RoadPowerFM: Graphormer-JEPA based Foundation Model for Road-Power Coupling Network

Zeyuan Niu, *Student Member, IEEE,* Yihong Tang, Jiamei Li, Qian Ai, *Senior Member, IEEE,* and Xing He, *Senior Member, IEEE*

***Abstract*—Coupling effects in road-power coupling networks (RPCNs) attract increasing attention, as they are crucial for addressing the road-power intertwined challenges caused by the rapid growth of electric vehicles (EVs). To solve these challenges, this paper, based on Graphormer and Joint-Embedding Predictive Architecture (JEPA), proposes a road power foundation model (RPFM) as a generic solution. Our RPFM, by employing graph-pretraining methods to bridge RPCNs, demonstrates exceptional performance in downstream tasks such as EV Charging Station (EVCS) Load Prediction, Road Traffic Prediction, and EVCS Location Planning. By experimenting on real world dataset, the proposed methodology is proved to be generic and achieves state-of-the-art performance across downstream tasks by improving an average of 7.53% of baseline models' performance.**

***Index Terms*—foundation model, road-power coupling network, graph-transformer, joint-embedding predictive architecture**

I. Introduction

THE large-scale adoption of electric vehicles (EVs) has significantly impacted urban transportation and energy systems, leading to coupling effects [1] in road-power coupling networks (RPCNs). These coupling effects represent a typical emergence phenomenon [2] that transcends the scope of traditional mechanism-based power system or traffic knowledge. Besides, data-driven approaches struggle due to the lack of labeled data, which hinders their ability to effectively capture and interpret the complex relationships.

Intuitively, the knowledge embedded within these coupling effects holds valuable insights for improving the efficiency and resilience of urban systems. These insights provide the foundation and motivation for this research, which aims to extract and leverage this knowledge to address the unique challenges posed by RPCNs.

Researchers have focused on understanding the intricate dynamics between EVs, power systems, and road networks. A key concept emerging in this field is the RPCN [3], which represents the interconnected and interdependent nature of transportation and power systems in the context of electric vehicles. Existing studies have explored the impact of EVs on power systems [4] and examined the interdependence between road and power networks [5]. The main goal of this research field is to optimize traffic flow or to improve the robustness of the power grids under massive EVs on the road [6, 7]. Several works have delved into the optimal planning of EVCSs using tailored models and algorithms. For instance, methods such as distribution network expansion planning (DNEP) and integrated deep learning approaches are applied in [8, 9, 10]. Nareshkumar et al. propose a two-stage strategy for the optimal planning of EVCSs in [11]. Furthermore, Li et al. investigate the role of electric buses (EBs) in enhancing the resilience of power distribution networks (PDNs) and urban transportation networks (UTNs) in [12].

In the field of RPCN models, authors in [13] study the optimal planning of Plug-in Electric Vehicle (PEV) fast-charging stations (FCSs) by considering the interactions between transportation and electrical networks, proposing a capacitated-flow refueling location model (CFRLM) that accounts for PEV charging demand, driving range constraints, and transportation flow. Qian et.al in [14] propose a holistic framework to improve the operations of coupled electric PDN and UTN by incorporating EV charging services by introducing a bi-level optimization model where the upper level focuses on determining EV charging service fees (CSF) to minimize total social costs. A novel approach to coordinating EV charging within interconnected transportation and power systems is introduced in [15], focusing on the challenge of reducing carbon emissions by optimizing both EV charging and traffic flows. Authors in [16] focus on the increasing interdependence between power networks and transportation networks, proposing a supply-demand-based methodology that incorporates the functional heterogeneity of FCSs to better understand the dependency between the two networks. Congestion awareness and network balance are studied in [17, 18, 19] together with the coordinated system joint reaction methodology.

Beyond developing complex models and optimization strategies, recent studies have explored model-free approaches to address the challenges of integrating EVs with power and transportation systems. These approaches incorporate constraints from road networks or prioritize certain optimization as a core objective. For instance, multi-agent reinforcement learning (MARL) and multi-agent deep reinforcement learning (MADRL) have been applied to optimize EV charging

This work was supported by the National Key R&D Program of China 2021YFB2401203.

Qian Ai, Zeyuan Niu, Jiamei Li, and Xing He are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. Yihong Tang is with the Department of Civil Engineering, McGill University, Montreal, Canada.

