beta, \tag{1}$$

Where, γ、$\beta \in R^C$ are the two parameters of the affine transformation, and $\mu(x)$、$\sigma(x) \in R^C$ respectively the mean and the standard deviation.

$$\mu(x) = \frac{1}{NHW}\sum_{n=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{nchw}, \tag{2}$$

$$\sigma(x) = \sqrt{\frac{1}{NHW}\sum_{n=1}^{N}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{nchw} - \mu(x))^2 + \varepsilon} \tag{3}$$

Ulyanov [30] found that replacing BN with Instance Normalization (IN) improved the performance of style migration. The difference between IN and BN is that IN is standardized for each sample, and the formula is as follows:

$$IN(x) = \gamma(\frac{x-\mu(x)}{\sigma(x)}) + \beta, \tag{4}$$

Where, γ、$\beta \in R^C$ are the two parameters of the affine transformation, and $\mu(x)$、$\sigma(x) \in R^C$ respectively the mean and the standard deviation for each sample.

$$\mu_{nc}(x) = \frac{1}{NW}\sum_{n=1}^{N}\sum_{w=1}^{W} x_{nchw}, \tag{5}$$

$$\sigma_{nc}(x) = \sqrt{\frac{1}{HW}\sum_{n=1}^{N}\sum_{w=1}^{W}(x_{nchw} - \mu_{nc}(x))^2 + \varepsilon}, \tag{6}$$

Huang *et al.* [31] introduced adaptive instance normalization (AdaIN). AdaIN can achieve any style transfer without learning parameters,γ、β.For example, if there are two graphs, namely feature x of the content graph and feature y of the style graph. AdaIN can transfer the style of y to x:

$$AdaIN(x,y) = \sigma(y)\frac{x-\mu(x)}{\sigma(x)} + \mu(y), \tag{7}$$

Where $\mu(y)$ represents the mean of the style graph and $\sigma(y)$ represents the variance of the style graph.

2.2. MixStyle

Zhou *et al.* [32] were inspired in AdaIN and improved the style transfer method. MixStyle generates new domains by mixing the types of training instances. Therefore, it increases the domain diversity of the source domain and improves the generalization of the training model. MixStyle can be implemented in small batch training. A given input batch x, MixStyle generates the reference batch $\tilde{x}$.If the domain label is known, x samples from two different domains, i and j, and then switches positions.

For structural response data, "style" refers to domain-dependent response statistics rather than visual texture. These statistics may include response amplitude, channel-wise mean and variance, excitation conditions, structural-parameter variation, boundary conditions, sensor characteristics, and environmental or measurement effects. A value of $\alpha = 0.1$ was used to apply relatively conservative feature-statistics perturbations, thereby limiting changes to features that may be relevant to damage localization.

$$\tilde{x}= [\text{Shuffle } (x_i), \text{Shuffle } (x_j)], \tag{8}$$

If the domain label is unknown, x is randomly sampled from the training data,

$$\tilde{x}= \text{Shuffle}(x), \tag{9}$$

Calculate the mean and variance of the x and $\tilde{x}$.Sample the mixed weights from the beta distribution, $\lambda \sim \text{Beta}(\alpha, \alpha)$. The mixed feature statistic is defined as:

$$\gamma_{mix} = \lambda\sigma(x) + (1-\lambda)\sigma(\tilde{x}), \tag{10}$$

$$\beta_{mix} = \lambda\mu(x) + (1-\lambda)\mu(\tilde{x}), \tag{11}$$

Finally, the mixed feature statistics are applied to the input batches after style standardization.

$$\text{Mixstyle}(x) = \gamma_{mix} \frac{x - \mu(x)}{\sigma(x)} + \beta_{mix} \tag{12}$$

2.3. K-means clustering

K-means clustering [33] is used to divide a known data set into K non-overlapping clusters. K-means clustering is optimized by continuously iterating so that the data points within the same cluster are similar and the data points in different clusters are different. Each cluster has a central point. The purpose of clustering is to minimize the sum of squared errors from each point to the center point [34].

$$J(c_k) = \sum_{x_i \epsilon c_i} \| x_i - \mu_k \|^2, \tag{13}$$

$$J(C) = \sum_{k=1}^{K} J(c_k) = \sum_{k=1}^{K} \sum_{x_i \epsilon c_i} \| x_i - \mu_k \|^2 = \sum_{k=1}^{K} \sum_{i=1}^{n} d_{ki} \| x_i - \mu_k \|^2, \tag{14}$$

Among them, $\| x_i - \mu_k \|$ represents the Euclidean distance from the data point to the cluster center.

$$d_{ki} = \begin{cases} 1, x_i \in c_i \\ 0, x_i \notin c_i \end{cases}, \tag{15}$$

In machine learning and data analysis, the role of clustering data is to model the data structure to improve model performance. Clustering can separate semantic features (such as object categories) from interfering factors (such as lighting and style). In this paper, pseudo-labels are generated after clustering the data generated by MixStyle. This can not only explicitly separate semantic content and domain-related style, but also benefit the subsequent model to extract cross-domain invariant features.

The K-means pseudo-labels represent groups of MixStyle-augmented features with similar distributional characteristics. They do not represent damage locations, damage severity, or damage categories. Physical damage labels supervise the damage classifier, whereas pseudo-domain labels provide auxiliary supervision to the domain discriminator.

2.4. Domain adversarial training based on MixStyle

In order to have good generalization ability in never-before-seen domains, the model needs to overcome distribution changes among multiple source domains and learn more general feature representations. We introduce the MixStyle method in adversarial training to enhance feature diversity

by mixing multi-source domain feature statistics (Figure 2). Without changing the structure of the classifier and discriminator, the cross-domain generalization ability of the model has been effectively improved [35]. Some studies [36] have shown that the underlying feature statistics (such as mean, variance) of CNN encode rich style information. By changing these feature statistics, researchers find that cross-domain style transfer can be achieved without changing the semantic content of images. Based on this discovery, the MixStyle method constructs new samples with mixed style features by linearly interpolating the feature statistics of samples from different source domains. Because the latent domains generated by the MixStyle method have no labels, we cluster the latent domains in each iteration and set pseudo-labels for different categories. The source domain datasets outputs high-level semantic features after passing through the feature extractor. These features are input into the domain discriminator to train the domain discriminator to recognize the domain of the features. The adversarial losses [37,38] are as follows:

The adversarial component consists of a feature extractor, a domain discriminator, and a gradient reversal layer. It does not include a conventional GAN generator that synthesizes samples. The method is therefore described as a MixStyle-based adversarial domain-generalization network.

$$L_{adv} = E_x[logD(y_{domain}|x) + \lambda \cdot KL(D(x_{same\ cluster})||Uniform)], \tag{16}$$

"Uniform" represents uniform distribution.

Then, adversarial training should be conducted on the discriminator and the feature extractor should be updated. Insert a gradient inversion layer between the feature extractor and the discriminator [33]:

$$R_\lambda(x) = x, \tag{17}$$

During back propagation, the gradient inversion layer will automatically reverse the gradient.

$$\frac{dR_\lambda}{dx} = -\lambda I, \tag{18}$$

I is the identity matrix and is the coefficient of the diagonal matrix.

The damage classifier compares the classification labels of the source domain with the predicted labels obtained by the feature extractor, and trains the classifier through the classification loss [36] function to make it perform better in the latent domain. The classified losses are as follows:

$$L_{class} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{k} y_{i,k}\log(p_{i,k}), \tag{19}$$

$y_{i,k}$ is the true probability of the sample category, and $p_{i,k}$ is the predicted probability of the neural network.

The overall objective function is as follows:

$$\mathrm{L} = \min L_{class}(F_f, F_c) - \lambda L_{adv}(F_f, F_d), \tag{20}$$

F_f is the output of the feature extractor, F_c is the output of the damage classifier, and F_d is the output of the discriminator.

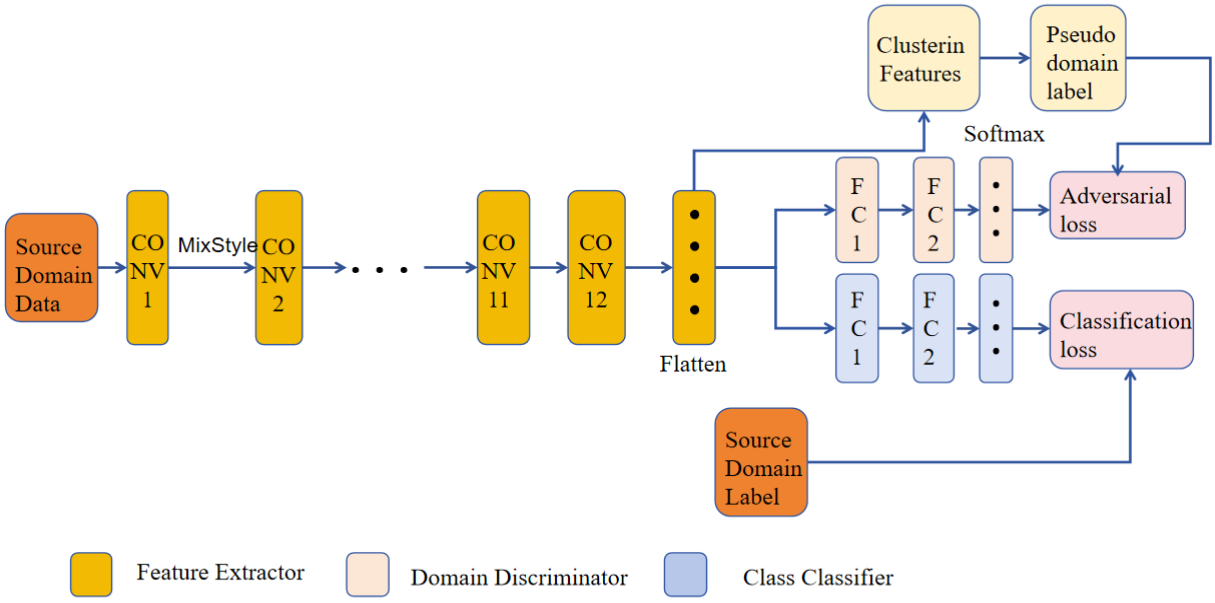


Figure 2. Domain generalization neural network architecture.

