A Transfer Learning-Based Method for Cross-Scenario Operation Step Generation and Generalization Optimization in Distribution Networks

Fangzhou Hao[1,*], Shibo Li[1], Jiangwei Wu[1], Shuhua Chen[1]

[1]Guangzhou Power Supply Bureau of Guangdong Power Grid Co., Ltd., Guangzhou, Guangdong, China

Abstract

In response to the issue of limited generalization in automatic generation of operation steps across different stations and operating conditions in distribution networks, this paper proposes a temporal-topological joint modeling method driven by transfer learning. The method takes device state sequences and network topology as inputs, utilizing a Transformer to extract temporal features and Graph Convolutional Networks (GCN) to model topological constraints. During the transfer phase, a "freeze the lower layers – progressively unfreeze" fine-tuning strategy is applied, combined with distribution alignment and domain adversarial learning to achieve cross-scenario adaptation. Evaluations on the Caltech 28-Bus Digital Twin dataset show that, compared to No Transfer and Fine-tuning, the proposed method reduces the Mean Squared Error (MSE) from 0.15 to 0.06, the Mean Absolute Error (MAE) from 0.39 to 0.245, and the Generalization Performance Index (GPI) to approximately 0.47 in the target domain. When only 10% of the target domain samples are used, the Relative Normalized Squared Error (RNSE) is approximately 0.55. For the sake of generation quality, BLEU is 0.83, and the Semantic Consistency Score (SCS) is 0.92. These findings show that the suggested framework can reliably produce operation steps that are semantically coherent and logically consistent across various topologies and operating conditions, offering a workable technological route for the automation of operation step generation in distribution network operation scheduling.

Keywords: network operation step generation; Transfer learning; Cross-scenario generalization; Domain adaptation; Transformer; Graph Neural Networks

Received on 04 December 2025, accepted on 30 January 2026, published on 11 August 2026

doi: 10.4108/ew.13906

1. Introduction

The distribution network is an essential part of the power system, and the efficiency of maintenance and the dependability of the power supply are directly impacted by its operational safety and intelligence level [1,2]. Operation steps are crucial execution guidelines that directly affect the capacity to avoid mistakes and guarantee safety during the distribution network's upkeep and operation. However, conventional approaches that rely on human experience to compile operation steps are ineffective, prone to mistakes, and have limited flexibility in complex operational environments [3,4].

Deep learning has been used in recent years to generate operation steps automatically, reaching a level of automation based on past processes. However, the generalization performance of current approaches is still restricted in applications across various sites and operating scenarios [5]. The main challenges are: (1) substantial differences between distribution network operation scenarios, with large distribution disparities in network topology, operational status, and step text features across various domains, making direct model transfer challenging [6]; and (2) deep learning models heavily rely on large

*Corresponding author. Email: 16676676716@163.com

amounts of labeled data, and it is frequently challenging to obtain sufficient and high-quality samples for cross-scenario tasks [7].

An efficient way to deal with these issues is through transfer learning. This approach can enhance generalization performance when there are substantial distribution differences or insufficient data by using knowledge from the source domain and adaptively modifying it for the target domain [8]. In tasks like fault diagnosis and load forecasting in power systems, transfer learning has made some progress, but systematic research on cross-scenario generalization in operation step generation is still lacking.

Furthermore, improving the semantic consistency of operation step generation has been made feasible by the advancement of natural language processing (NLP) and semantic modeling techniques [9]. Deep semantic features can be extracted from procedural texts by using pre-trained domain models (like BERT), which improves the model's comprehension of the steps' semantics. The use of graph neural networks and attention mechanisms in complex system modeling has demonstrated their superior accuracy and robustness in recent years [10], with the potential to enhance the flexibility of cross-scenario operation step generation in distribution networks.

This study suggests a transfer learning-based approach for distribution network generalization optimization and cross-scenario operation step generation based on this analysis. The main concepts are: comparing different deep learning architectures to maximize cross-scenario generation performance; using pre-trained domain models to improve the semantic modeling of operation steps; and developing a domain-adaptive transfer strategy that combines parameter freezing and progressive fine-tuning. The suggested approach performs better than the comparison methods in terms of generation accuracy and cross-scenario generalization ability, according to evaluation using the Caltech 28-Bus Digital Twin dataset. This paper’s primary contributions are as follows: (1) developing a cross-scenario operation step generation framework that combines semantic modeling and transfer learning; (2) creating a domain-adaptive transfer strategy to maximize generalization performance across various scenarios; and (3) verifying the method's efficacy on actual distribution network data in cross-scenario tasks, offering a workable technical solution for the upkeep and operation of smart distribution networks.

This paper is structured as follows: In Section 2, relevant research is reviewed; in Section 3, the method framework is introduced; in Section 4, the experimental design and results analysis are presented; and in Section 5, the paper is concluded, and future work directions are outlined.

2. Related Work

2.1. Transfer Learning Techniques

Transfer learning improves model performance in situations with little data or notable distribution differences by utilizing knowledge from the source domain to support modeling in the target domain. Transfer learning lowers training costs and improves generalization performance in cross-scenario tasks when compared to traditional deep learning techniques that train models from scratch. This is especially important for small-sample, cross-domain applications like operation step generation in distribution networks [11].

Formally, let the target domain be the target task, and the source domain be the source task. Using information from the source domain to enhance learning performance in the target task is the aim of transfer learning.

Common transfer strategies include: (1) Parameter transfer, where parts of the source model’s weights are shared to facilitate knowledge reuse; (2) Feature transfer, which creates a shared representation space between the source and target domains; (3) Fine-tuning, in which only a few parameters are changed to adapt to the target domain and portions of the network are frozen. Some optimization techniques, like Maximum Mean Discrepancy (MMD), minimize the distribution differences between the source and target domains for transfer adaptation:

$$\mathrm{MMD}(D_S, D_T) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(x_i^S) - \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(x_j^T) \right\|_H^2 \tag{1}$$

where $\phi(\cdot)$ denotes the feature mapping in the Hilbert space H, and x_i^S, x_j^T are the samples from the source and target domains, respectively.

In order to improve cross-domain robustness, some studies also use adversarial learning strategies, in which a discriminator minimizes the domain classification loss to achieve feature space alignment [12].

However, the generation of operation steps in distribution networks requires strong domain-specific features and intricate semantic structures. Migration failures or even “negative transfer” phenomena can result from significant variations in equipment status patterns, operation sequences, and topology constraints across various scenarios. In order to overcome these difficulties, this paper suggests a transfer framework that improves the generation quality and generalization capacity of cross-scenario operation step generation by combining source domain pre-training, topology feature guidance, and target domain feature adaptation mechanisms.

2.2. Graph Neural Network Modeling Techniques

A class of deep learning models called Graph Neural Networks (GNNs) was created especially to handle graph-structured data. These models are able to model topological relationships and node features at the same time. GNNs have demonstrated strong structural modeling capabilities in complex systems such as transportation networks, power systems, and bioinformatics [13]. The main idea is to use message passing and feature aggregation mechanisms to integrate information from neighboring nodes into the target node's representation in order to learn embedding vectors suitable for nodes, edges, and even the entire graph.

In the standard GNN framework, the representation $h_i^{(l)}$ of node v_i at the l-th layer is updated by aggregating the features of the neighboring node set $N(v_i)$ as follows:

$$h_i^{(l+1)} = \sigma\left(W^{(l)} \cdot \mathrm{AGG}^{(l)}\left(\{h_j^{(l)}: j \in N(v_i)\}\right) + b^{(l)}\right) \tag{2}$$

Here, $\mathrm{AGG}(\cdot)$ represents the feature aggregation function (e.g., averaging, weighted summation, attention mechanisms, etc.), $W^{(l)}$ is the learnable weight, and σ is a nonlinear activation function.

When it comes to operation step generation for distribution networks, the network topology directly affects scheduling sequences and operational logic. Significant differences in node connectivity, switch layouts, and operating modes during cross-scenario generation may reduce the model's ability to generalize. While traditional sequence models or table-based models struggle to capture such topological dependencies, GNNs are naturally able to model structural relationships like “node-line-switch,” exhibiting advantages in handling topological constraints, state evolution, and path dependencies. For example: (1) Graph Convolutional Networks (GCNs) perform graph convolutions based on Laplacian spectra, making them suitable for global information propagation; (2) Graph Attention Networks (GATs) use attention mechanisms to assign differential weights to neighboring nodes, improving the representation of critical nodes; and (3) GraphSAGE uses sampling and inductive learning strategies, enabling effective training on large-scale graphs [14].

Despite the advantages of GNNs in modeling structural information, there are still problems with cross-scenario operation step generation, such as high model complexity, sensitivity to topological changes, insufficient training stability, and limited generalization ability. Therefore, a crucial approach for enhancing the stability and generalization performance of operation step generation across various distribution network scenarios is to combine GNNs with transfer learning strategies and introduce structural alignment and feature adaptation mechanisms.

2.3. Transformer Architecture and Temporal Modeling

The Transformer architecture, initially introduced by Vaswani et al., replaces the recursive structure of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks in sequence modeling with a design based only on the self-attention mechanism. Natural language processing tasks have greatly improved as a result of this discovery [15]. The Transformer can model global dependencies through parallel computation, improving training efficiency and the capacity to capture long-range dependencies, in contrast to conventional techniques that rely on time-step recurrence.

The self-attention mechanism, which forms the basis of the Transformer, is calculated as follows:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{3}$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors, which is used for normalization. In order to capture long-distance contextual relationships, this mechanism creates dependencies between any position in the sequence. To add explicit positional information while preserving sequence order information, the Transformer employs positional encoding:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{4}$$

where, pos is the position index and, d_{model} is the model's hidden dimension. This allows the model to perceive the sequence order even without a recurrent structure.