Zeyuan Niu and Yihong Tang contribute equally to this work.

Corresponding author: Yihong Tang (yihong.tang@mail.mcgill.ca).

Manuscript received on April 23, 2025

schedules within smart grids, particularly under vehicle-to-grid (V2G) scenarios [20, 21, 22, 23, 24]. These methods dynamically balance EV energy demands with grid capacity while considering transportation system constraints. Furthermore, neural networks (NNs) have demonstrated potential in forecasting EV energy needs and modeling traffic interactions, providing data-driven solutions to the complex interplay between EVs, power systems, and road networks [25, 26].

Despite the considerable progress in the field, existing studies face significant limitations. They are either heavily dependent on specific models or rely on model-free approaches that require vast amounts of labeled data, making them challenging to generalize to scenarios beyond their original context. However, the rapid advancements in foundation models (FMs) [27], such as large language models (LLMs) [28], known for their versatility, accessibility, and adaptability across diverse scenarios [29, 30], inspire new approaches to uncovering underlying mechanisms for solving complex problems with these general and powerful models.

FMs are a class of large-scale, pretrained machine learning models that serve as a versatile and general-purpose foundation for a wide range of downstream tasks. They are distinguished from traditional models by scalability, adaptability, and capability to generalize across domains. Studies have been conducted in fields like medical [31] and computer vision [32, 33], demonstrating their wide applicability. For example, Huang et al. propose a foundation model to solve the power system's time series problem in [34]. However, despite these advancements, there remains a significant gap in the application of foundation models to RPCN systems.

The construction of an FM requires a powerful and flexible architecture. Since the emergence of the Transformer [35] architecture, proposed to address the computational efficiency and long-range dependency modeling issues in traditional sequence-to-sequence models [36], many researchers have worked on this network architecture, leading to numerous Transformer-based architectures. Graph Transformer (GT) models graph data by mapping the structural information of graphs (nodes and edges) into a form suitable for the Transformer model [37]. Unlike traditional Graph Convolutional Networks (GCNs), Graph Transformer relies on the self-attention mechanism to capture relationships between nodes, enabling flexible modeling of the global structure of graphs [38]. Building on the standard Transformer architecture, the concept of Graphormer is proposed in [39], leading to various subsequent studies. Lin et al. introduce Mesh Graphormer in [40], a graph-convolution-reinforced transformer for 3D human pose and mesh reconstruction. Chang et al. propose a Multi-channel Graphormer in [41], ensuring each node interacts with relevant and valuable targets.

To train and build an FM, self-supervised learning (SSL) presents an effective approach. Traditional SSL methods, such as Joint-Embedding (Contrastive) Architecture and Generative Architecture [42], have been widely used. One promising method, Joint-Embedding Predictive Architecture (JEPA) [43], learns the structural relationships of data by making predictions in the embedding space. Relevant to our research, Skenderi et al. [44] demonstrate that graph-level representations can be effectively modeled via Graph-JEPA. However, Graph-JEPA struggles with processing large graphs. To address this, we propose a Graphormer-JEPA-based road power foundation model (RPFM), where we replace the encoder with Graphormer to better handle large graph learning.

Despite significant research interest in exploring the complex interactions among EVs, power systems, and transportation networks within RPCNs, existing approaches still encounter critical limitations:

- **Limited Generalization Ability:** Existing studies often rely heavily on specialized models tailored for specific scenarios or tasks, such as DNEP, RL-based methods, or traditional GCN-based architectures, which frequently struggle to generalize efficiently to diverse scenarios involving urban scales, graph topologies, and limited sensor data availability.
- **Heavy Reliance on Labeled Data:** Current supervised and reinforcement learning approaches typically require substantial amounts of labeled data or interaction-based data, limiting their practical scalability. Particularly in RPCNs, obtaining large-scale, high-quality labeled data is costly and challenging, restricting the applicability of existing models in data-scarce environments.
- **Lack of a Unified Foundation Model Framework:** Most prior studies focus on single-task solutions, lacking a comprehensive foundation model framework capable of effectively handling multiple tasks within RPCNs simultaneously.