2.5. Adversarial and MixStyle based SDD method

Figure 3 outlines the overall methodology. During training, the model uses preprocessed finite element response data as input. MixStyle is applied in the early layers of the feature extractor to blend instance-level feature statistics (mean and standard deviation) across source domains, altering style attributes while preserving structural information. This increases feature diversity, enabling more robust and generalizable representations. The style-augmented features are then processed by the feature extractor, which projects the multi-source data into a unified feature space. This yields discriminative features that perform consistently under varying structural and environmental conditions.

To overcome the challenge of unlabeled latent domains created by MixStyle, we implement a K-means clustering approach that generates pseudo-labels by analyzing semantic feature relationships within the learned representations. This clustering methodology is further enhanced by applying a uniform distribution regularization term to the discriminator's output predictions through our specifically designed loss function. The architecture strategically positions a gradient reversal layer between the feature extractor and domain discriminator, effectively confounding the discriminator's ability to determine feature provenance and thereby strengthening domain-invariant feature learning. During the adversarial optimization process, we maintain a dynamic pseudo-labeling system where feature representations are periodically reclustered and relabeled, ensuring continuous alignment between the evolving feature space and label assignments while preventing the degradation of pseudo-label relevance during model training.

Finally, the extracted features enter the damage classifier, which will attempt to predict the damage category of the input data. These predictions are compared with the real damage labels to form classified loss terms, which will guide the network to predict the damage better.

Our goal is to expand the latent domain through MixStyle in the case of data without real Bridges, and obtain a feature extractor that is not affected by the domain and is sensitive to damage through adversarial training, so that it has good generalization ability on unseen datasets. During the testing phase, we fix all the parameters, remove the discriminator, and then input the dynamic response data of

the real bridge into the neural network. The predicted labels obtained by the damage classifier are compared with the real labels to obtain the accuracy rate.

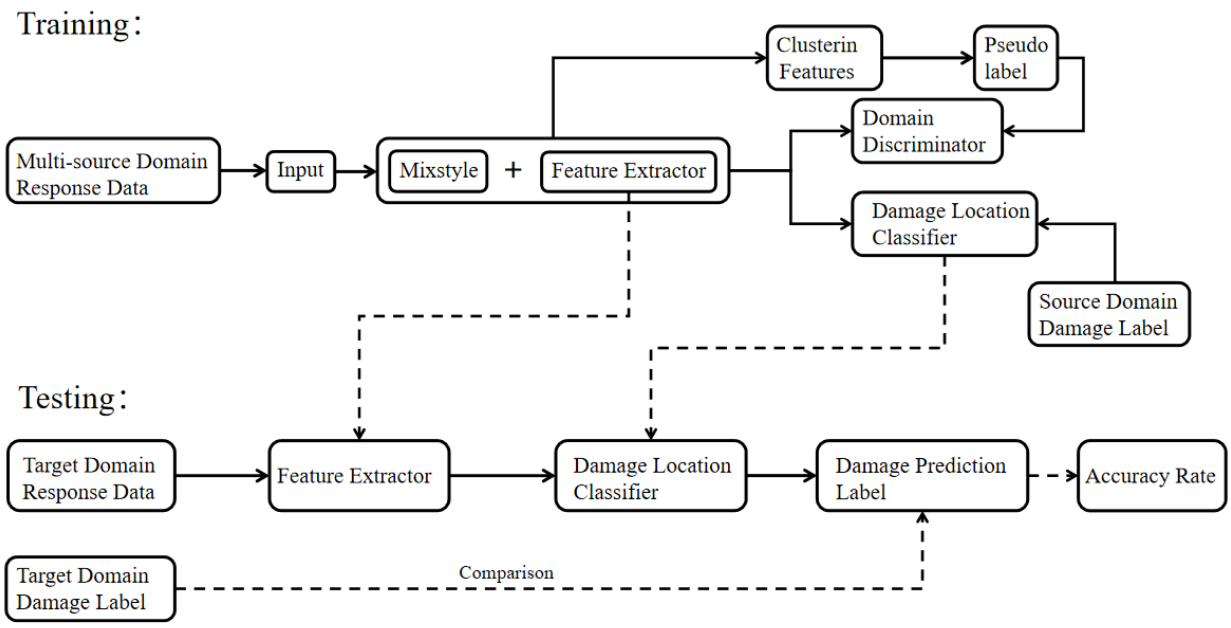


Figure 3. Overview of damage detection procedure.

3. Numerical simulations

3.1. Numerical model

In practical bridge engineering, structural components often exhibit slight dimensional variations from their design specifications, while material properties like elastic modulus and density of concrete and steel typically undergo gradual changes throughout the service life. To better capture these real-world conditions in our numerical modeling approach, we introduce controlled parameter variations for key structural properties including length, stiffness, and elastic modulus. This parametric modeling strategy offers three significant advantages: it explicitly accounts for inherent discrepancies between design values and actual bridge conditions; indirectly simulates long-term material deterioration through parameter fluctuations; and tests model robustness by evaluating sensitivity to input variations while simultaneously enhancing generalization capability through exposure to diverse data scenarios. Specifically, our numerical model randomly samples each key parameter within predefined ranges, ensuring that every generated source domain dataset exhibits natural variations representative of potential real-world bridge states.

As shown in Table 1, the simply supported beam is 1350 mm long and has a cross-section of 79 × 7.6 mm. It is divided uniformly into five Euler—Bernoulli beam elements. The Young's modulus is 206 GPa and the density is 7900 kg/m3. Rayleigh damping is used in the simulation, and the governing equation is:

$$C = \alpha K + \beta M \tag{21}$$

α and β are $4.5*10^{-6}$ and $5.5 \times 10^{-1}s^{-1}$ respectively. K is the stiffness matrix and M is the mass matrix.

Table 1. The specific properties of the numerical model.

Materials	Steel
Length	1350 mm
Cross-section	79*7.6 mm
Unit number	5
Young Modulus	206 GPA
Density	7900 kg/m^3
Damping	Rayleigh Damping

The diversity within our datasets stems from two primary sources: variations in damage simulation and excitation uncertainty. Our model incorporates five progressive damage levels in addition to the intact state, simulated through localized stiffness reductions of 10%, 20%, 30%, 40%, and 50% respectively. To generate comprehensive loading scenarios, we applied random excitations at five nodal positions (numbered 1 through 5), with each location subjected to 32 distinct load applications. This systematic approach yielded a total of 4800 unique data samples, calculated as 32 load applications × 5 excitation locations × 5 damage positions × (5 damage levels + 1 undamaged state).

The time-domain response data first undergoes low-pass filtering with a 250 Hz cutoff frequency, selected to capture the first five structural modes while excluding higher frequencies (the fifth natural frequency being 232 Hz). This filtered signal is then down sampled to 2000 Hz to remove redundant spectral content while preserving critical structural dynamic characteristics.

As illustrated in Figure 4, our network architecture employs a 12-layer convolutional neural network for feature extraction. Both the damage classifier and domain discriminator share this identical base architecture, with modifications limited to their respective output layers to accommodate their distinct functional roles in either damage state prediction or domain identification.