When it comes to operation step generation for distribution networks, equipment status changes, operation sequences, and scheduling logic all exhibit significant temporal dependencies. Variations in operational procedures, topology, and event trigger sequences may reduce the model's ability to generalize temporal patterns when transferring across scenarios. The Transformer's multi-head attention mechanism enables it to model temporal dependencies in parallel across multiple subspaces, capturing long-range operational relationships and logical patterns in a variety of scenarios, whereas conventional RNN-based models frequently struggle with vanishing gradients and memory constraints when handling lengthy sequences.

Notably, Transformer still has two major issues with cross-scenario applications: first, it is highly dependent on computational resources and large-scale labeled data; second, it finds it difficult to generalize in scenarios with substantial differences or complicated operational conditions [16]. Therefore, by integrating Transformer with transfer learning techniques and adding domain adaptation, temporal pattern alignment, and parameter fine-tuning mechanisms, the accuracy and stability of operation step generation in various distribution network environments could be enhanced.

2.4. NLP and Semantic Generation Techniques

Intelligent operation and maintenance (O&M) and industrial document automation have made extensive use of Natural Language Processing (NLP), a crucial technology for comprehending and producing structured operation step information. In distribution networks, scheduling instructions and operational procedures are usually the source of the operation steps. These steps include intricate semantic logic in addition to a highly structured expression format. In addition to using standardized language, their composition depends on rigorous adherence to safety regulations and operational procedures. Furthermore, different scenarios have different rules and terminology [17]. Because of this, NLP techniques are essential for the automatic creation and validation of operation steps in various scenarios.

The accuracy of semantic modeling has greatly increased with the creation of pre-trained language models (like BERT, T5, etc.), offering a new technical route for comprehending and producing operation steps. The objective of step generation, which can be thought of as a conditional text generation problem, is to maximize the likelihood of producing the desired step sequence given the input conditions:

$$\hat{Y} = \arg\max_{Y} P(Y \mid X; \theta) \tag{5}$$

Here, X represents the input information (e.g., operational procedure text, equipment status descriptions), Y is the generated sequence of steps, and θ denotes the parameters of the language model.

Masked Language Modeling (MLM) is a technique used by pre-trained models such as BERT, in which the original words are predicted to capture contextual dependencies while a portion of the input sequence is randomly masked. The following is its loss function:

$$L_{\text{MLM}} = -\sum_{i \in M} \log P(x_i \mid X_M; \theta) \tag{6}$$

where M represents the set of masked words, and X_M is the input sequence with the masked words removed.

NLP models must concurrently model procedural texts, equipment statuses, and step logic across various sites and operational conditions in order to generate cross-scenario operation steps for distribution networks. This guarantees that the produced results satisfy the particular scenario's safety and compliance requirements in addition to maintaining the proper global operation sequence. The controllability and consistency of step generation are improved by combining supervised or reinforcement learning with pre-trained language models to facilitate cooperative optimization between syntactic constraints and semantic control.

However, general language models often encounter challenges when directly transferred to the distribution network domain, such as difficulties in recognizing specialized terminology, significant semantic ambiguities, and large differences in textual styles across scenarios. These issues can impair cross-scenario transfer performance [18]. To address these challenges, this paper introduces transfer learning and domain-specific data fine-tuning strategies. Through semantic alignment, feature adaptation, and parameter optimization, the model's semantic understanding and generalization ability for step generation across multiple scenarios are enhanced, providing stable language generation support for intelligent operation and maintenance in distribution networks.

2.5. Pre-training and Parameter Transfer Strategies

To improve deep models' capacity for generalization in cross-scenario and small-sample tasks, pre-training and parameter transfer are crucial tactics. The basic idea is to obtain model parameters with general feature extraction capabilities by conducting supervised or self-supervised pre-training on extensive source domain data. After that, these parameters are moved to the target domain and adjusted to fit the novel situation [19]. This method relieves performance bottlenecks in low-resource scenarios and drastically lessens the target domain's reliance on labeled data.

For example, Causal Language Modeling (CLM) in language models aims to maximize the probability of each word in a sequence conditioned on its preceding context. The definition of the loss function is:

$$L_{\text{CLM}} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta) \tag{7}$$

where x_t represents the t-th word in the sequence, $x_{<t}$ denotes all the words preceding this position, and θ refers to the model parameters. The model can acquire useful contextual dependencies thanks to this pre-training technique, which offers a strong feature base for cross-scenario transfer.

During the parameter transfer phase, suppose the model parameters θ_s are obtained from training on source domain data D_s. The goal is to optimize these parameters θ_t on the target domain data D_t. The common strategies for parameter transfer include:(1) Full parameter transfer, where all parameters are updated in the target domain and serve as the initial value; (2) partial freezing fine-tuning, where only the higher layers pertinent to the task are fine-tuned and the parameters of the lower layers responsible for general feature extraction remain unchanged; and (3) selective parameter sharing, where parameters are shared between specific layers or structural adjustments are introduced to balance knowledge retention and task adaptation.

Pre-training and parameter transfer techniques have been used in conjunction with Transformer and GNN architectures in recent years, showing outstanding transferability and generalization performance in temporal modeling and structural modeling tasks [20]. When training models directly, variations in equipment types, topology structures, and procedural language styles among sites may result in lower generalization performance in cross-scenario operation step generation for distribution networks. By combining parameter transfer with pre-trained general representations, knowledge from the source domain can be retained while rapidly adjusting to the target scenario. This increases the precision and resilience of step generation and creates the technical foundation for additional optimization of generalization.

3. Methodology

3.1. Overall Methodology Framework

In order to solve the problem of inadequate generalization, this paper suggests a transfer learning-based approach for cross-scenario operation step generation and generalization optimization in distribution networks. performance in various locations and operating environments using current techniques. The four main components of the proposed approach are: deep architecture with topology and temporal semantics; cross-scenario transfer learning strategy; state-action sequence modeling and domain pre-training; and model training and generalization optimization process. These four components work together to produce a complete cross-scenario step generation solution, from data representation to model optimization. The overall framework of the method is shown in Figure 1: (1) Modeling state-action sequences and pre-training domains: Based on distribution network operation data (including device status time series, network topology, and metadata), a temporal-semantic representation of the operation steps is constructed. Transformer encoding, step logic consistency constraints, and the discretization of states and actions yield high-quality feature representations that represent the operational logic. In order to capture the distinct distribution network scheduling and execution patterns, the model is subsequently pre-trained using domain data. (2) Learning strategy for cross-scenario transfer: A transfer mechanism that combines progressive fine-tuning and parameter freezing is used to transfer the state-action mapping obtained from the source domain to the target scenario. Strategies like distribution alignment and cross-scenario domain adversarial learning are used to minimize feature distribution differences across different sites and operating conditions, thereby improving the model's adaptability to new scenarios. (3) Combining temporal semantics and topology in deep architecture: A Transformer is used to capture the temporal dependencies of operation steps, while a Graph Neural Network (GNN) is used to model the topology of the distribution network. A Cross-Attention mechanism is introduced to facilitate the interaction between topological information and temporal semantics, ensuring that the model maintains stable generation performance across a range of network structures. (4) The training and generalization model optimization process: After pre-training on the source domain, the target domain undergoes staged fine-tuning. Techniques like layer freezing, early stopping, and regularization are employed to prevent overfitting. The learning rate and weight update strategy are dynamically adjusted during training in line with generalization objectives to guarantee that the model generates operation steps that follow scheduling logic and safety regulations under cross-scenario conditions.

Through the cooperative operation of these modules, the proposed approach can effectively transfer the operational knowledge from the source domain and produce distribution network operation steps with high generalization ability, even in situations with little data and significant scenario differences.

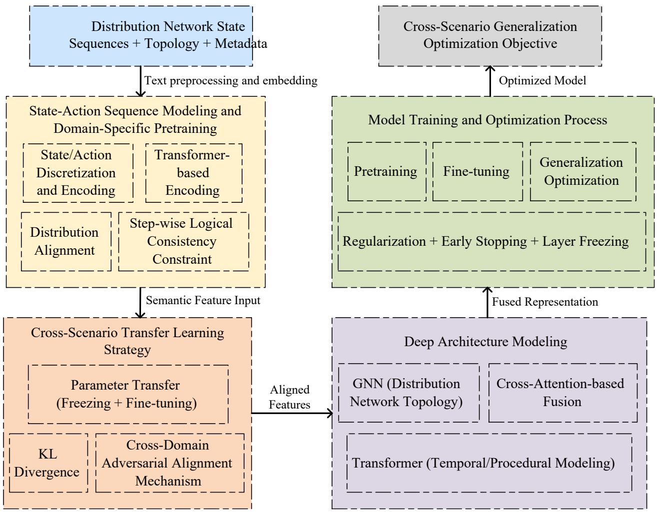

Figure 1. Overall Framework

3.2. Operation Step: Semantic Modeling and Domain Pre-training

In cross-scenario generation tasks, feature extraction and sequence comprehension depend on the semantic modeling of operation steps. The objective is to convert the state-action sequences in the distribution network operation data into stable, transferable, high-dimensional semantic representations that provide the necessary information for subsequent transfer learning and generalization optimization. This paper proposes a semantic modeling framework for cross-scenario tasks in distribution networks that combines temporal modeling and domain-adaptive pre-training.