In the context of RPCNs, the inherent graph-structured nature of road networks and power grids presents unique challenges and opportunities for model development. Most downstream tasks related to these systems, such as predicting optimal EVCS locations, forecasting EVCS loads, and modeling traffic patterns, rely heavily on understanding and processing graph data. To address these challenges and build a method that works across various road and power network graphs while supporting diverse tasks, we propose the RPFM for RPCNs based on Graphormer-JEPA. This model is pretrained using a self-supervised learning approach to enhance its adaptability and generalization. For the node-level encoder, we use Graphormer because of its scalability and flexibility, which make it more suitable for pretraining compared to traditional GCNs. This choice ensures our model can efficiently handle complex graph structures. Additionally, recognizing the limitations of incomplete sensor data, we adopt the JEPA, a self-supervised learning paradigm designed to learn the intrinsic relationships within graph structures without compromising generalizability. We also introduce key innovations tailored to the unique context of RPCNs. Specifically, the model incorporates a novel community-aware positional encoding mechanism, which enhances the capture of hierarchical dependencies within coupled road-power networks, and an adaptive node and edge feature augmentation method to help the model with graph representation learning.

The resulting foundational model demonstrates strong transferability, allowing it to be easily adapted for various downstream tasks. To evaluate its effectiveness, we designed three key tasks: EVCS Load Prediction, Road Traffic Prediction and EVCS Location Planning.

Table I: Comparison between RoadPowerFM and other road-power coupling network approaches. (* indicates that the method is capable but with limited ability)

	Graph Embedding	Transferability	Emergence	Label Efficiency	Multi-task	Computational Efficiency
Mechanism-based Models	×	×	×	✓	×	×
Data-driven Methods	✓	×	×	×	×	×
Multi-agent RL [20–24]	✓	×	✓	×	×	×
DNEP Methods [8–10]	×	×	×	✓	×	✓
Two-stage Strategies [11]	×	×	×	✓	×	✓
Neural Networks [25–26]	✓	×	×	×	×	×
Generative Architecture [42]	*	✓	✓	✓	✓	×
RoadPowerFM (ours)	✓	✓	✓	✓	✓	✓

In regard to the defects of current studies mentioned above, the comparison of existing studies is shown in Table I and key contributions of this work are summarized as follows:

- Conceptually: We propose the concept of FM in RPCNs to address their inherent complexities and emergence phenomenon.
- Methodologically: We propose a pretraining framework named **R**oad **P**ower **F**oundation **M**odel (RPFM), as shown in Figure 3. It integrates advanced positional encoding mechanisms to model the intricate dynamics of RPCNs.
- Practically: The RPFM framework demonstrates strong adaptability to diverse downstream tasks. We also design a tailored fine-tuning method to adjust the pretrained FM for these tasks. Experimental results show its effectiveness, with average improvements of 10.90%, 3.19%, and 8.49% in EVCS Load Prediction, Road Traffic Prediction, and EVCS Location Planning tasks, respectively.

The rest of the contents are organized as follows: Section II introduces the relevant backgrounds and formulates the research problem. Section III demonstrates the design and applications of the proposed RPFM. Section IV verifies our method with a case study consisting of three different downstream tasks based on real-world dataset. Section V is the conclusion and outlook of our work.

II. Preliminaries

A. Problem formulation

The goal of the pretraining process in this research is to learn a representation of the road-power coupling network that can be used effectively for a variety of downstream tasks.

In this work, we model the RPCN as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. The structure of the graph is efficiently represented using the sparse adjacency matrix A, which is a matrix of shape $[2, E]$, where E is the number of edges. The rows of A contain the indices of source and target nodes for each edge, respectively. The EVCS graph is combined with the road graph through location matching, where the locations of EVCSs are aligned, and the *has_EVCS* feature is added to the node features of the road graph. The formulation of nodes' feature $\mathbf{X}$ is:

$$\mathbf{X} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbb{R}^l, v_i \in V, i = 1, \ldots, N\} \in \mathbb{R}^{N \times l} \tag{1}$$

where $\mathbf{x}_i$ is the feature vector of node v_i, l is the dimension of feature, and N is the total number of nodes. The definition of $\mathbf{x}_i$ is shown in (2).

$$\mathbf{x}_i = x_i^{\text{Location}} \| x_i^{\text{has\_EVCS}} \tag{2}$$

where x_i^{Location} is the coordinates vector of node i, consisting of longitude and latitude, and $x_i^{\text{has\_EVCS}}$ is a 0-1 vector indicating whether the node has an EVCS.