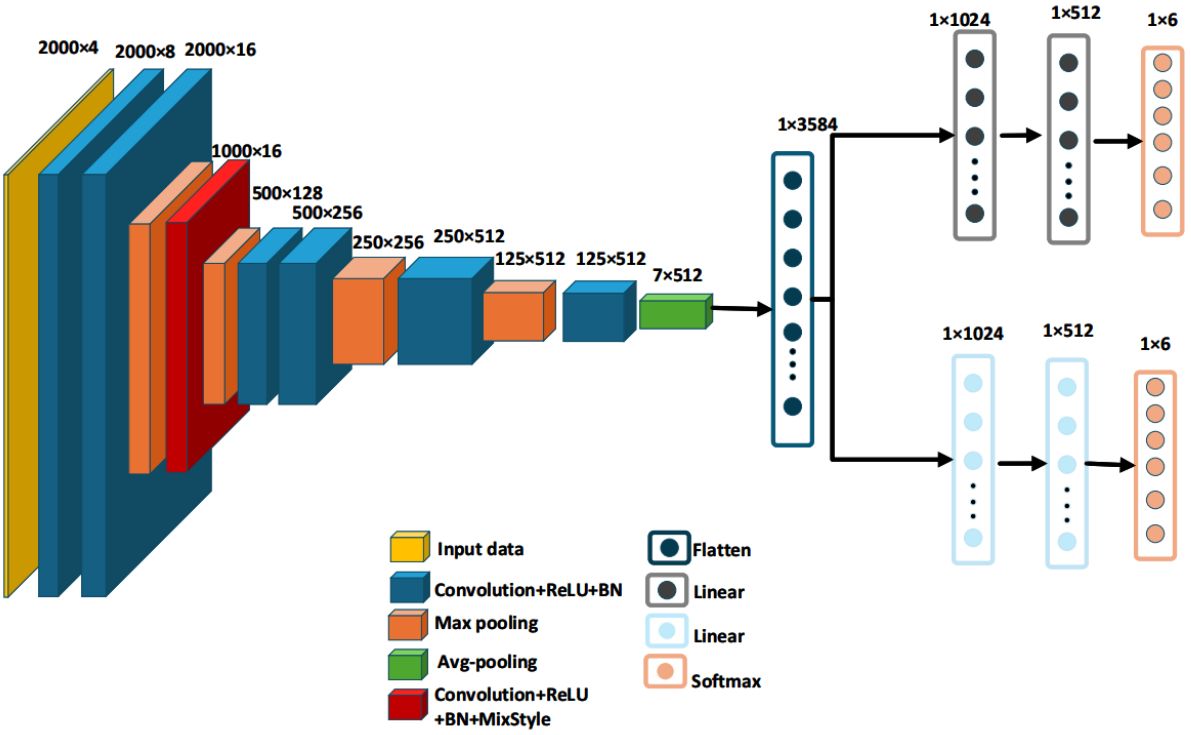


Figure 4. The detailed configuration of the branch structure in neural networks.

We maintain identical hyperparameter settings across all numerical simulations. The complete hyperparameter configuration is provided in Table 2.

The training objective jointly updates the feature extractor, damage classifier, and domain discriminator. MixStyle is applied during feature extraction, while physical damage labels and pseudo-domain labels supervise the localization and domain-discrimination objectives, respectively.

Table 2. The specific hyper parameters of the network architectures.

Learning rate	**5e−5**
Batch size	32
L^2 rate	0.001
Patience	100
Weight	0.008
α	0.1
λ-weight	0.5

3.2. Numerical experiment

To evaluate the generalization performance of source-domain-trained models for target domain damage localization without using any target domain data during training, we established one source domain model and four distinct target domain models. All models incorporate controlled variations in key structural parameters—including length, elastic modulus, material density, and longitudinal stiffness. As shown in Table 3, this tiered experimental design systematically increases parametric divergence from the source domain while maintaining physically realistic variation ranges, ultimately producing one source domain model and four progressively more challenging target domain models for comprehensive generalization testing.

Table 3. Source domain dataset and target domain dataset sizes and parameter settings.

Data type	size	Bias
Multi-source domain dataset	(4800,2000,4)	± 0.5%
Target domain dataset 1	(4800,2000,4)	± 0.7%
Target domain dataset 2	(4800,2000,4)	± 1%
Target domain dataset 3	(4800,2000,4)	± 2%
Target domain dataset 4	(4800,2000,4)	± 3%

To evaluate damage-localization performance, we compared a conventional CNN, a domain-adversarial baseline, and the proposed MixStyle-ADGN model across four target domains. The models were evaluated under the same data protocol. As shown in Figure 5, the proposed model maintained higher accuracy than the selected baselines as the parameter difference between the source and target domains increased.

The performance comparison across the four target domains is shown in Figure 5. In Target Domain 1, the CNN, domain-adversarial baseline, and MixStyle-ADGN model achieved accuracies of 90.1%, 92.2%, and 94.7%, respectively. The corresponding values in Target Domain 2 were 89.4%, 91.7%, and 93.1%. In Target Domain 3, the accuracies were 68.3%, 70.9%, and 74.4%, respectively. In Target Domain 4, the CNN

achieved 58.7%, whereas MixStyle-ADGN achieved 67.7%. These results show that the proposed model retained a higher localization accuracy than the selected baselines under the tested parameter variations.

As the parameter variation increased, the accuracy of the MixStyle-ADGN model decreased gradually but remained higher than that of the selected baselines. MixStyle changes the statistics of intermediate features, and the domain discriminator provides an additional training signal for reducing dependence on domain-specific characteristics. The present experiments do not separate the contributions of these two components; therefore, the observed improvement is attributed to the combined model rather than to either component alone.

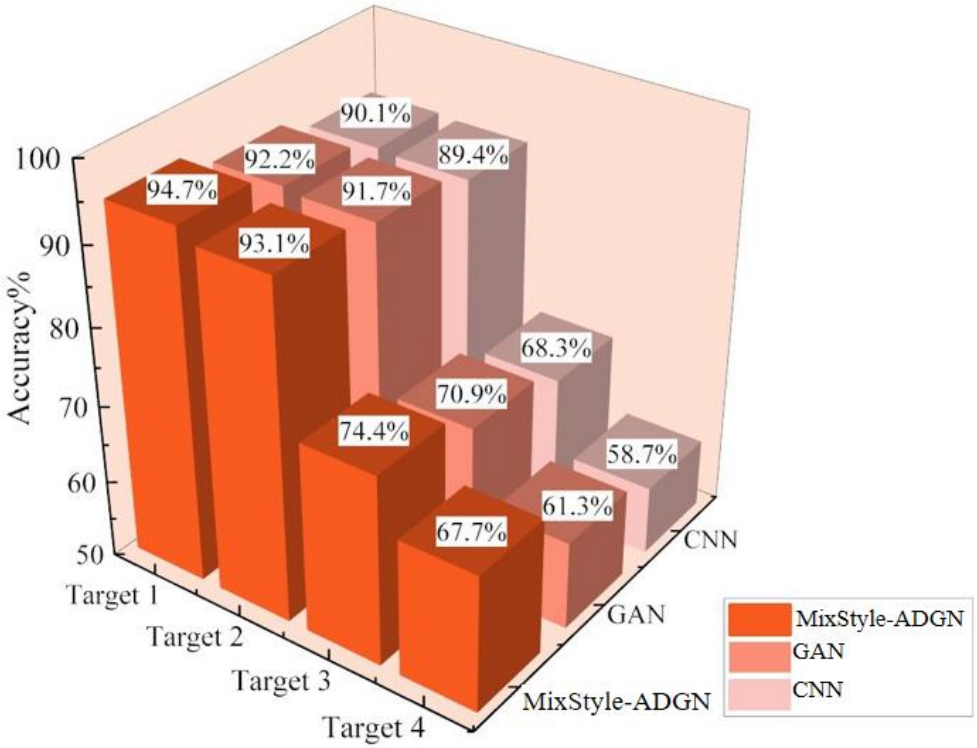


Figure 5. Damage-localization accuracy of the CNN, domain-adversarial baseline, and MixStyle-ADGN models in the four target domains.

To better demonstrate the effectiveness of the MixStyle-ADGN model, we employ confusion matrix analysis for quantitative evaluation. Figures 6 and 7 presents the performance comparison between the CNN and MixStyle-ADGN models on Target Domain 4, revealing significant differences in classification behavior. The CNN model achieves approximately 60% accuracy along the confusion matrix diagonal, with particularly poor performance (38%) in identifying undamaged states. In contrast, the MixStyle-ADGN model demonstrates superior damage localization capability with 70% overall accuracy, and notably achieves 75% accuracy for undamaged cases—a 40% improvement over CNN. This performance gap stems from the substantial domain shift between Source Domain and Target Domain 4 data. The MixStyle-ADGN model actively blends style statistics across samples during training. This forces the model to focus on structurally meaningful content features rather than superficial style characteristics, resulting in more robust feature representations and significantly improved identification of both damaged and undamaged states across domains.

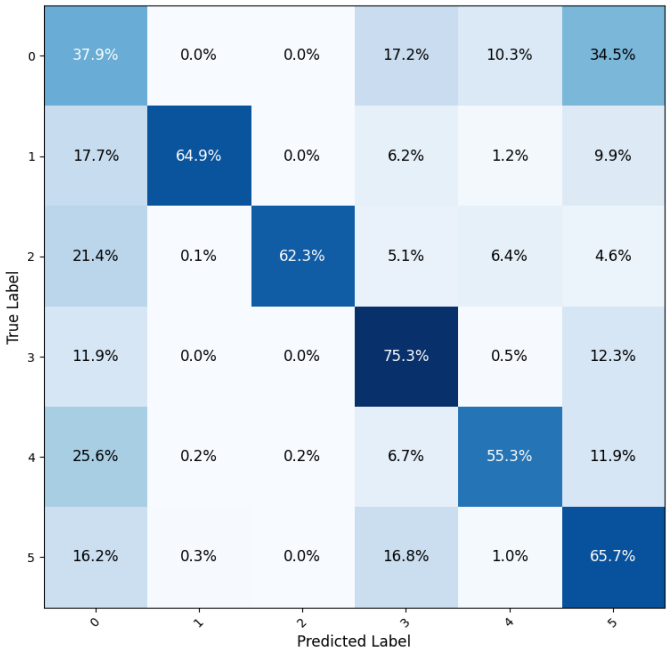


Figure 6. The confusion matrix of the CNN model in the target domain 4.