(1) Data Preparation and Sequence Construction

Using real-time data collected from the distribution network operation environment, including execution action logs, device status time series, and network topology information, state-action pairs are generated in chronological order. Data preprocessing techniques include uniform encoding, noise filtering, sequence segmentation, and action category normalization to reduce redundancy and distribution bias in the input features. For low-frequency operation instructions, a sub-word/action unit decomposition mechanism is applied to alleviate the sparsity problem.

(2) Temporal Semantic Representation

Given a state-action sequence $X = \{x_1, x_2, \ldots, x_n\}$, it is embedded into a vector space and input into the Transformer encoder to compute the contextual dependency representations:

$$h_i = \text{Transformer}(x_i, X_i; \theta_B) \tag{8}$$

where h_i represents the contextual semantic vector of time step x_i, and θ_B refers to the parameters of the pre-trained model. Both the local action logic and the global temporal dependencies of the operation steps are captured in this representation.

(3) Domain-Adaptive Constraints

To enhance cross-scenario adaptability, a distribution alignment strategy is introduced to ensure that the source domain sequence distribution $P_s(h)$ and the target domain sequence distribution $P_t(h)$ are as consistent as possible in the embedding space. The Kullback–Leibler divergence constraint is used:

$$L_{\text{DA}} = D_{\text{KL}}(P_s(h) \parallel P_t(h)) \tag{9}$$

By encouraging the pre-trained model to align the sequence features between the source and target domains, this loss function lessens the drop in migration performance caused by scenario differences.

(4) Logical Consistency Constraints

To maintain structural consistency between semantically similar operation sequences, an Euclidean distance constraint is applied to logically similar sequence pairs $(i, j) \in S$:

$$L_{\text{SC}} = \sum_{(i,j)\in S} \|h_i - h_j\|_2^2 \tag{10}$$

where S represents the set of logically similar sequence pairs. This restriction guarantees that the model represents the semantics of the steps while taking logical structure into account.

(5) Pre-training Objective Function

Domain adaptation (DA), logical consistency (SC), step relation prediction (NSP), and masked sequence modeling (MLM) are the four components that are considered in the pre-training optimization goal:

$$L_{\text{total}} = \lambda_1 L_{\text{MLM}} + \lambda_2 L_{\text{NSP}} + \lambda_3 L_{\text{SC}} + \lambda_4 L_{\text{DA}} \tag{11}$$

where λ_i represents the weights for each loss term.

This design produces high-quality input representations for transfer learning by aligning feature distributions across different scenarios and effectively capturing the temporal and semantic features of operation steps in distribution networks.

3.3. Transfer Learning Strategy Design

This paper develops a transfer learning strategy that combines parameter transfer, domain adaptation, and adversarial regularization to improve the generalization ability of operation step generation across various scenarios in distribution networks, thereby mitigating the impact of distribution differences between the source and target domains on model performance.

(1) Parameter Transfer and Layer-wise Fine-Tuning

The model is initially pre-trained on the source domain $D_s = \{(x_s^i, y_s^i)\}_{i=1}^{N_s}$, obtaining the parameters θ_s. The model is then transferred to the target domain $D_t = \{(x_t^j, y_t^j)\}_{j=1}^{N_t}$ for fine-tuning. The layers are then gradually unfrozen from top to bottom, enabling the model to progressively adjust to the unique operational features of the target domain. This staged optimization guarantees smooth transfer and lowers the possibility of catastrophic forgetting brought on by sudden parameter changes.

(2) Domain-Adaptive Distribution Alignment

To minimize the distribution discrepancy between the source and target domain feature distributions, let the sequence embeddings of both domains be represented as H_s and H_t, with distributions $P_s(H_s)$ and $P_t(H_t)$, respectively. The definition of the distribution alignment loss is:

$$L_{\text{align}} = D_{\text{KL}}(P_s(H_s) \parallel P_t(H_t)) \tag{12}$$

This constraint increases the alignment of the source and target domains in the feature space by lowering the Kullback-Leibler divergence, which improves the target domain's ability to express semantics.

(3) Domain-Adversarial Feature Extraction

An additional domain adversarial mechanism is introduced by constructing a domain classifier to distinguish the source domain of each sample. The domain adversarial loss is defined as:

$$L_{\text{adv}} = -\frac{1}{N}\sum_{i=1}^{N} [z_i \log D(H_i) + (1 - z_i)\log(1 - D(H_i))] \tag{13}$$

where $z_i \in \{0,1\}$ represents the domain label and H_i is the sample feature representation. The model improves cross-scenario adaptability by learning domain-invariant features through adversarial optimization between the feature extractor and domain classifier.

(4) Overall Transfer Learning Objective Function

The transfer learning phase's overall optimization objective considers both cross-domain adaptability and task generation accuracy:

$$L_{\text{TL}} = L_{\text{task}} + \alpha L_{\text{align}} + \beta L_{\text{adv}} \tag{14}$$

where L_{task} is the task loss for operation step generation in the target domain, and α and β are the weight coefficients.

The model effectively adapts to the target domain's operational patterns while preserving the operational patterns acquired in the source domain by employing the previously mentioned strategies. qualities, preserving excellent generalization performance and generation accuracy across scenarios. The study's suggested transfer learning approach is depicted in Figure 2.

3.4. Deep Learning Architecture for Generalization

This paper proposes a deep learning architecture that combines topology feature representation with operational state sequence modeling to address the problem of inadequate generalization in operation step generation across scenarios. The architecture enables joint modeling of topology structure and state information by combining the strengths of Graph Neural Networks (GNN) in capturing structural features with the benefits of Transformer in modeling global temporal dependencies. To improve cross-domain adaptability, transfer learning mechanisms are also included.

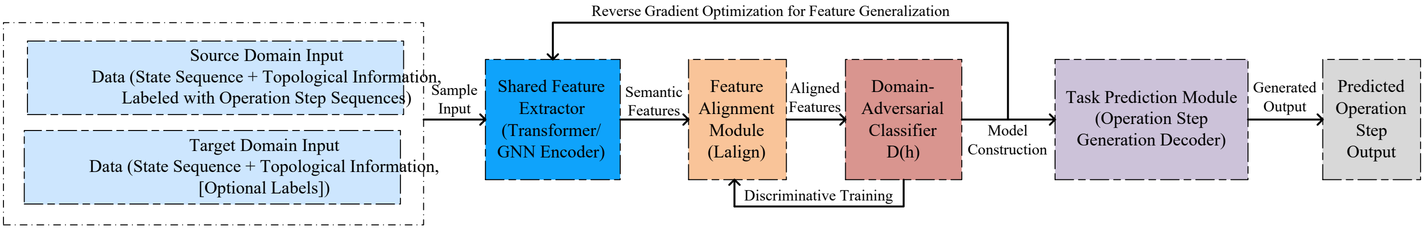

Figure 2. Transfer Learning Strategy Framework

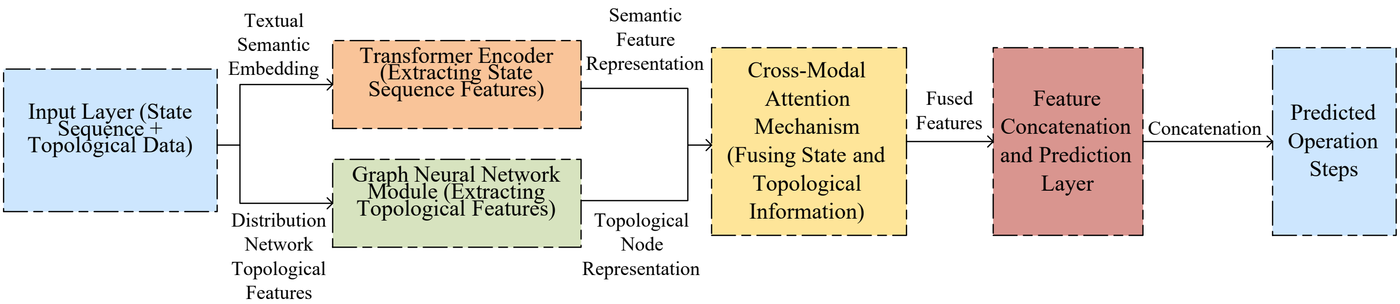

Figure 3. Deep Learning Architecture for Generalization Optimization

(1) Input Layer and Feature Extraction

The input consists of operational state sequence features $H_s \in \mathbb{R}^{n \times d}$ and distribution network topology node features $H_g \in \mathbb{R}^{m \times d}$, where n represents the length of the state sequence, m is the number of nodes, and d is the embedding dimension. Operational state features are extracted using a Transformer-based temporal encoder, capturing long-range dependencies and sequential logic. Topology node features are extracted using a Graph Convolutional Network (GCN), with the update formula as follows:

$$H_g^{(l+1)} = \sigma(\tilde{A} H_g^{(l)} W^{(l)}) \tag{15}$$

where $\tilde{A}$ is the normalized adjacency matrix, $W^{(l)}$ is the weight for the l-th layer, and $\sigma(\cdot)$ is the nonlinear activation function.

(2) Fusion Layer and Cross-modal Attention

To model the relationships between state and topology features, a cross-modal attention mechanism is introduced. Let the state features be the query vector $Q = H_s W_q$, and the topology features as key-value pairs $K = H_g W_k, V = H_g W_v$. The attention calculation is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{16}$$

where W_q, W_k, W_v are trainable parameters, and d_k is the scaling factor. This mechanism dynamically captures the dependency between the operational state and the topology structure, enhancing the model's generalization ability across different distribution network structures.

(3) Prediction Layer and Regularization

To create the anticipated operation step sequence, which includes device actions and execution order, the fused features go through a fully connected layer and normalization. The prediction function is:

$$\hat{y} = f_\theta(H_s, H_g) = \sigma\big(W_o[H_s' \oplus H_g'] + b_o\big) \tag{17}$$

where H_s' and H_g' are the fused state and topology representations, $\oplus$ denotes feature concatenation, and W_o, b_o are the output layer parameters. Dropout regularization and residual connections are introduced during training to prevent overfitting and ensure gradient stability.