The feature of edges $\mathbf{E}$ is defined as:

$$\mathbf{E} = \{\mathbf{e}_{ij} \mid \mathbf{e}_{ij} \in \mathbb{R}^k, e_{ij} \in E, i, j = 1, \ldots, N\} \in \mathbb{R}^{E \times k}, \tag{3}$$

where $\mathbf{e}_{ij}$ is the feature vector of edges (v_i, v_j), E is the total number of edges, and k is the dimension of edge feature. The definition of $\mathbf{e}_{ij}$ is shown as follows:

$$\mathbf{e}_{ij} = e_{ij}^{\text{Length}} \| e_{ij}^{\text{Traffic}} \tag{4}$$

where e_{ij}^{Length} is treated as a 1-dimensional vector representing the length of the road between nodes i and j, and e_{ij}^{Traffic} represents a vector containing two elements: the average traffic speed and the traffic jam factor on this road, both measured over the past 30 days.

Based on the definitions provided above, the pretraining process can be understood as the model learning the mapping $\mathcal{M}$ between context and target graphs. This offers an intuitive understanding of the pretraining procedure, where the objective of pretraining is to learn a graph representation $\mathbf{z}$, which can be described as follows:

$$\mathbf{z} = \mathcal{M}(G) \tag{5}$$

The following subsections introduce the background of the sub-architectures of the proposed RPFM.

B. Joint-Embedding Predictive Architecture (JEPA)

JEPA, conceptually similar to Generative Architectures, based on which many LLMs are constructed, learns to predict the embeddings of a signal y from a compatible signal x. In our study, JEPA enhances the modeling capability of RPCNs, which are highly structured and organized, by capturing predictive relationships between subgraphs (communities). This enables the model to understand not only local graph properties but also predictive spatial dependencies between different regions of the RPCN.

The workflow of JEPA is illustrated in Figure 1. Input signals, x for context and y for target, are encoded into embeddings through the x-encoder and y-encoder, respectively. The predictor takes the embeddings of x and optionally additional information z to predict $\hat{s}_y$. A function $D(\hat{s}_y, s_y)$ measures the discrepancy between the predicted embedding and the true

target embedding, serving as the pretraining objective. The resulting model can then be applied to downstream tasks.

The pretrained parameters of x-encoder are then utilized in downstream tasks, functioning as a generator of initial network representations. In our work, the purpose of pretraining is to learn a generalizable foundation representation that encompasses both the topological and feature-level representation in a large, heterogeneous RPCN. This yields a more compact high-level embedding space rather than raw features and a more expressive latent space enriched by the JEPA-based self-supervision, ultimately benefiting downstream tasks.

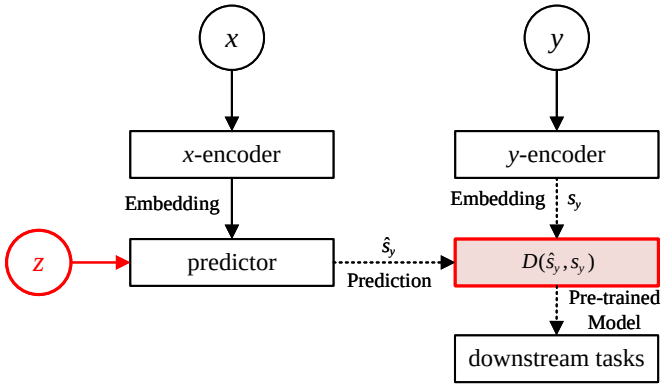

Figure 1: Joint-Embedding Predictive Architecture.

C. Graphormer

Figure 2 illustrates the architecture of the Graphormer Encoder, which is composed of N identical blocks. Each block includes five key sub-modules: a Layer Norm [45], a Multi-Head Self-Attention module [35], a Graph Residual Block [46], a second Layer Norm, and finally, a Multi-Layer Perceptron (MLP) [47]. What sets Graphormer apart is its seamless integration of graph convolution into the network, allowing it to model fine-grained local interactions while maintaining scalability effectively.