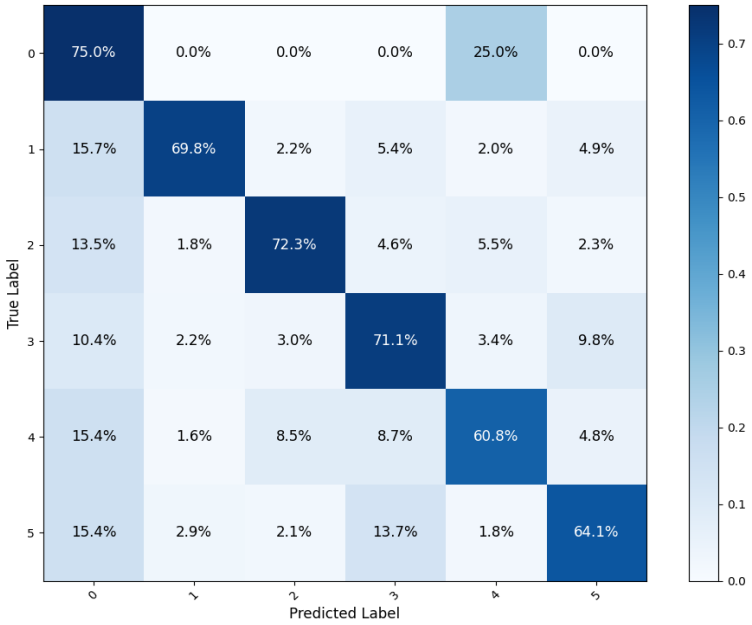
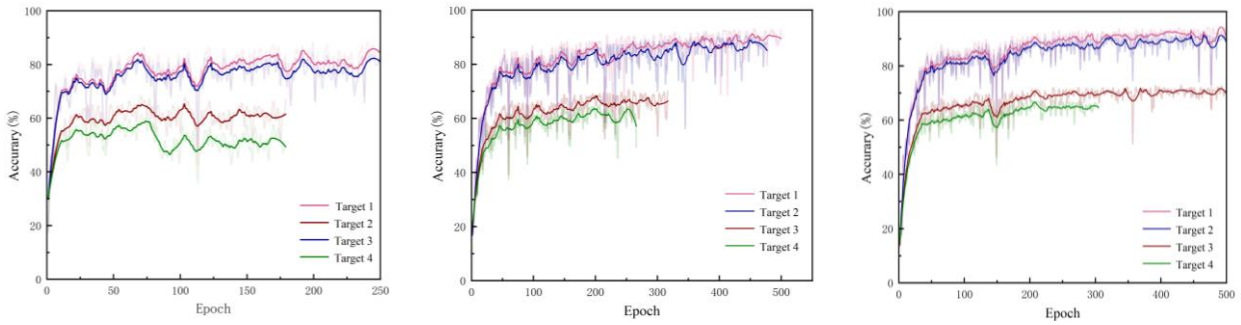


Figure 7. The confusion matrix of the MixStyle-ADGN model in the target domain 4.

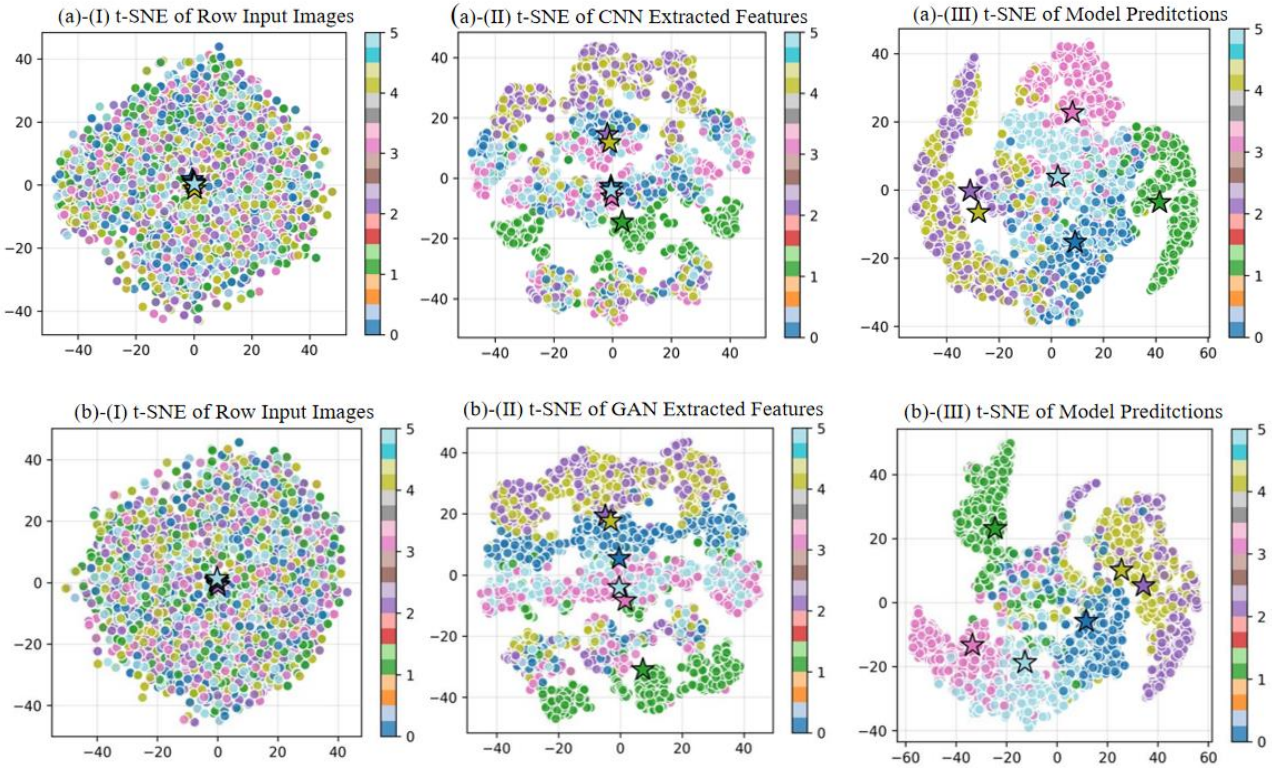
The training dynamics and model convergence characteristics can be further understood by examining the target domain test accuracy trajectories throughout the learning process. As shown in Figure 8, while the CNN performance plateaus after 100 cycles, indicating limited capacity for further feature refinement, and the domain-adversarial baseline shows unstable convergence with fluctuating accuracy, our MixStyle-ADGN model achieves sustained, steady accuracy gains. This continuous improvement stems directly from the enhanced adversarial training framework, which progressively strengthens the model's ability to extract domain-invariant representations, thereby consistently improving its generalization capability across varying structural conditions.



(a) (b) (c)

Figure 8. Test accuracy of the CNN, domain-adversarial baseline, and MixStyle-ADGN models.

The comparative t-SNE visualization in Figure 9 clearly demonstrates the enhanced domain generalization capability of our proposed MixStyle-ADGN approach relative to CNN and domain-adversarial methods. the CNN feature and the domain-adversarial feature space exhibits dispersed data distributions with substantial cluster overlap, indicative of the model's limited capacity to accommodate domain variations. In marked contrast, the lower panel reveals that MixStyle-ADGN features display significantly improved clustering characteristics. The five-pointed star representing the centroid of each category, computed as the mean of all samples within that category, provides a clear visual indication of the overall position and compactness of each cluster in the embedded space. This improved feature organization directly results from the method's two key technical innovations: MixStyle augmentation generates features with robust style invariance, maintaining consistent spatial distributions of simulated damage patterns despite parametric variations, while adversarial training produces more compact and well-separated decision boundaries, as clearly visible in subfigure (c).