(4) Architecture Advantages

This architecture has the following characteristics: ① It enhances the accuracy and completeness of step generation by jointly modeling topology features and operational state. ② The model can adapt to different network structures because the cross-modal attention mechanism improves information interaction. ③ The performance and stability of cross-scenario generalization are improved by combining regularization and transfer learning techniques. Figure 3 shows the overall structure of this architecture.

3.5. Model Training and Optimization Protocol

This paper develops a staged training and optimization strategy to guarantee that the cross-scenario operation step generation model efficiently applies knowledge from the source domain and sustains stable generalization performance after being transferred to the target domain. This strategy includes three core phases: source domain pre-training, target domain transfer fine-tuning, and generalization optimization. The overall process is shown in Figure 4.

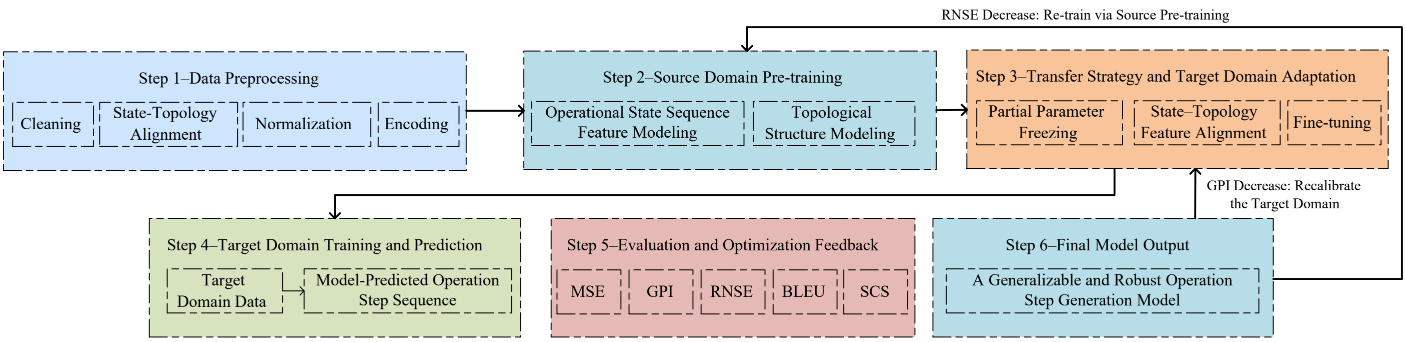

Figure 4. Model Training and Optimization Process

(1) Source Domain Pre-training

In the source domain, multimodal feature modeling is performed using operational state sequences and network topology information. The training objective combines the operational state modeling loss L_{state} and topology modeling loss L_{topo}:

$$L_{\text{pretrain}} = L_{\text{state}} + \gamma L_{\text{topo}} \tag{18}$$

where γ is a weight coefficient used to balance the contributions of the two modal features. The optimization process utilizes the AdamW optimizer, along with a learning rate warm-up and decay strategy, to accelerate model convergence and stabilize feature learning.

(2) Target Domain Transfer Fine-tuning

During the transfer phase, the lower layers responsible for general feature extraction are fixed, and only the higher layers are updated to retain shared features across scenarios. In this phase, the distribution alignment loss L_{align} and domain adversarial loss L_{adv} are introduced to reduce the feature distribution differences between the source and target domains. The optimization objective for the fine-tuning phase is:

$$L_{\text{finetune}} = L_{\text{task}} + \alpha L_{\text{align}} + \beta L_{\text{adv}} \tag{19}$$

where L_{task} represents the task loss for operation step generation in the target domain, and α, β are balancing parameters.

(3) Generalization Optimization

Following fine-tuning, an early stopping mechanism is used in conjunction with a layer-wise unfreezing strategy to gradually release additional training parameters. Overfitting is avoided by introducing an L2 regularization term:

$$L_{\text{generalize}} = L_{\text{finetune}} + \lambda \|W\|_2^2 \tag{20}$$

where W represents the model weights, and λ is the regularization coefficient. Dropout and residual connections are incorporated during training to enhance robustness under varying operational conditions.

Lastly, mini-batch stochastic gradient descent (SGD) is used to update the model, and the validation set is used to track generalization performance in real time. The learning rate and regularization strength are dynamically adjusted. Following training, the model can reliably generate operation steps that follow scheduling logic and safety constraints across a range of locations and operating conditions.

4. Experimental Setup and Results

4.1. Overview of Experiments

4.1.1. Experiment Design

This study has created five related experimental tasks, each concentrating on a distinct evaluation aspect, to validate the efficacy and applicability of the suggested transfer learning-based approach for cross-scenario operation step generation and generalization optimization in distribution networks. Model convergence characteristics, target domain performance, adaptability under various network architectures, quality analysis of generated steps, and the variation of generalization ability under various data scales are all covered in these tasks. Based on variables like training efficiency, prediction accuracy, operational logic consistency, and cross-scenario generalization ability, the experiment's design offers a systematic assessment of the suggested method and comparison strategies.

The experiments concentrate on evaluating the model's practical applicability under actual operating conditions in addition to contrasting transfer learning strategies with conventional training methods (such as no transfer and direct fine-tuning). This includes assessing how well the generated operation steps follow operational guidelines, how logical the execution order is, and how accurate they are. In order to verify the method's stability and robustness in intricate and dynamic distribution network environments, some experiments also incorporate network topology encoding, operational state features, and cross-domain modeling mechanisms. Table 1 provides a summary of the particular experiment design and goals.

A comparison of the overall performance differences between transfer and non-transfer models is the first step in the progressive experiment design. Next, various transfer strategies are assessed. The adaptability, generalizability, and operational stability of the suggested approach are then thoroughly validated through multi-architecture experiments, real-world operation step generation analysis, and data scale variation tests. The results and analysis that follow are clearly supported structurally by this methodical experimental framework.

Table 1. Overview of Different Experimental Tasks and Objectives

Experiment No.	Experiment Title	Experiment Objective Description
Experiment 1	Convergence Verification of Transfer Learning Strategy	Compare training convergence speed and stability of different transfer strategies in the target domain, validating the convergence performance advantage of transfer structures.
Experiment 2	Performance Comparison in Target Domain	Evaluate the operation step generation accuracy and generalization performance of the Proposed TL and comparison strategies on the target domain test set.
Experiment 3	Evaluation of Different Deep Learning Models	Assess the adaptability and robustness of the proposed strategy across different neural network architectures, exploring how architectural differences impact transfer performance.
Experiment 4	Cross-Scenario Operation Step Generation and Semantic Comparison	Analyze the order rationality, logical correctness, and task adaptation of the models in generating operation steps in real-world tasks.
Experiment 5	Generalization Performance Analysis Across Different Target Domain Data Proportions	Investigate the performance trend of the transfer learning model under varying target domain data sizes, assessing its applicability in small-sample scenarios.

4.1.2. Dataset Description

The experiments are validated using the Caltech 28-Bus Digital Twin Distribution Network dataset [21]. This dataset is based on monitoring data of a real distribution system in California and includes time-series data on operational states, network topology, and device metadata. It provides a comprehensive depiction of the system's dynamic operational and structural features. The experiment design divides the data into two subsets: the Source Domain and the Target Domain, which are utilized for the pre-training phase of transfer learning and the cross-scenario adaptability validation phase, respectively. The specific data components are described as follows:

(1) Time-Series Data by State

During operation, this data documents the status changes of various devices, including circuit breakers and switches. The data includes timestamps and device status identifiers, where "NC" (Normally Closed) and "NO" (Normally Open) indicate the respective states of devices. The sampling interval, which covers a variety of operating conditions, is one second. Temporal feature extraction and device behavior modeling can both benefit from this data. This information aids in capturing the dependencies and order of state changes in operation step generation, which can subsequently be utilized to deduce a sensible order of operational steps for various operating conditions.

(2) Network Topology Information

The distribution network's switch layout, line parameters, and node connectivity are all described by the topology data, which is kept in a structured file format. Lines are identified by their starting and ending nodes, and this information is used directly to construct graph-based features, facilitating training that incorporates topological constraints. By identifying physical connections between nodes and the limitations they place on the order of operations, this data enables the model to make sure that the steps that are produced adhere to safety regulations.

(3) Device Metadata

In order to create correlations between device identifiers and their actual locations, this includes channel numbers, node mappings, and device registration data. This metadata supports the fusion of state data and network topology. During operation step generation, metadata lowers the possibility of logical or safety conflicts by ensuring that the operation objects match real devices.

(4) Data Preprocessing and Feature Encoding

The time-series data is preprocessed to deal with missing values, and gaps in the data are filled using forward and backward interpolation. Device states are encoded in binary form (0 for NC, 1 for NO) and combined with topology features to form the training input. To lessen the effect of dimensional differences, all features are standardized.

(5) Source Domain and Target Domain Division

Source Domain: For the pre-training stage, several operational state datasets from different scenarios are chosen. Without being constrained by particular scenarios, the model learns general distribution network operational patterns, such as operational step expression patterns, temporal logic reasoning, and topology comprehension.

Target Domain: To verify the model's cross-scenario adaptability and generalization performance, operational state data from a particular scenario that differs greatly from the source domain in terms of distribution is chosen. An 80:20 split is made between training and validation sets of the target domain data. The validation set does not take part in the final test evaluation; instead, it is used for performance monitoring and parameter adjustment during training. The goal of the target domain training is to produce steps that satisfy safety regulations and operational guidelines while rapidly adapting the model to a new scenario under constrained sample conditions.