III. Design and Applications of RPFM

A. Pretrain

The FM is essentially the graph representation learned from Equation (5). The process of training the FM is referred to as pretraining, which serves as the foundation for its downstream applications. The key feature of RPFM is to explicitly capture hierarchical and spatial dependencies within RPCNs in the pretraining stage, leveraging a community-aware positional encoding method named Random Walk Structural Embedding (RWSE)[48, 49]. Additionally, by applying the Graphormer combined with the JEPA-based architecture, RPFM can efficiently encode both local interactions and global relationships. After pretraining, an additional training process is carried out on the pretrained FM, known as fine-tuning. During fine-tuning, RPFM adopts tailored optimization strategies to address the spatiotemporal characteristics of RPCNs. This adoption and integration strategy ensures temporal dependencies, along with spatial structure embedded from pretraining, are

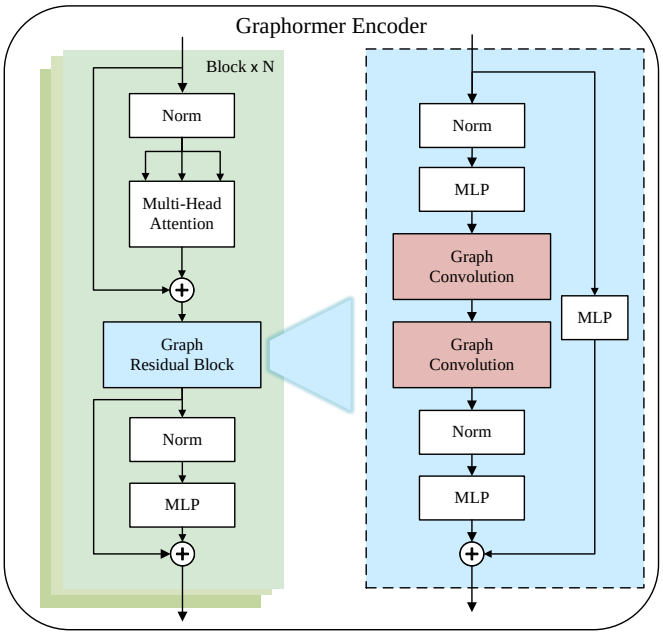

Figure 2: Architecture of a Graphormer Encoder.

jointly optimized, enhancing the model's capability in dealing with RPCN characteristics such as spatiotemporal dynamics.

The pretraining process does not entail the resolution of a specific task or the identification of a particular problem for the model. Rather, it represents a stage in the learning process through which the model gains access to global and local information between nodes in large-scale graph data. Accordingly, under most scenarios, it is necessary to fine-tune the pretrained model to adapt it to the requirements of specific tasks. The whole procedure of pretraining RPFM is demonstrated in Figure 3.

The pretraining of RPFM involves partitioning the input graph into subgraphs (communities) and predicting the representation of a randomly selected target community based on a single context community. This partitioning serves two main purposes. First, it reduces computational costs by focusing on smaller subgraphs. Second, it enables the model to better capture local structures within each community and learn dependencies across different parts of the graph. This enhances transferability between graph structures, making the pretrained model effective for diverse downstream tasks.

The Louvain algorithm [50] is known for its efficiency in detecting communities in large networks by optimizing modularity. At each training epoch, we randomly select one community as the context x and m others as targets $Y = \{y_1, y_2, ..., y_m\}$, enabling the model to generalize community representations effectively.

The encoding process of the Context and Target communities can be outlined as follows. First, the embedding of the context community is computed as:

$$\mathbf{z}^x = E_c(x), \tag{6}$$

Similarly, the embeddings of the target communities are

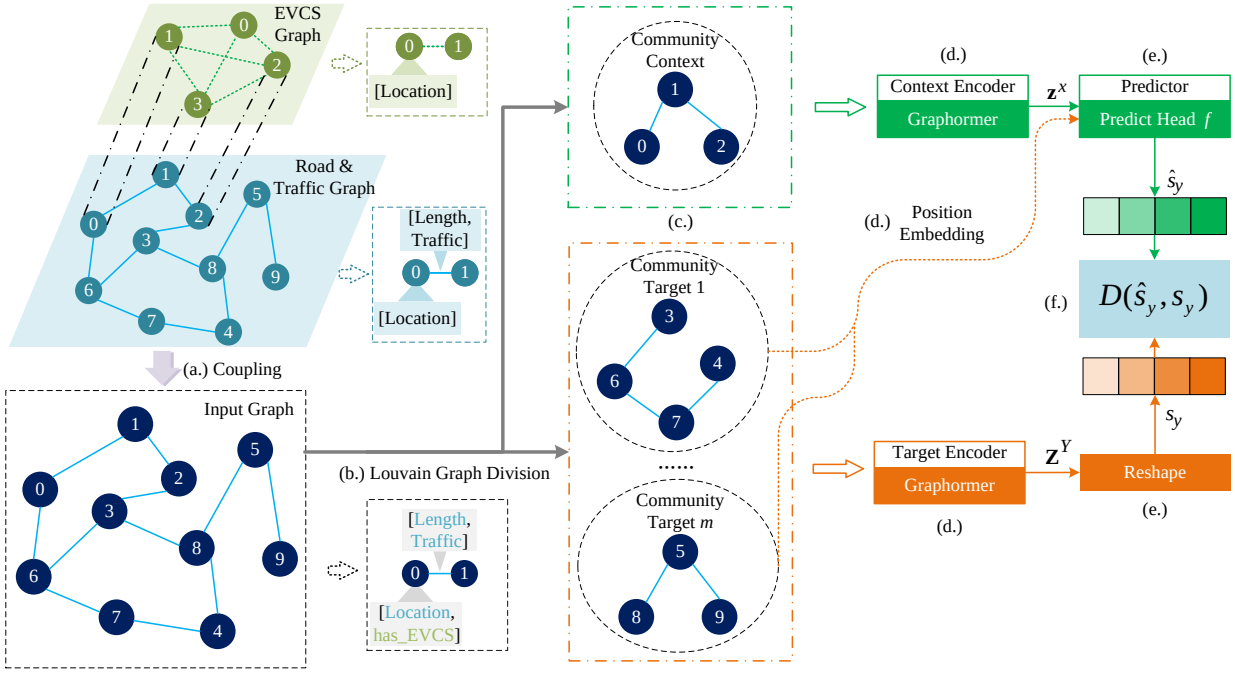

Figure 3: Pretraining procedure of RPFM: (a) Construct the RPCN graph by combining the EVCS graph and the road graph. (b) Divide the RPCN graph into communities. (c) Pair the divided communities at a 1:m ratio, designating them as context and target. (d) Generate community embeddings, $\mathbf{z}^x$ and $\mathbf{z}^y$, using the Context and Target Encoders, respectively. (e) Generate vectors $\hat{s}_y$ and s_y, with the target also providing a Position Embedding, which is combined with $\mathbf{z}^x$ and input into the Predictor to generate $\hat{s}_y$. (f) Through the energy function D and iterative learning, the graph representation of the RPCN is refined.

obtained as:

$$\mathbf{Z}^Y = \{\mathbf{z}^{y_1}, ..., \mathbf{z}^{y_m}\}\ ,\ where\ \ \mathbf{z}^{y_m} = E_t(y_m), \tag{7}$$

where $\mathbf{z}^x \in \mathbb{R}^d$ is the embedding of context community and $\mathbf{Z}^Y \in \mathbb{R}^{m\times d}$ is the embedding of target communities, d is the embedding dimension which is a hyper-parameter. E_c and E_t represent the context and the target graphormer encoders, respectively. It's worth mentioning that E_c and E_t can share the same set of parameters.

To predict the target community representation, the predictor f operates as follows:

$$\hat{s}_y = f(\mathbf{z}^x, \mathbf{P}) = \mathbf{W}_2\sigma(\mathbf{W}_1(\mathbf{z}^x + \mathbf{P}) + \mathbf{b}_1) + \mathbf{b}_2, \tag{8}$$

where $\mathbf{P}$ is the position embedding of the target communities, implemented as RWSE, $\mathbf{W}_n$ is the n-th trainable weight matrix, and $\mathbf{b}_n$ is the n-th bias vector, and $\sigma(\cdot)$ is the sigmoid nonlinearity [51]. The vector s_y is a reshaped version of $\mathbf{Z}^Y$, where $s_y, \hat{s}_y \in \mathbb{R}^d$.