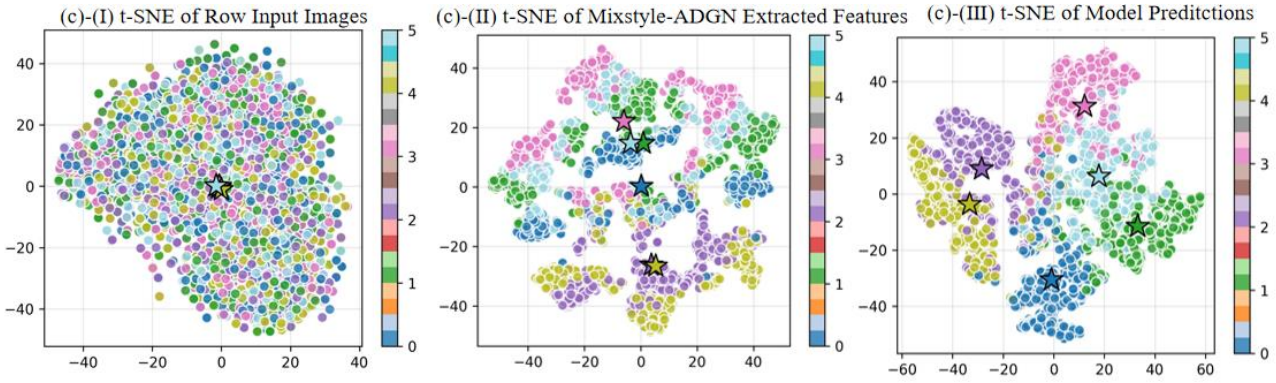


Figure 9. Comparison of feature distributions for target 3. **(a)** t-SNE of CNN-extracted features; **(b)** t-SNE of GAN-extracted features; **(c)** t-SNE of MixStyle-ADGN-extracted features; **(I)** Raw input data distribution; **(II)** Feature space visualization; **(III)** Prediction probability distribution (The five-pointed star represents the centroid of each category, computed as the mean of all samples within that category, to visually indicate its overall position in the space).

4. Experimental verifications

The numerical results show that the localization accuracy of the CNN decreases as the difference between the source and target domains increases. MixStyle-ADGN achieved higher accuracy than the selected baselines in the numerical tests. These results do not by themselves establish performance under operational bridge conditions, so a laboratory experiment was conducted as a further simulation-to-experiment evaluation.

The experimental methodology comprises four key phases. First, we establish a numerical model based on the geometric, material, and dynamic parameters obtained from the physical simply supported beam specimen. The numerical model is subsequently updated using modal test results from the actual beam to ensure parameter fidelity. Second, both the physical beam and numerical model undergo excitation tests to collect baseline (undamaged) response data and corresponding labels. Third, controlled damage is introduced at specific locations on both systems, with subsequent dynamic testing generating labeled datasets for various damage scenarios. For model validation, we implement a rigorous cross-domain evaluation protocol: during training, the network exclusively processes source domain data from the numerical model (both undamaged and damaged states), while testing employs only experimental data from the physical beam as the target domain. This approach directly assesses the model's ability to generalize from simulation to reality. For comparative analysis, we evaluate the CNN architecture and domain-adversarial architecture that mirror the MixStyle-ADGN's feature extractor (excluding the MixStyle layer) while maintaining identical classifier structures, enabling direct performance comparison of the domain generalization components.

4.1. Experimental setup

The experimental setup, illustrated in Figure 10a, Figure 11 and Table 4, features a steel simply supported beam with dimensions of 1.35 meters in length, 79 mm in width, and 7.6 mm in height, possessing a material density of 7896 kg/m. To achieve ideal simply supported boundary conditions, ball bearing supports are installed at both ends of the beam (Figure 11c). The basic properties of this beam

model are summarized in Table 5. Its first five natural frequencies are 10.5 Hz, 38.2 Hz, 82.28 Hz, 152.42 Hz, and 215.2 Hz, corresponding to the first through fifth modes. For comprehensive vibration monitoring, the beam is divided longitudinally into five equal segments, with four accelerometers strategically positioned along its span. Impact hammer excitation is employed to induce structural vibrations, with the resulting acceleration responses being recorded by a dedicated data acquisition system.

To accurately simulate real bridge damage mechanisms in our experimental setup, we introduced controlled damage to the simply-supported steel beams through precision cutting, creating localized stiffness reductions analogous to structural cracks in actual bridges. The damage severity was systematically quantified by varying cut depths, with five distinct damage levels established at 10%, 20%, 30%, 40%, and 50% stiffness reduction—corresponding to precise cut depths of 7.9 mm, 15.8 mm, 23.7 mm, 31.6 mm, and 39.5 mm respectively across the beam's width. As shown in Figure 10a, four sensors are set up on the beam in this experiment, and the beam is divided into five sections. Cutting was carried out at the center of the first, second and third sections. In this experiment, only single-exit damage was considered, so there are three possible damage locations. This experiment adopts the excitation method of hammering. When the structure is intact, the hammering points are located at the center of all sections of the beam, with a total of 5 hammering points. When the structure is damaged, the hammering point of the damaged section is at the non-damaged position, while the hammering point of other sections remains at the center of the section. Set 10 hammering strikes for each hammering point and record all signals. Therefore, we can obtain 900 sets of experimental data ((5 degrees of damage × 3 damage positions × 5 loading positions + 1 intact state × 5 loading positions × 3 beams) × 10 hammering).

The cutting procedure provides a controlled way to produce localized stiffness reduction. It does not reproduce all damage mechanisms found in bridges, including corrosion, fatigue cracking, concrete cracking, connection damage, bearing deterioration, or multiple interacting defects.

Figure 10. Experimental setup. **(a)** Simply supported beam and damage location; **(b)** Data acquisition system; **(c)** Ball bearing support.

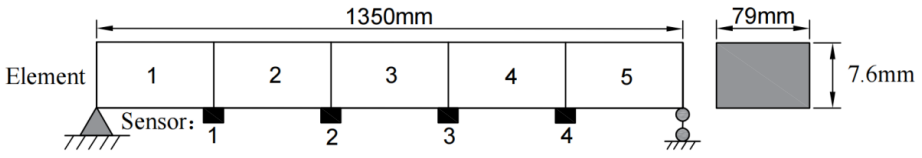


Figure 11. The structure of the finite element model.

Table 4. Basic properties of simply-support beam model.

length	**1.35m**
width	79 mm
height	7.6 mm
Material	Steel
Density	7896 kg/m^3
Boundary condition	Ball bearing support
Sensor model	Donghua 1A001
Acquisition System	DH5922
Sampling frequency (modal test)	1000 Hz
Incentives	Hammering

Table 5. Basic properties of beam model.

Model order	**first**	**Second**	**Third**	**Fourth**	**Fifth**
Frequency (Hz)	10.5	38.2	82.28	152.42	215.2

4.2. Model update

While theoretical models often rely on idealized assumptions, real-world structures inevitably exhibit manufacturing and material variations that lead to discrepancies between simulation and experiment. To address this for deep learning, we use an updated finite element modeling approach that generates training data with rich feature diversity through realistic damage simulation. Model updating systematically adjusts five physical parameters—beam length, width, height, Young's modulus, and density—by optimizing an objective function, closing the gap between simulation and actual behavior while preserving the ability to simulate realistic damage for effective feature learning.

$$L = \sum_{i=1}^{5} \left\| \frac{f_t^{(i)} - f_m^{(i)}}{f_t^{(i)}} \right\|_2 + \lambda \sum_{i=1}^{5} (1 - MAC(\varphi_t^{(i)}, \varphi_m^{(i)})), \tag{22}$$

$$MAC(\varphi_t^{(i)}, \varphi_m^{(i)}) = \frac{(\varphi_t^{(i)} \cdot \varphi_m^{(i)})^2}{\left\| \varphi_t^{(i)} \right\|_2^2 \left\| \varphi_m^{(i)} \right\|_2^2}, \tag{23}$$

L is the objective function (loss function) of Model Updating. The formula is divided into two parts, which respectively measure the frequency error and the modal shape error. Table 6 lists the natural frequencies of the numerical model after model correction and the relative errors of the first five natural frequencies.

Table 6. The natural frequency of the numerical model and the relative error compared with the actual structure.

Model order	**first**	**Second**	**Third**	**Fourth**	**Fifth**
Freuency (Hz)	10.44	38.20	82.37	148.98	232.02
Relative error	−0.6190%	0.0042%	2.5409%	−2.2573%	7.8169%

In the finite element simulation, the applied impact load adopts the form of a half-period sine wave, with an amplitude of 12 Newtons, the initial data of the impact is 0.02 seconds, and the frequency is 200. Its mathematical expression is:

$$f(x) = \begin{cases} A\sin(\omega(t - t_0)), t_0 \le t \le t_0 + \dfrac{\pi}{\omega}, \\ 0, otherwise \end{cases} \tag{24}$$

In the numerical model, we simulate the damage by reducing the element stiffness of the beam, setting five damage positions and five degrees of damage. The beam is divided into 5 sections. Under each working condition, 32 hammering loads are applied to each section. 4800 sets of data can be obtained from the finite element model. ((5 damage locations + 1 intact state) × 5 damage degrees × 5 sections × 32 loads).

4.3. Data processing

The experimental and numerical data underwent systematic preprocessing to ensure analytical consistency. Experimental measurements were first processed using an 8 Hz high-pass filter to enhance high-frequency signal components relevant to damage detection. Conversely, finite element simulation outputs were treated with a 250 Hz low-pass filter (accounting for the first five structural modes, with the fifth natural frequency at 232 Hz) and down sampled to 500 Hz to match experimental sampling rates. All datasets were then normalized and uniformly reformatted into a standardized [2000 × 4] matrix structure, where 2000 represents time-series length and 4 corresponds to measurement channels. This rigorous preprocessing pipeline preserves critical signal features while establishing a consistent data framework for subsequent comparative analysis between numerical predictions and experimental observations.