Table 2 summarizes the main components, file types, feature descriptions, and their roles in the experiment.

Table 2. Dataset Composition and Feature Statistics

Data Type	File Type	Feature Description	Time Span	Purpose in Experiment
State Time-Series	CSV	Circuit breaker and switch statuses (0/1), 1-second interval	Several hours (subset)	Temporal feature extraction for operation step generation
Network Topology	JSON	Node connections, line parameters, device topology	Static structure	Graph feature extraction, topology constraint modeling
Device Metadata	JSON	Device registration, node mapping, channel number	Static configuration	Data parsing and operation object localization
Target Domain State Data	CSV	State sequence for specific scenario, used as target domain training data	Several hours (subset)	Target domain training and validation for transfer learning
Source Domain State Data	CSV	State sequence for other scenarios, used as source domain pre-training data	Several hours (subset)	Source domain pre-training model construction

Following the aforementioned processing, the experimental data reflect the network's structural constraints and maintain the dynamic features of operational states, offering strong data support for the transfer learning strategy and cross-scenario generalization optimization.

4.1.3. Data Preprocessing and Parameter Setup

The performance of the suggested method in the cross-scenario operation step generation task has been thoroughly evaluated by this study using a systematic preprocessing pipeline, reasonable training configurations, and multidimensional evaluation metrics to guarantee that the input data satisfies the model training requirements in terms of dimensional consistency, temporal continuity, and structural adaptation.

(1) Preprocessing Methods

In order to preserve temporal continuity, the first step for the operational state time-series data is to fill in any missing state values by combining forward and backward filling. During the encoding phase, device statuses "NC" (closed) and "NO" (open) are mapped to binary values 0 and 1, respectively. Following this, multiple device state files are aligned using timestamps, and a unified time-series feature matrix is constructed. Topological features are extracted from the network structure files by calculating node degrees and other graph structural attributes based on line connection information (represented by fbus and tbus). These features are then used as inputs for graph feature embeddings. In order to improve training stability and convergence speed, all features are finally standardized (mean = 0, standard deviation = 1).

(2) Training Configuration

MLP, LSTM, Transformer, GNN, and the suggested Transfer Learning-based model (Proposed TL) are among the models that are compared in the experiment. To ensure comparability, all models use the same primary training parameters, which are as follows:

- Optimizer: Adam
- Initial learning rate: 0.001
- Batch size: 64
- Epochs: 50
- Loss function: Mean Squared Error (MSELoss)
- Data split ratio: 70% for the source domain, 30% for the target domain

To evaluate the effectiveness of the transfer learning strategy, three strategies are compared in the experiment:

① No Transfer: No information from the source domain is utilized; only data from the target domain is used for training.

② Fine-tuning: All parameters are moved to the target domain and adjusted using a lower learning rate after pre-training on the source domain.

③ Proposed TL: To enhance cross-scenario generalization performance and stability, this study suggests a transfer learning approach that combines source domain pre-training, partial parameter freezing, and target domain feature adaptation.

These strategies will be systematically compared in Sections 4.2 and 4.3 based on convergence curves and performance metrics.

(3) Evaluation Metrics

To comprehensively evaluate model performance, the following four categories of metrics are used: prediction error (MSE, MAE), generalization performance (GPI, RNSE), and generation quality (BLEU, SCS).

① Mean Squared Error (MSE): Measures the squared error between predicted values and true values. It is defined as:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{21}$$

② Mean Absolute Error (MAE): Represents the average of the absolute prediction errors:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{22}$$

③ Generalization Performance Index (GPI): This index measures the reduction in error achieved by transfer learning compared to the no-transfer model. It is defined as:

$$\mathrm{GPI} = \frac{\mathrm{MSE}_{\text{No Transfer}} - \mathrm{MSE}_{\text{Proposed TL}}}{\mathrm{MSE}_{\text{No Transfer}}} \tag{23}$$

④ The improvement in generalization performance in comparison to a baseline model is measured by the Relative Normalized Squared Error (RNSE). It is computed as:

$$\mathrm{RNSE} = 1 - \frac{\mathrm{MSE}_{\text{Model}}}{\mathrm{MSE}_{\text{Baseline}}} \tag{24}$$

Here, the Baseline refers to results obtained using simple prediction techniques, such as the most recent mean or a moving window. With zero denoting no improvement and negative values denoting performance degradation or negative transfer, a higher RNSE denotes improved performance in comparison to the baseline.

⑤ Generation Quality Metrics (BLEU and SCS): These metrics assess the action order, object consistency, and semantic alignment of the generated operation step sequences.

BLEU (Bilingual Evaluation Understudy): Measures the n-gram matching between the generated sequence and reference steps:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{25}$$

where p_n is the precision of n-grams, w_n is the weight coefficient, and BP is the brevity penalty factor. A higher BLEU score indicates better surface-level match between the generated sequence and reference steps.

The Semantic Consistency Score (SCS) evaluates how well the generated sequence and reference steps match up in the semantic embedding space:

$$SCS = \frac{\mathrm{v}_{gen} \cdot \mathrm{v}_{ref}}{\|\mathrm{v}_{gen}\|\|\mathrm{v}_{ref}\|} \tag{26}$$

where v_{gen} and v_{ref} are the semantic vectors of the generated and reference sequences, respectively. A score closer to 1 indicates higher semantic consistency.

The performance comparison under various transfer strategies and cross-scenario conditions is guaranteed to be consistent and scientifically valid thanks to the aforementioned preprocessing, consistent parameter settings, and thorough evaluation system.

4.2. Experiment 1: Convergence Validation of Transfer Learning Strategy

This study compares the proposed transfer learning strategy with two alternative strategies, No Transfer and Fine-tuning, in order to verify its convergence performance in the cross-scenario operation step generation task for the distribution network.

Eighty percent of the target domain dataset is used for training and twenty percent is used for validation in the experiment. The same network architecture, optimizer (Adam), learning rate (0.001), batch size (64), and 50 epochs are used for training all three strategies. The transfer mechanism is the sole distinction between the strategies:

(1) No Transfer: Training directly from scratch on the target domain training set without using source domain information.

(2) Fine-tuning: After pre-training the entire model on the source domain, transfer all parameters to the target domain for fine-tuning at a low learning rate.

(3) Proposed TL: The target domain training set will integrate feature adaptation mechanisms, freeze particular structural parameters, and expand upon source domain pre-training in order to enhance cross-scenario generalization.

The model's fitting error and convergence speed in the target domain operation step generation task are assessed by calculating the validation loss (Validation Loss) at the conclusion of each training epoch. To lessen the effect of random fluctuations, the outcomes from three separate runs are averaged.

At the end of each training epoch, the validation loss (Validation Loss) is computed to evaluate the model's fitting error and convergence speed in the target domain operation step generation task. To lessen the effect of random fluctuations, the outcomes from three separate runs are averaged.

Figure 5 shows how the validation loss changes over the course of training. The validation loss values, which represent the model's fitting error on the validation set, are displayed on the y-axis, while the x-axis displays the training epochs.

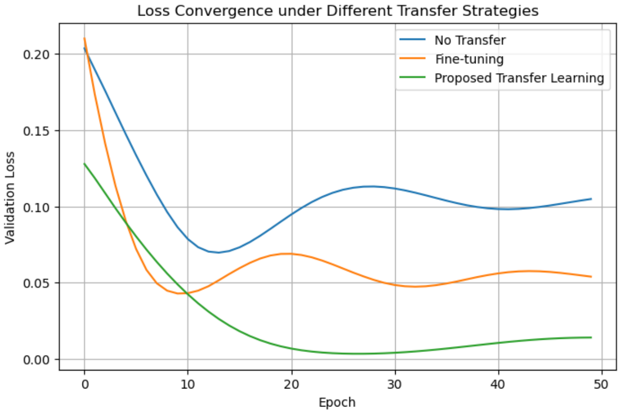

Figure 5. Loss Convergence under Different Transfer Strategies

Figure 5 illustrates how the Proposed TL converges considerably more quickly than the other strategies and shows a much lower loss early in training. It maintains a very low loss level and reaches convergence after about 12 epochs. In contrast, Fine-tuning takes about 25 epochs to converge, with continued fluctuations in the later stages. When compared to the other two strategies, No Transfer converges the slowest and has a much larger final loss.

Table 3 presents evaluation metrics such as initial loss, optimal convergence loss, and convergence speed improvement for a quantitative comparison of the convergence efficiency and stability of the three strategies.

Table 3. Measures of Quantitative Assessment

Strategy	Initial Loss (Epoch=0)	Converged Loss (Optimal Epoch)	Convergence Epoch	Convergence Speed Improvement (%)
No Transfer	0.205	0.105	45	—
Fine-tuning	0.210	0.052	25	44%
Proposed TL	0.128	0.010	12	73%

As demonstrated in Figure 5 and Table 3, the recommended transfer learning strategy clearly outperforms the other methods in terms of convergence speed and final performance. This suggests that this approach can swiftly and efficiently learn and produce operation step sequences that match the operational logic of the target scenario, even with limited target domain data. This provides a strong basis for further generalization performance optimization.

4.3. Experiment 2: Performance Evaluation of Transfer Strategies in the Target Domain

To fully assess the effectiveness of various transfer strategies in the context of cross-scenario operation step generation for distribution networks, this experiment was conducted using the same target domain data split as in Section 4.2. Every model was trained using the same network architecture, optimization parameters, and training epochs to guarantee an impartial and controlled comparison. Performance was assessed using a different test set that was not used for training or post-training validation.