Finally, the energy function $D(\cdot,\cdot)$ is computed to quantify the similarity between the predicted and true representations as follows:

$$D(\hat{s}_y, s_y) = \begin{cases} \frac{1}{2}\|\hat{s}_y - s_y\|^2, & \text{if } \|\hat{s}_y - s_y\| < \delta, \\ \delta \cdot \|\hat{s}_y - s_y\| - \frac{1}{2}\delta^2, & \text{otherwise.} \end{cases} \tag{9}$$

where δ is a smooth factor, typically set to 1 [52, 53] to ensure smooth convergence during the pretraining process and to balance the contribution of the energy term across different scenarios. The use of energy function also helps to mitigate gradient explosion when errors are large, thereby enhancing the stability of training [43, 54].

B. Fine-tune and Downstream Tasks

To adapt the pretrained RPFM to different downstream tasks, we use the Low-Rank Adaptation method (LoRA) [55], which facilitates efficient fine-tuning.

The proposed RPFM could be fine-tuned with different data on various graph structures, making it highly adaptable to a wide range of RPCN applications. However, a challenge arises due to data dimensions and formats differences between the pretraining and fine-tuning phases. To resolve this, a mapping layer is introduced to align the fine-tuning data dimensions with those used during pretraining, ensuring compatibility and consistency. In general, linear mappings can effectively resolve the dimensional mismatch. However, tailored mapping layers can be employed for more specialized tasks to handle task-specific requirements better.

The optimization objective of LoRA is defined as:

$$\min_{\theta} \mathbb{E}\left[\mathcal{L}\left(\text{Adapt}\left(\mathbf{X}', \mathbf{E}'; \theta\right), L\right)\right], \tag{10}$$

where $\mathbb{E}[\cdot]$ represents the expectation operator, $\mathcal{L}$ is the loss function quantifying the discrepancy between model predictions and ground-truth labels L, and Adapt$(\cdot)$ denotes the LoRA adapter layer. Here, $\mathbf{X}'$ and $\mathbf{E}'$ are dimension-aligned node and edge features encoded by the specially designed mapping layer $f_{\text{map}}(\cdot)$. It is important to observe that $\mathbf{X}'$ and $\mathbf{E}'$ do not represent conventional *data* in their raw form,

but rather exist in a transformed state more analogous to embeddings, having undergone dimensional alignment to facilitate compatibility with the pretrained model architecture. θ includes all trainable parameters in the fine-tuning stage, such as those of the mapping layers and LoRA adapters, while the pretrained RPFM parameters remain fixed during fine-tuning.

This paper presents three downstream tasks targeting both node-level and edge-level features of RPCNs: EVCS load prediction, road traffic prediction, and EVCS location planning. These tasks effectively combine RPFM with different baseline models, demonstrating the adaptability and utility of the proposed framework.

EVCS Load Prediction: EVCS load prediction is a spatiotemporal prediction task aimed at forecasting the EVCS load over time, based on historical data. This task involves modeling the temporal dynamics of the load, making it crucial for efficient energy management and planning.

Due to the strong temporal dependencies inherent in this task, a specialized, task-specific mapping layer is essential. The function $f_{\text{map}}(\cdot)$ must effectively handle these temporal dependencies, with the Long Short-Term Memory (LSTM) network [56] serving as an example due to its widespread use and strong empirical performance. However, it is important to emphasize that the LSTM architecture is not mandatory; the proposed RPFM framework offers remarkable flexibility, allowing for integration with various temporal modeling architectures based on specific application requirements.

In this task, the input data is represented as $\mathcal{X} \in \mathbb{R}^{N_1 \times T}$, where N_1 denotes the number of nodes (i.e., EVCSs), and T represents the total number of time steps. For a specific node i, $\mathcal{X}_i \in \mathbb{R}^T$ corresponds to the time series of load values for the i-th EVCS. Furthermore, $\mathcal{X}_i^t$ denotes the load value of the i-th EVCS at the time step t.

Formally, the objective of this task could be denoted as:

$$\min_{\theta_1} \mathbb{E}\left[\mathcal{L}_{\text{MSE}}(\text{Adapt}(f_{\text{map}}(\mathcal{X}, \mathbf{X}'); \theta_1), L_1)\right], \tag{11}$$

where θ_1 denotes the trainable parameters during fine-tuning of the RPFM with EVCS load data, L_1 represents the ground truth future loads of EVCSs, which have a shape similar to $\mathcal{X}$ but correspond to future time steps, and $\mathcal{L}_{\text{MSE}}$ is the mean squared error (MSE) loss function. For simplicity, $f_{\text{map}}(\cdot)$ is assumed to include the concatenation and necessary dimension alignment processes.