4.4. The architecture of neural network

Despite model optimization, significant discrepancies remain between simulated and experimental results. To bridge this gap, our neural network architecture is explicitly designed for generalization, enabling models trained solely on simulation data to perform well on experimental measurements. This is achieved through a deepened network structure, detailed in Figures 12 and 13, with optimized hyperparameters fully listed in Table 7.

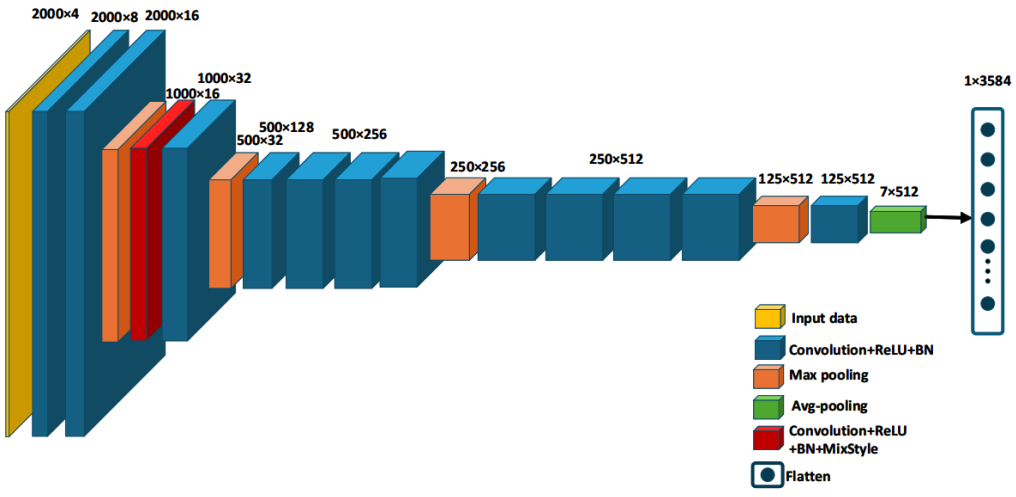


Figure 12. The architecture of the feature extractor.

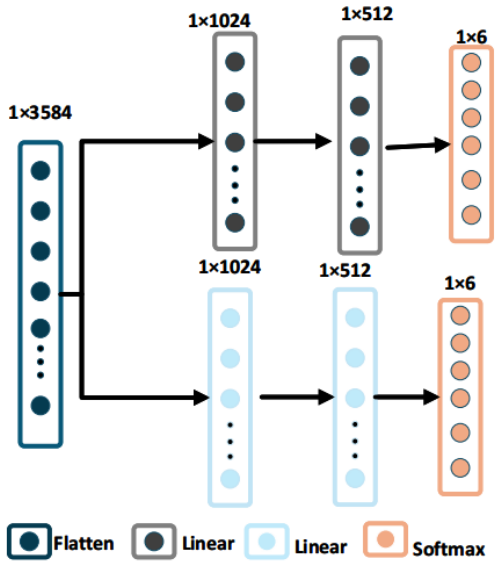


Figure 13. The architecture of the damage classifier and domain determiner.

Table 7. The specific hyperparameters of the network architectures.

Learning rate	**1e−5**
Batch size	32
L^2 rate	0.001
Patience	200
Weight	0.004
α	0.5
λ-weight	0.3

The comparative performance evaluation, as illustrated in Figure 14, demonstrates the superior accuracy of the MixStyle-ADGN model over CNN and domain-adversarial baseline, with the former achieving approximately 80% accuracy compared to the latter's 50%. This significant performance gap substantiates that the MixStyle-ADGN learns more domain-invariant features essential for cross-domain damage localization.

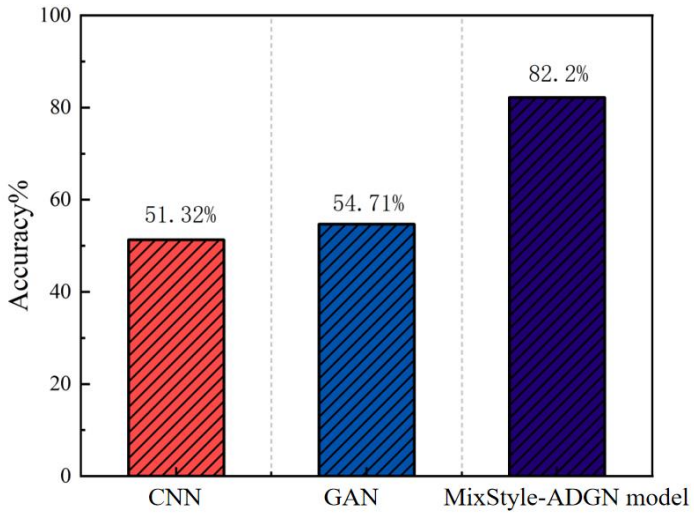


Figure 14. Accuracy of the CNN, domain-adversarial baseline, and MixStyle-ADGN models.

Detailed analysis of the confusion matrices (Figure 15) reveals critical insights: the CNN performs particularly poorly in undamaged state identification and at the third damage location, with accuracy rates substantially lower than other categories. These shortcomings stem from two fundamental limitations: (1) inadequate feature generalization capacity to bridge the substantial source-target domain discrepancy, and (2) insufficient model complexity to learn discriminative features for challenging categories. While the domain-adversarial model shows improved performance in certain categories—such as a high 99% recognition rate for label 1—it still exhibits notable inconsistencies, particularly with poor generalization for labels 0 and 3. The MixStyle-ADGN overcomes these limitations through its dual mechanism of style-augmented feature learning and adversarial domain adaptation, enabling robust discrimination between undamaged and damaged states across all locations despite significant domain shifts.

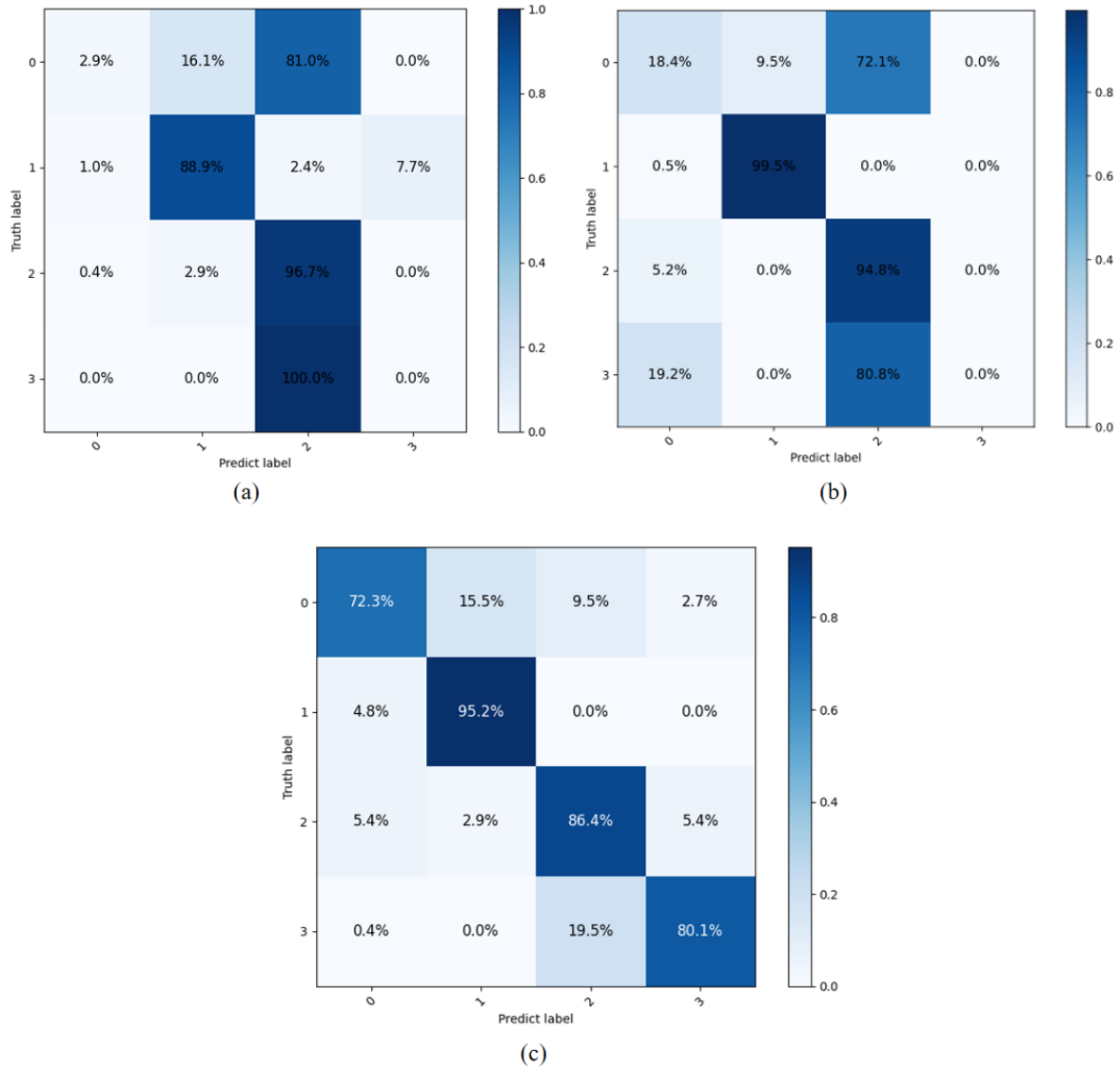


Figure 15. The confusion matrix of the CNN **(a)**, the GAN **(b)** and the MixStyle-ADGN model **(c)**.