Two categories of evaluation metrics were employed: (1) Generation Accuracy: Determined by utilizing Mean Squared Error (MSE) and Mean Absolute Error (MAE) to compare the reference sequences and the expected operation step sequences in terms of temporal logic and execution states. (2) Generalization Capability: Determined by measuring the relative increase in prediction accuracy brought about by transfer learning strategies over the baseline using the Generalization Performance Index (GPI), as described in Section 4.1.3.

Figure 6 compares the MSE, MAE (left subfigure), and GPI (right subfigure) of the three methods: No Transfer, Fine-tuning, and the Proposed Transfer Learning (Proposed TL). The horizontal axis in both charts indicates the transfer strategy, while the vertical axes correspond to prediction errors and generalization gains, respectively.

In terms of prediction accuracy (left), the Proposed TL strategy outperformed the others significantly, achieving an MSE of 0.06 and an MAE of 0.245, which are markedly lower than those of No Transfer (MSE = 0.15, MAE = 0.39) and Fine-tuning (MSE = 0.11, MAE = 0.33). These findings show that the temporal dependencies and structural constraints inherent in distribution network operations can be efficiently captured by combining source domain pretraining with topology-aware transfer mechanisms, resulting in more precise and logically consistent operation step generation.

Regarding generalization capability (right), the Proposed TL approach achieved the highest GPI value (approximately 0.47), significantly outperforming Fine-tuning (≈ 0.23), while the No Transfer baseline, by definition, yields a GPI of 0. This illustrates the proposed TL strategy's superior cross-scenario transferability. The amount and distribution of target domain data determine how effective fine-tuning is, even though it produces some performance gains. Interestingly, fine-tuning is vulnerable to catastrophic forgetting when domain shifts are significant, leading to sequences that depart from target domain operational norms.

4.4. Experiment 3: Deep Learning Model Performance Assessment

This experiment compares five model types in the task of cross-scenario operation sequence generation for distribution networks in order to assess the adaptability of the suggested method across various deep learning architectures. The Transformer with attention mechanisms, Graph Neural Network (GNN), Long Short-Term Memory network (LSTM), Multi-Layer Perceptron (MLP), and the suggested Transfer Learning-based model (Proposed TL) are some of the models.

The same target domain dataset is used for both training and evaluating each model. With an 80/20 split between training and validation sets, the data partitioning method is similar to that of Experiment 1 (Section 4.2). Performance trends and convergence curves are tracked using the validation subset. The Proposed TL model utilizes pretrained weights from the source domain and integrates a topology-guided transfer mechanism. The initialization of every other model is random. Every model is trained with the same batch size, epoch count, learning rate, and input features to guarantee equity.

The evaluation is conducted from three angles: (1) Generation Accuracy (measured by the Root Mean Square Error (RMSE) and sequence-level matching accuracy between the generated and reference operation sequences); (2) Cross-scenario Generalization (measured by the Relative Normalized Squared Error (RNSE), which shows the performance improvement over a baseline model); and (3) Training Efficiency (measured by the total training time on the target domain task).

Figure 7 displays each model's generalization results and convergence behavior on the validation set. The RMSE convergence curves over training epochs are shown in the left panel; closer alignment with the reference sequences is indicated by lower values. Each model's RNSE score, which shows how well it can generalize under domain shifts, is shown in the right panel.

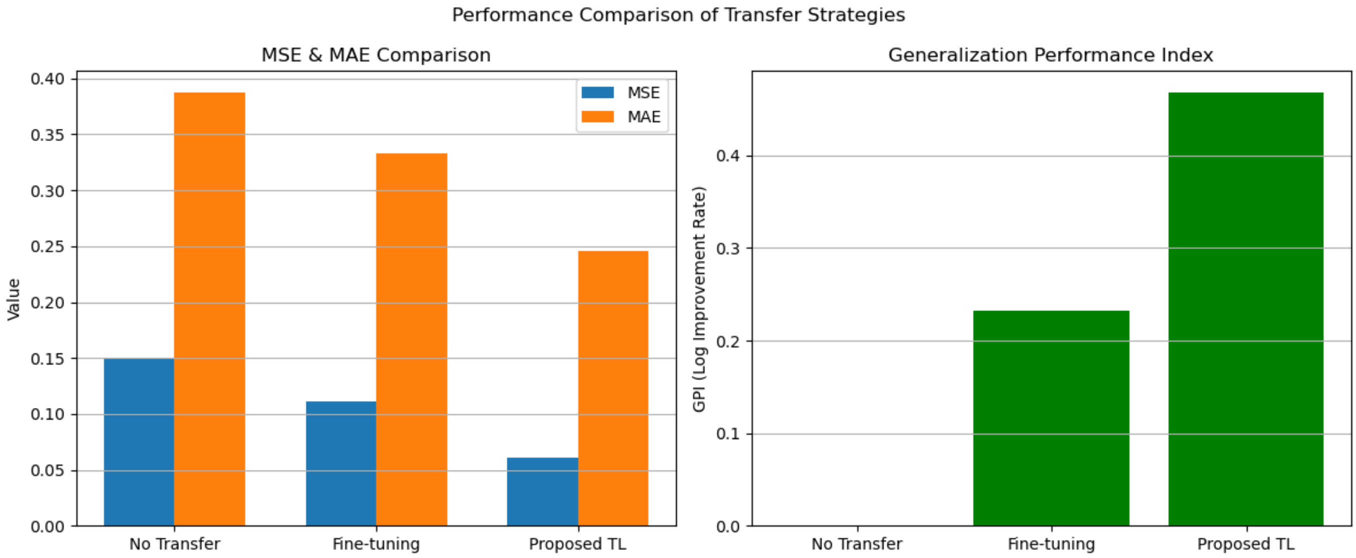

Figure 6. Performance comparison of transfer strategies on the target domain test set

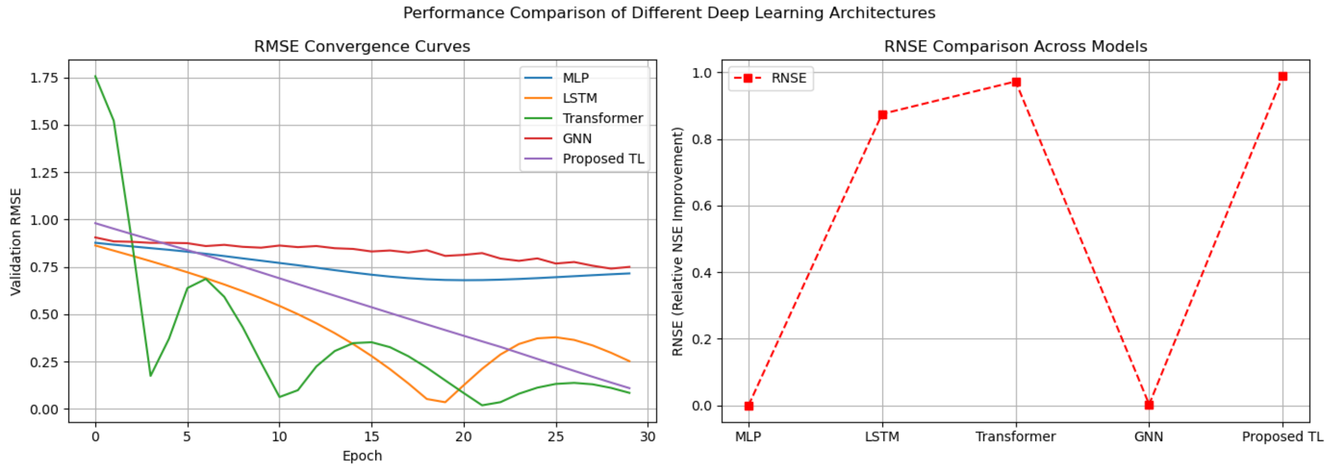

Figure 7. Cross-Scenario Operation Sequence Generation Performance Comparison of Various Deep Learning Models

Both MLP and GNN have comparatively low generation accuracy and convergence rates, as seen in Figure 7, with RNSE values near zero, indicating limited generalization ability. With lower RMSE and RNSE scores of 0.85 and 0.93, respectively, LSTM and Transformer models significantly outperform other models in modeling temporal dependencies and structural constraints. When topology-guided transfer and source-domain pretraining are applied, the proposed TL model has the lowest final RMSE and the best convergence stability. Its validation loss steadily decreases over training epochs, and its RNSE approaches 0.96, indicating strong generalization under unknown scenario distributions.

Table 4 summarizes the comprehensive performance of all five models, including generation accuracy, RNSE, and total training time. The findings demonstrate that the significant improvements in accuracy and generalization outweigh the suggested TL model's marginally higher training overhead.

Table 4. Deep Learning Models' Comparative Results in the Target Domain

Model	Accuracy	RNSE	Training Time (s)
MLP	0.74	0.00	32
LSTM	0.82	0.85	45
Transformer	0.84	0.93	58
GNN	0.78	0.05	52
Proposed TL	0.89	0.96	60

In summary, the proposed TL model consistently performs better than its competitors in terms of generation fidelity, convergence robustness, and domain generalization across a range of neural architectures. These results confirm that the approach is appropriate and efficient for handling operations sequence generation tasks in heterogeneous distribution network scenarios.

4.5. Experiment 4: Cross-Scenario Operation Sequence Generation and Semantic Analysis

By comparing the quality of generated results with outputs from baseline models, this experiment thoroughly evaluates the efficacy of the suggested method in cross-scenario distribution network operation sequences. As an essential component of distribution network scheduling, operation sequences must exhibit logical consistency, semantic completeness, and appropriate execution order. Therefore, in order to verify the method's practical applicability under cross-scenario conditions, this study also includes representative output samples and expert validation in addition to quantitative metrics.