The training dynamics revealed in Figure 16 further elucidate this advantage: while the CNN shows rapid initial convergence within 50 cycles (even temporarily surpassing the domain generalization model), it plateaus at around 40% accuracy, indicating limited capacity for further feature refinement. In contrast, the MixStyle-ADGN maintains steady accuracy improvements throughout training, ultimately establishing its performance superiority.

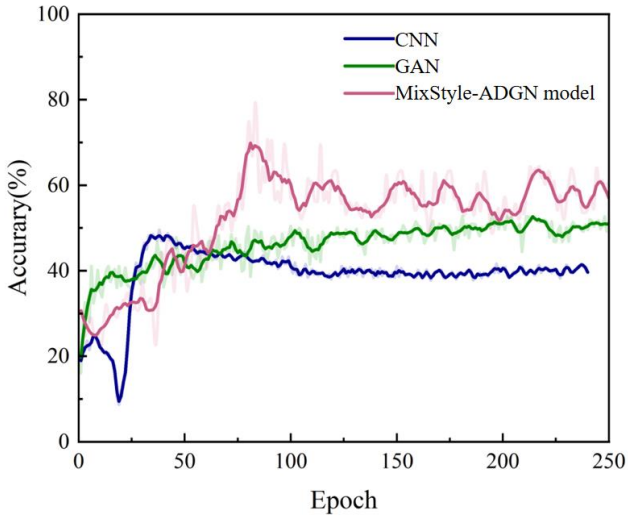
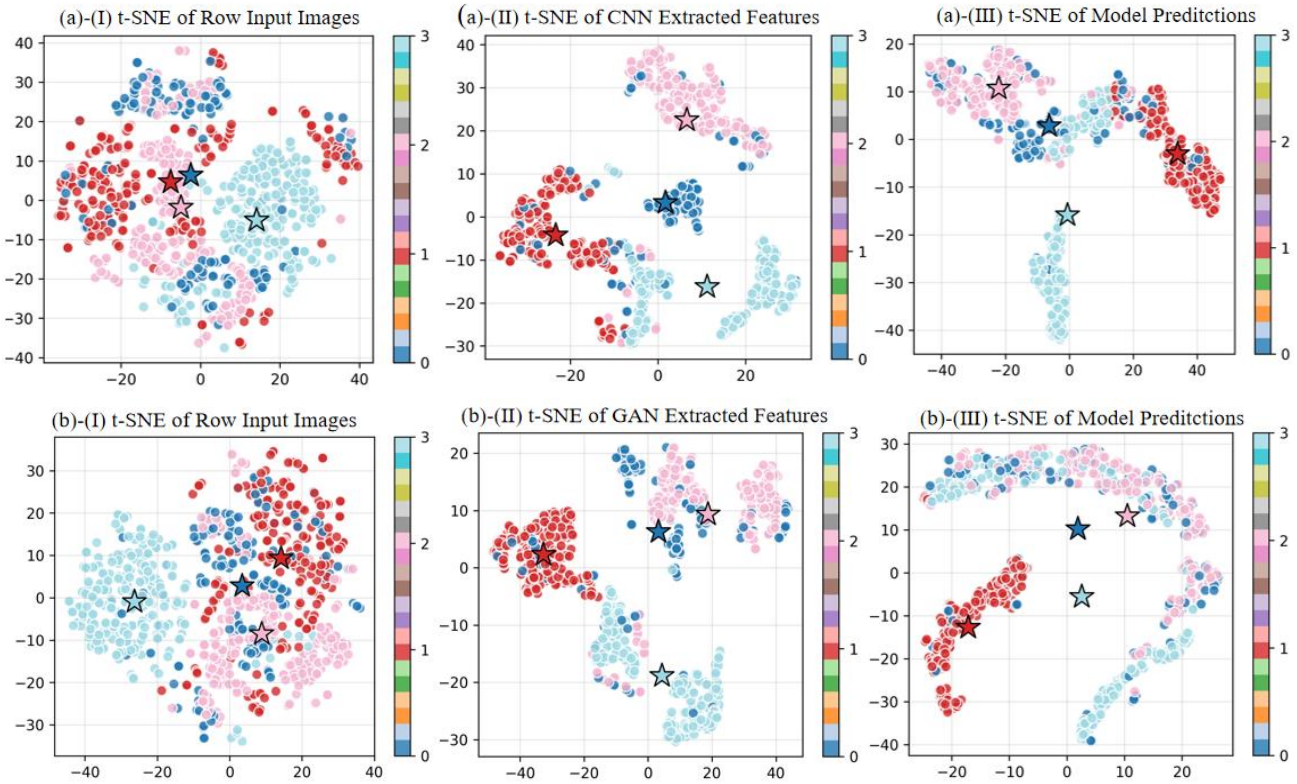


Figure 16. Test accuracy of the CNN, domain-adversarial baseline, and MixStyle-ADGN models.

The t-SNE visualization in Figure 17 compares feature distributions from a conventional CNN, a domain-adversarial baseline, and the proposed MixStyle-ADGN. The CNN features appear scattered and overlapping, indicating limited representational capacity. The GAN shows moderately tighter clusters but retains noticeable overlap between some damage states, reflecting incomplete domain adaptation. In contrast, MixStyle-ADGN displays clearer category separation and more compact groupings, demonstrating improved discriminative learning through style mixing. This is further supported by the prediction probability distribution (Ⅲ), where MixStyle-ADGN's results align more consistently with feature clustering, indicating more accurate feature-to-category mapping. The five-pointed stars mark each category's centroid. Together, these results confirm that MixStyle-ADGN surpasses both CNN and GAN in feature extraction quality and classification performance.



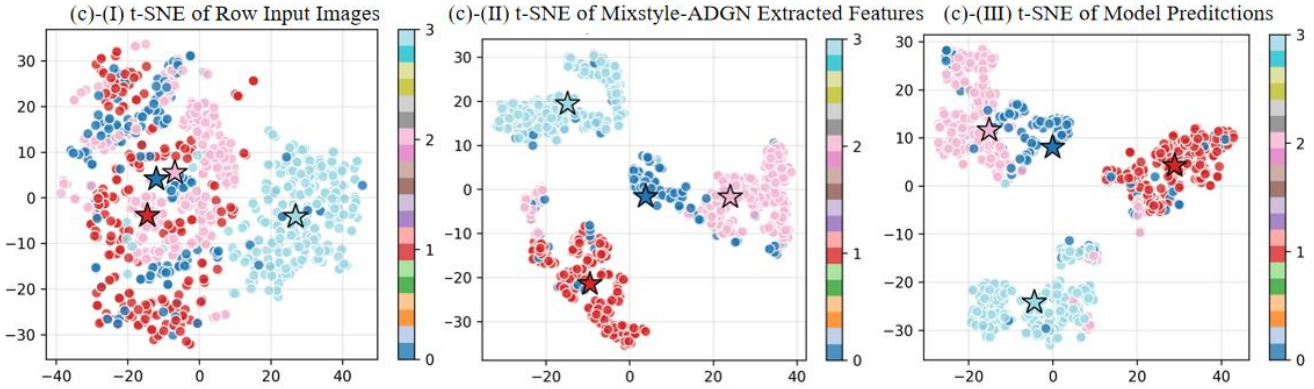


Figure 17. Feature distributions for the experimental data obtained with CNN **(a)**; the domain-adversarial baseline **(b)**; and MixStyle-ADGN **(c)**.

The validation is limited to a simply supported beam and controlled damage scenarios. Continuous girders, cable-stayed bridges, and operational bridges involve more complex coupling, boundary conditions, sensor layouts, environmental variability, missing data, and measurement noise. Applying the method to such structures would require additional source-domain simulations and validation. The present study evaluates localization only. Severity quantification could be pursued with a regression or ordinal-classification head, but its cross-domain performance would require separate experiments.

5. Conclusions

This study addresses the critical challenge of data scarcity in Structural SHM by developing a novel Domain Generalization (DG) approach for bridge damage localization. The key findings and contributions are summarized as follows:

(1) Proposed a DG framework combining MixStyle data augmentation and adversarial training

(2) Developed domain-invariant feature learning without requiring target domain data

(3) Established a style-transfer mechanism to enhance source domain diversity

(4) Ensure robustness in real-world environments and successful application to different damage scenarios.

Our research demonstrates that while traditional Domain Adaptation (DA) methods remain constrained by their dependence on target domain data, the proposed DG method effectively overcomes this limitation by learning transferable features exclusively from numerical simulation data (source domain). Through systematic numerical studies accounting for various uncertainties - including stiffness variation, geometric tolerances, and material property fluctuations - we verified the model's superior generalization capability compared to CNN and GAN, particularly under significant domain shifts.