(1) Designing Experiments

In order to provide a trustworthy testbed for generalization, the evaluation is carried out using historical operation data from the target domain, which is different from the source domain in terms of equipment state distributions, network topology, and operational conventions. An 8:2 ratio is used to separate the dataset into training and testing sets. All models are trained under identical configurations. A topology-guided transfer mechanism and pretrained weights from the source domain are used to initialize the suggested TL model. In contrast, baseline models (MLP, LSTM, Transformer) are trained from scratch.

Following training, each model produces fifty operation sequences on the test set. Two representative cases with a range of circumstances and levels of complexity are chosen from these for additional examination. Expert review is conducted along three dimensions: correctness of execution order, semantic completeness, and safety compliance.

Furthermore, the linguistic and semantic alignment between generated and reference sequences is quantitatively evaluated using BLEU and Semantic Consistency Score (SCS) metrics (see Section 4.1.3).

(2) Sample Case Comparison and Analysis

A multidimensional comparison of produced outputs in two common scenarios is shown in Table 5. The analysis shows: 1) MLP frequently omits critical steps and violates safety protocols. 2) LSTM shows gains in semantic richness, but it has trouble keeping the right action sequence. 3) The transformer produces more fluent expressions but occasionally lacks essential safety annotations. 4) By consistently upholding exact sequencing, semantic integrity, and adherence to operational norms, the suggested TL closely resembles the ground truth.

(3) Quantitative Evaluation of Generation Quality

Table 6 displays BLEU and SCS scores for each model in order to further validate generation performance. Strong semantic alignment and linguistic fluency even in the face of scenario shifts are demonstrated by the Proposed TL model's highest values on both metrics.

Table 5. Comparative Example of Operation Sequence Outputs Across Models

Scenario	Model	Sequence Snippet	Correct Order	Semantic Integrity	Safety Compliance	Notes
A	Reference	Close #5 breaker → Check voltage → Load switching	√	√	√	Ground truth
A	MLP	Close #5 breaker → Load switching	×	×	×	Missing voltage check
A	LSTM	Close #5 breaker → Load switching → Check voltage	×	√	×	Incorrect sequence
A	Transformer	Close #5 breaker → Check voltage → Load application	√	√	×	Missing safety notice
A	Proposed TL	Close #5 breaker → Check voltage → Load application	√	√	√	Fully aligned with logic
B	Reference	Open #7 switch → Isolate fault → Restore load	√	√	√	Ground truth
B	MLP	Open #7 switch → Restore load	×	×	×	Isolation step missing
B	LSTM	Restore load → Open #7 switch → Isolate fault	×	√	×	Misordered actions
B	Transformer	Open #7 switch → Restore load → Isolate fault	×	√	×	Misordered safety steps
B	Proposed TL	Open #7 switch → Isolate fault → Restore load	√	√	√	Fully correct

Table 6. Quantitative Evaluation of Operation Sequence Generation Quality

Model	BLEU Score	SCS (Semantic Consistency)
MLP	0.42	0.58
LSTM	0.61	0.74
Transformer	0.68	0.81
Proposed TL	0.83	0.92

The results demonstrate that the suggested TL model outperforms baseline architectures in generating cross-scenario operation sequences. Through the synergistic integration of transfer learning and domain-specific semantic modeling, the approach ensures adherence to safety and operational protocols while simultaneously improving linguistic and semantic fidelity. This creates a solid technical foundation for the deployment of intelligent scheduling systems in real-world distribution networks.

4.6. Experiment 5: Impact of Target Domain Data Proportion on Generalization Performance

In order to investigate how the size of target domain data influences the generalization capability of cross-scenario operation sequence generation, this experiment compares the performance of the proposed transfer learning model (Proposed TL) with a baseline model trained without transfer (No Transfer) under various proportions of target domain data. All experiments are conducted under identical pretraining conditions on the source domain.

Based on the Caltech 28-Bus Digital Twin dataset described in Section 4.1.2, the source and target domains are constructed as follows: (1) The fixed source domain for pretraining includes a range of topological configurations and operational scenarios. (2) The target domain is used for model adaptation and validation during the transfer stage to assess the cross-scenario operation sequence generation capability.

10%, 30%, 50%, 70%, and 100% of the training data come from the target domain, respectively. For each proportion, stratified random sampling is used to ensure that network structures and operational conditions are representative. All models are trained under consistent hyperparameters (batch size = 64, learning rate = 1×10^{-4}, training epochs = 50). The performance metric, which is computed on a fixed target domain validation set, is the Relative Normalized Squared Error (RNSE). Each setting is repeated three times and the average is reported to minimize the influence of randomness.

Figure 8 displays the trend of RNSE values for both approaches across various target domain training data percentages. The x-axis displays the data proportion (from 10% to 100%), and the y-axis represents the RNSE, which displays the generalization performance across different data scales.

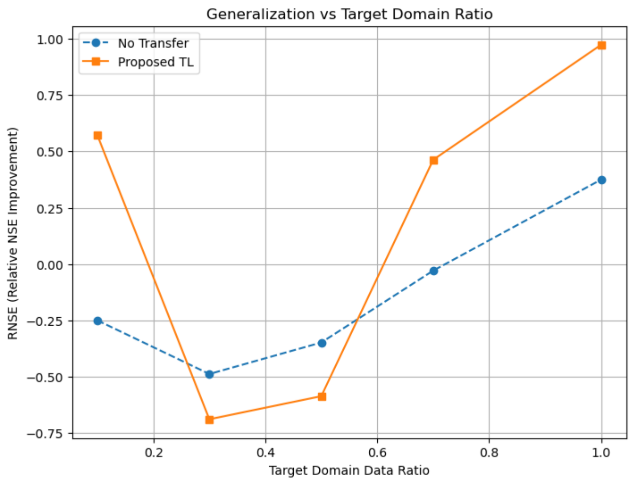

Figure 8. RNSE Trends Under Varying Target Domain Data Proportions

Three major observations can be made from Figure 8: (1) Low Data Regime (10%): The Proposed TL model achieves a positive RNSE (~0.55) under very limited target domain data, whereas No Transfer produces a negative RNSE (~−0.25). This suggests that the transfer learning approach makes good use of source knowledge to generalize in cross-scenario settings with limited resources. (2) Medium Data Regime (30%–50%): The RNSE of Proposed TL declines to negative values (~−0.70 and ~−0.60), falling below No Transfer (~−0.50 and ~−0.35). This implies the existence of negative transfer, in which generalization at intermediate data scales may be hampered by unstable domain alignment. (3) High Data Regime (70%–100%): Proposed TL's RNSE significantly outperforms No Transfer (~0.38) when the target domain proportion reaches 70%, turning positive once more (~0.47) and peaking at 100% (~0.98). Although No Transfer benefits modestly from additional data, its overall improvement remains limited.

The findings show that in both low-data and high-data regimes, Proposed TL achieves superior generalization. It efficiently repurposes source domain knowledge to produce high-quality operation sequences in low-resource environments. The fact that its performance keeps getting better in environments with lots of data shows how scalable it is. However, intermediate data scales must carefully manage domain alignment to lower negative transfer risks. These features imply that the suggested approach is suitable for real-world implementation in distribution networks, providing robust flexibility in situations where data is limited and improved performance in situations where data is plentiful, enabling dependable and effective cross-scenario intelligent dispatching.

4.7. Experimental Results Analysis

In order to fully evaluate the proposed transfer learning-based method for cross-scenario operation sequence generation and generalization in distribution networks, this section looks at the results from six perspectives: convergence behavior, cross-domain performance, architectural adaptability, generation quality, data sensitivity, and robustness under perturbation.

(1) Convergence Behavior: The proposed TL model achieves low validation loss from the early training stages and converges in fewer epochs by utilizing source domain pretraining and partial parameter freezing. The convergence speed increases by roughly 44% for fine-tuning and 73% for Proposed TL when compared to the No Transfer baseline. Additionally, with smoother loss descent curves and lower final loss values, the proposed TL model demonstrates improved training efficiency and stability in challenging cross-scenario tasks.

(2) Cross-Scenario Performance: The suggested TL model outperforms both Fine-tuning and No Transfer approaches in target domain testing, achieving lower prediction errors (MSE = 0.06, MAE = 0.245) and higher generalization metrics (GPI ≈ 0.47). The suggested TL benefits from the topologically guided transfer mechanism, which improves domain alignment and lessens the impact of catastrophic forgetting, but fine-tuning shows significant performance variance when data is limited.

(3) Architectural Adaptability: The suggested TL model consistently demonstrates good generalization performance across several deep learning frameworks, including MLP, LSTM, Transformer, and GNN. Significantly, it attains the highest RNSE score of 0.96, demonstrating that the suggested approach preserves strong transfer capabilities regardless of the underlying network architecture.

(4) Quality of Generated Operation Sequences: The proposed TL produces operation sequences that are very similar to the reference in terms of action order, semantic coherence, and safety protocol compliance. It attains a BLEU score of 0.83 and an SCS (semantic consistency score) of 0.92, outperforming all comparison models. These findings show that the model satisfies the practical and legal requirements of power dispatch operations while maintaining both temporal logic and semantic integrity.

(5) Sensitivity to Data Scale: The suggested TL achieves an RNSE of about 0.55 when only 10% of the target domain data is available, compared to −0.25 for No Transfer, showing a notable advantage in low-resource scenarios. Proposed TL's performance briefly drops at 30%–50%, most likely due to incomplete knowledge transfer and erratic domain alignment. But it rapidly recovers at 70% and exceeds all baselines at 100%, with a peak RNSE of about 0.98. This implies a high degree of adaptability across data scales, especially in cases where data is scarce.