Experimental validation on simply supported beam structures confirmed the practical viability of our approach, showing consistent damage localization accuracy (approximately 82%) despite substantial domain discrepancies, outperforming CNN baselines by 30%. This performance advantage stems from the model's ability to maintain discriminative power for both damaged and undamaged states (72%ccuracy for undamaged cases) through learned domain-invariant representations. The study acknowledges two primary limitations requiring future investigation: (1) the need to extend validation to complex bridge systems (e.g., continuous beams, cable-stayed bridges) with more sophisticated dynamic characteristics.

(2) the challenge of environmental noise and measurement inaccuracies in real-world monitoring scenarios. Subsequent research should focus on developing more robust feature extraction algorithms and verifying the method with actual bridge monitoring data to advance toward practical implementation.

This work provides a framework for damage localization when target-domain damage labels are unavailable. The results support its use under the numerical and laboratory conditions examined here. Further validation is needed before applying the method to operational bridges or comparing it broadly with other domain-generalization methods.

Data availability statement

The raw numerical and experimental data generated during this study are not publicly available because they are part of an ongoing research project that has not yet been concluded. These data are considered proprietary to the research group and are subject to continued analysis and validation. Access to the data can be requested from the corresponding author on a case-by-case basis and will be considered subject to reasonable request and internal approval.

Declaration of generative AI and AI-assisted technologies

During the preparation of this work, the authors used ChatGPT solely for language polishing and readability improvement. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the final content of the publication.

Acknowledgments

The authors gratefully acknowledge financial support from the funding of the National Natural Science Foundation of China (Grant No. 52178289 and Grant No. 52408320); Guangdong Provincial Key Laboratory of Intelligent Disaster Prevention and Emergency Technologies for Urban Lifeline Engineering (2022) (Grant No. 2022B1212010016); Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024B1515120032); Youth S&T Talent Support Programme of Guangdong Provincial Association for Science and Technology (Grant No. SKXRC2025430).

Authors' contribution

Conceptualization, X.W. and Y.L.; Methodology, X.W. and Y.L.; Software, Z.N.; Validation, J.L.; Formal analysis, X.W.; Investigation, X.W. and Z.N.; Resources, H.M.; Data curation, X.W.; Writing—original draft, X.W.; Writing—review and editing, Y.L. and J.L.; Visualization, J.L.; Supervision, H.M. and Z.N.; Project administration, H.M.; Funding acquisition, Y.L. and Z.N. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Farrar CR, Worden K. An introduction to structural health monitoring. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 2007, 365(1851):303–315.

[2] Teughels A, Maeck J, De Roeck G. Damage assessment by FE model updating using damage functions. *Comput. Struct.* 2002, 80(25):1869–1879.

[3] Yan Y, Cheng L, Wu Z, Yam L. Development in vibration-based structural damage detection technique. *Mech. Syst. Signal Process.* 2007, 21(5):2198–2211.

[4] Simoen E, De Roeck G, Lombaert G. Dealing with uncertainty in model updating for damage assessment: A review. *Mech. Syst. Signal Process.* 2015, 56:123–149.

[5] Xiong Y, Chen W, Tsui KL, Apley DW. A better understanding of model updating strategies in validating engineering models. *Comput. Methods Appl. Mech. Eng.* 2009, 198(15–16):1327–1337.

[6] Ferson S, Oberkampf WL, Ginzburg L. Model validation and predictive capability for the thermal challenge problem. *Comput. Methods Appl. Mech. Eng.* 2008, 197(29–32):2408–2430.

[7] Levin RI, Lieven NA. Dynamic finite element model updating using neural networks. *J. Sound Vib.* 1998, 210(5):593–607.

[8] Ren W, Chen H. Finite element model updating in structural dynamics by using the response surface method. *Eng. Struct.* 2010, 32(8):2455–2465.

[9] Goller B, Schueller GI. Investigation of model uncertainties in Bayesian structural model updating. *J. Sound Vib.* 2011, 330(25):6122–6136.

[10] Nie Z, Hao H, Ma H. Using vibration phase space topology changes for structural damage detection. *Struct. Health Monit.* 2012, 11(5):538–557.

[11] Nie Z, Guo E, Li J, Hao H, Ma H, *et al.* Bridge condition monitoring using fixed moving principal component analysis. *Struct. Control Health Monit.* 2020, 27(6):e2535.

[12] Li X, Wang L, Law SS, Nie Z. Covariance of dynamic strain responses for structural damage detection. *Mech. Syst. Signal Process.* 2017, 95:90–105.

[13] Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M, Inman DJ. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* 2017, 388:154–170.

[14] Avci O, Abdeljaber O, Kiranyaz S, Hussein M, Inman DJ. Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks. *J. Sound Vib.* 2018, 424:158–172.

[15] Lin Y, Nie Z, Ma H. Structural damage detection with automatic feature-extraction through deep learning. *Comput. Aided Civ. Infrastruct. Eng.* 2017, 32(12):1025–1046.

[16] Lin Y, Nie Z, Ma H. Dynamics-based cross-domain structural damage detection through deep transfer learning. *Struct. Control Health Monit.* 2022, 37(1):24–54.

[17] Nie Z, Xu S, Chen K, Xu L, Lin Y, *et al.* Damage detection in bridge via adversarial—based transfer learning. *Struct. Control Health Monit.* 2024, 2024(1):5548218.

[18] Giglioni V, Poole J, Venanzi I, Ubertini F, Worden K. A domain adaptation approach to damage classification with an application to bridge monitoring. *Mech. Syst. Signal Process*. 2024, 209:111135.

[19] Li Z, Weng S, Zhu H, Lei A. Cross-domain damage detection through partial conditional adversarial domain adaptation. *Mech. Syst. Signal Process*. 2025, 225:112263.

[20] Long M, Zhu H, Wang J, Jordan MI. Unsupervised domain adaptation with residual transfer networks. In *Proceedings of the Advances in Neural Information Processing Systems*, Barcelona, Spain, December 5–10, 2016, pp. 136–144.
[21] Wang J, Lan C, Liu C, Ouyang Y, Qin T, *et al.* Generalizing to unseen domains: a survey on domain generalization. *IEEE Trans. Knowl. Data Eng.* 2022, 35(8):8052–8072.
[22] Fan Z, Xu Q, Jiang C, Ding SX. Deep mixed domain generalization network for intelligent fault diagnosis under unseen conditions. *IEEE Trans. Ind. Electron.* 2023, 71(1):965–974.
[23] Wang X, Jiao J, Zhou X, Xia Y. Knowledge distillation-based domain generalization enabling invariant feature distributions for damage detection of rotating machines and structures. *Reliab. Eng. Syst. Saf.* 2025, 257:110842.
[24] Ragab M, Chen Z, Zhang W, Eldele E, Wu M, *et al.* Conditional contrastive domain generalization for fault diagnosis. *IEEE Trans. Instrum. Meas.* 2022, 71:1–12.
[25] Nie Z, Gao Y, Zhu Z, Chen Y, Ma H. Structural damage detection with domain generalization based on deep transfer learning. *Eng. Struct.* 2025, 343:121065.
[26] Wang J, Lan C, Liu C, Ouyang Y, Qin T, *et al.* Generalizing to unseen domains: a survey on domain generalization. *IEEE Trans. Knowl. Data Eng.* 2022, 35(8):8052–8072.
[27] Gatys LA, Ecker AS, Bethge M. Image style transfer using convolutional neural networks. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 27–30, 2016, pp. 2414–2423.
[28] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shif. In *Proceedings of the International Conference on Machine Learning*, Lille, France, July 6–11, 2015, pp. 448–456.
[29] Kohler J, Daneshmand H, Lucchi A, Zhou M, Neymeyr K, *et al.* Towards a theoretical understanding of batch normalization. *Stat* 2018, 1050:27.
[30] Ulyanov D, Vedaldi A, Lempitsky V. Instance normalization: the missing ingredient for fast stylization. *arXiv* 2016, arXiv:1607.08022.
[31] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October 22–29, 2017, pp. 1501–1510.
[32] Zhou K, Yang Y, Qiao Y, Xiang T. Domain generalization with MixStyle. *arXiv* 2021, arXiv:2104.02008.
[33] Burkardt J. K-means clustering. Virginia tech. 2009. Available: https://people.sc.fsu.edu/~jburkardt/classes/isc_2009/clustering_kmeans.pdf (accessed on 31 July 2026)
[34] Kodinariya TM, Makwana PR. Review on determining number of cluster in K-means clustering. *Int. J.* 2013, 1(6):90–95.
[35] Fu J, Zhong Y, Yang F. Adversarial domain generalization with MixStyle. In *Proceedings of the 2022 International Conference on Advanced Robotics and Mechatronics (ICARM)*, Chongqing, China, July 8–11, 2022, pp. 379–385.
[36] Dumoulin V, Shlens J, Kudlur M. A learned representation for artistic style. *arXiv* 2016, arXiv:1610.07629.
[37] Kim T, Oh J, Kim NY, Cho S, Yun SY. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv* 2021, arXiv:2105.08919.
[38] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, *et al.* Generative adversarial networks. *Commun. ACM* 2020, 63(11):139–144.