(6) Robustness under Noise: To evaluate the model's stability in the presence of input disturbances, Gaussian noise with varying standard deviations (σ = 0.05, 0.1, 0.2, 0.3, 0.5) is added to simulate sensor inaccuracies, communication fluctuations, and data acquisition anomalies.

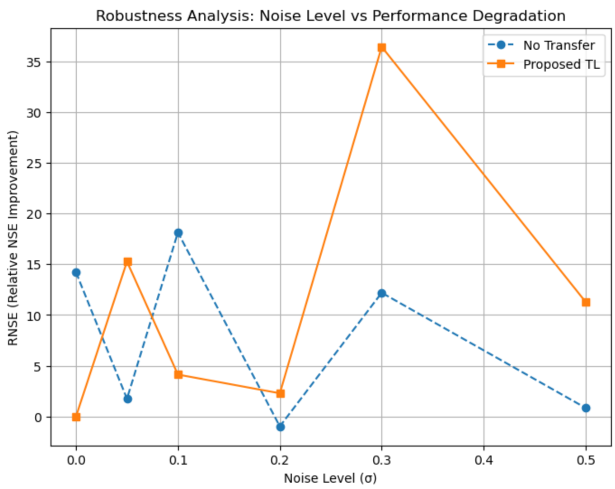

Figure 9. Robustness Evaluation under Varying Noise Levels

In Figure 9, the x-axis denotes noise standard deviation (σ), and the y-axis shows corresponding RNSE values. While Proposed TL performs slightly better, both models maintain strong generalization under low noise conditions ($\sigma \leq 0.1$). As noise rises to $\sigma = 0.2$, the RNSE of No Transfer gets closer to zero, but the Proposed TL remains unchanged. Proposed TL shows noticeably greater resilience at higher noise levels ($\sigma \geq 0.3$). For example, at $\sigma = 0.3$, its RNSE is around 36 (vs. 12 for No Transfer), and even under $\sigma = 0.5$, it maintains an RNSE of ~11, whereas No Transfer degrades to ~1, yielding numerous illogical actions in the generated sequences.

Overall, the proposed transfer learning approach performs better on cross-scenario operation step generation tasks. Strong noise resistance, rapid and steady convergence, excellent generation quality, and high predictive accuracy are all displayed. Data heterogeneity, signal interference, and small sample sizes are just a few of the real-world distribution network operations issues that it can successfully handle thanks to these features. As a result, it offers a solid basis for the deployment of intelligent dispatch systems in real-world engineering scenarios.

5. Conclusions

This paper suggests a transfer learning-based framework for operation sequence generation and generalization optimization in order to address the inadequate generalization capability of operation step generation models in cross-scenario distribution network applications. The suggested approach efficiently transfers prior knowledge through source-domain pretraining, topology-aware guidance, and feature adaptation mechanisms in the target domain. This increases the model's adaptability and stability when dealing with difficult operating conditions and sparse data.

A thorough experimental framework was built in order to confirm the efficacy of the suggested strategy. This framework included analyses of convergence characteristics, target domain performance, adaptability to various deep learning architectures, generated sequence quality, sensitivity to training data proportions, and robustness against perturbations. The experimental results demonstrate that: (1) The proposed transfer learning strategy achieves faster convergence, higher prediction accuracy, and better generalization performance when compared to non-transfer and traditional fine-tuning techniques. (2) The proposed TL consistently achieves optimal performance across various deep learning architectures, with a focus on task adaptation and generalization. (3) In terms of action order, semantic integrity, and safety protocol compliance, the output from the proposed TL is extremely similar to actual operation documents in operation sequence generation tasks. (4) The model's robustness under varying target domain data proportions and noise interference, as well as its strong few-shot transfer capabilities, make it ideal for real-world deployment scenarios.

Despite these positive results, there are still some limitations. On the one hand, performance may deteriorate in situations with extreme heterogeneity, even though the suggested feature alignment mechanism somewhat reduces domain distribution discrepancies. However, the method's use in completely unsupervised or zero-shot scenarios is limited because it still depends on partially labeled data from the target domain. Furthermore, the study mostly uses static datasets rather than analyzing the model's performance in dynamic scenarios like online dispatch or real-time system scheduling.

The proposed transfer process for this study can be summarized as follows: Initially, the model learns fundamental skills from the source domain, including operational expression patterns, topological comprehension, and temporal reasoning. The development of safe and logical operation sequences even in the presence of sparse data is made possible by the application of these skills in particular target domain scenarios. In addition to making cross-scenario knowledge migration easier, this procedure lays the groundwork for the creation of high-caliber, automated operation steps.

In conclusion, the suggested generalization-enhanced transfer learning approach successfully addresses important engineering issues like domain distribution shifts, data noise, and small sample sizes. By enhancing the robustness and generalization of operation sequence generation models across scenarios, it offers solid theoretical and technical support for the practical implementation of intelligent distribution dispatch systems. In order to improve the model's adaptive optimization capabilities in dynamic environments and enable wider practical applications, future research will investigate the integration of online transfer mechanisms and reinforcement learning.

Acknowledgements
We would like to extend our gratitude to the project coordinators and funding bodies for their support and resources, which were instrumental in the successful completion of this research.

Author Contributions
Fangzhou Hao: conceptualization, methodology, investigation, and writing—original draft. Shibo Li: data curation, formal analysis, validation, and visualization. Jiangwei Wu: resources, supervision, project administration, and writing—review & editing. Shuhua Chen: funding acquisition, supervision, project administration, and writing—review & editing.

Funding
This work was supported by the Science and Technology Project of Guangzhou Power Supply Bureau, "Research on a Comprehensive IoT-Based Anti-Misoperation System for the Full-Process Operation of Distribution Networks" (Grant No. 030114KC23110003).

Data Availability Statement
All relevant data are included in the paper. Further inquiries can be directed to the corresponding author.

Conflicts of Interest
The authors declare no conflicts of interest.

References

[1] Qinglong L, Xiaodong W, Song X, et al. Analysis of distribution network reliability based on distribution automation technology. Energy Informatics, 2025, 8(1): 1-18.

[2] Yan Z, Gao Z, Navesi R B, et al. Smart distribution network operation based on energy management system considering economic-technical goals of network operator. Energy Reports, 2023, 9: 4466-4477.

[3] Zhang P, Xu Y, Yue X, et al. Research on Intelligent Anti-misoperation of Operation Ticket in Urban Distribution Network. Journal of Physics: Conference Series. IOP Publishing, 2022, 2384(1): 012039.

[4] Pylarinos D. Investigating the effect on productivity of a geospatial ticket management system for power distribution network studies. Engineering, Technology & Applied Science Research, 2023, 13(5): 11616-11621.

[5] Wang J, Lan C, Liu C, et al. Generalizing to unseen domains: A survey on domain generalization. IEEE transactions on knowledge and data engineering, 2022, 35(8): 8052-8072.

[6] Chen C, Tang L, Liu F, et al. Mix and reason: Reasoning over semantic topology with data mixing for domain generalization. Advances in Neural Information Processing Systems, 2022, 35: 33302-33315.

[7] Talaei Khoei T, Ould Slimane H, Kaabouch N. Deep learning: systematic review, models, challenges, and research directions. Neural Computing and Applications, 2023, 35(31): 23103-23124.

[8] Hosna A, Merry E, Gyalmo J, et al. Transfer learning: a friendly introduction. Journal of Big Data, 2022, 9(1): 102.

[9] Olujimi P A, Ade-Ibijola A. NLP techniques for automating responses to customer queries: a systematic review. Discover Artificial Intelligence, 2023, 3(1): 20.

[10] Buterez D, Janet J P, Oglic D, et al. An end-to-end attention-based approach for learning on graphs. Nature Communications, 2025, 16(1): 5244.

[11] Li J, Yu Z, Du Z, et al. A comprehensive survey on source-free domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(8): 5743-5762.

[12] Chen L, Chen H, Wei Z, et al. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 7181-7190.

[13] Liao W, Bak-Jensen B, Pillai J R, et al. A review of graph neural networks and their applications in power systems. Journal of Modern Power Systems and Clean Energy, 2021, 10(2): 345-360.

[14] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 2020, 32(1): 4-24.

[15] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Advances in neural information processing systems, 2017, 30.

[16] Cai X, Xiao R, Zeng Z, et al. ITran: A novel transformer-based approach for industrial anomaly detection and localization. Engineering Applications of Artificial Intelligence, 2023, 125: 106677.

[17] Floriano A C, Avila S L, Fernandes R C. Structured vocabulary specific to power operation control centers. Energy Systems, 2024, 15(3): 1081-1104.

[18] Nayak A, Timmapathini H, Ponnalagu K, et al. Domain adaptation challenges of BERT in tokenization and sub-word representations of out-of-vocabulary words. Proceedings of the first workshop on insights from negative results in NLP. 2020: 1-5.
[19] Liu Z, Xu Y, Xu Y, et al. Improved fine-tuning by better leveraging pre-training data. Advances in Neural Information Processing Systems, 2022, 35: 32568-32581.
[20] Yang Z R, Han J, Wang C D, et al. Graphlora: Structure-aware contrastive low-rank adaptation for cross-graph transfer learning. Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1. 2025: 1785-1796.
[21] Liu H, Bi T, Xu S, et al. A full-view synchronized measurement system for the renewables, controls, loads, and waveforms of power-electronics-enabled power distribution grids. IEEE Transactions on Smart Grid, 2021, 13(5): 3879-3890.