Road Traffic Prediction: Road traffic prediction is a spatiotemporal forecasting task, similar to EVCS load prediction, aimed at predicting traffic conditions, such as vehicle flow or speed, over a network of roads based on historical traffic data. The key distinction lies in the focus: while EVCS load prediction primarily deals with node features, road traffic prediction centers on edge features, represented as $\mathcal{E}^T \in \mathbb{R}^{E_1 \times T}$, where E_1 denotes the number of road segments (edges) and T represents the total number of time steps.

Similarly, the objective of road traffic prediction is:

$$\min_{\theta_2} \mathbb{E}\left[\mathcal{L}_{\text{MSE}}(\text{Adapt}(f_{\text{map}}(\mathcal{E}, \mathbf{E}'); \theta_2), L_2)\right], \tag{12}$$

EVCS Location Planning: This task is designed to identify potential locations for EVCSs within an RPCN. This task plays a key role in optimizing the deployment of EVCSs to ensure efficient charging infrastructure for electric vehicles.

Formally, the task is a binary classification problem that predicts whether a node hosts an EVCS based on its features $\mathbf{X}'$. The objective can be expressed as:

$$\min_{\theta_3} \mathbb{E}\left[\mathcal{L}_{\text{BCE}}(\text{Adapt}(\mathbf{X}'; \theta_3), L_3)\right], \tag{13}$$

where the notations align with those used in the previous tasks. Here, $\mathcal{L}_{\text{BCE}}$ represents the binary cross-entropy (BCE) loss [57], θ_3 denotes the trainable parameters for this task, and L_3 is the binary label indicating whether a node has an EVCS.

Eventually, the pseudo-codes of the proposed Graphormer-JEPA framework's pretraining and fine-tuning procedure are organized and descibed in Algorithm 1 and 2, respectively.

Algorithm 1 Pretraining Procedure of RPFM

Input: Graph $G = (V, E)$, node features $\mathbf{X}$, edge features $\mathbf{E}$
Output: Pretrained model parameters θ
Initialize Context Encoder E_c and Target Encoder E_t with parameters θ
Initialize Predictor f with parameters ϕ
for each training epoch **do**
    Divide input graph G into communities using Louvain algorithm
    Randomly select one community as context x
    Randomly select m communities as targets $\mathbf{Y} = \{y_1, y_2, ..., y_m\}$
    Compute context embedding: $z^x = E_c(x)$
    Compute target embeddings: $\mathbf{Z}^Y = \{z^{y_1}, ..., z^{y_m}\}$, where $z^{y_m} = E_t(y_m)$
    Generate position embedding P using RWSE
    Predict target representation: $\hat{s}_y = f(z^x, P)$
    Reshape $\mathbf{Z}^Y$ to s_y
    Compute energy function: $D(\hat{s}_y, s_y)$
    Update parameters θ, ϕ to minimize $D(\hat{s}_y, s_y)$
end for
return Pretrained parameters θ

Algorithm 2 Fine-tuning Process for Downstream Tasks

Input: Pretrained RPFM parameters θ, task-specific data $Data$, task-specific labels L, task type
Output: Fine-tuned model parameters θ_{task}
Load pretrained RPFM parameters θ and freeze them
Initialize LoRA adapter parameters θ_{task}
Initialize mapping layer parameters for dimension alignment
for each training epoch **do**
    Apply mapping layer: $Data' = f_{\text{map}}(Data)$
    Compute predictions: $\hat{L} = \text{Adapt}(Data'; \theta_{\text{task}})$
    Compute task-appropriate loss: $\mathcal{L}(\hat{L}, L)$
    Update parameters θ_{task} to minimize $\mathcal{L}$
end for
return Fine-tuned parameters θ_{task}

IV. Case Study

A. Data

The pretraining dataset used in this case study is constructed using open transportation[1], and EVCS data[2] of 10 U.S. cities.

[1] geoffboeing.com, here.com
[2] openchargemap.org

(a) Chicago (b) Los Angeles (c) New York City (d) Philadelphia

Figure 4: Visualization of graphs. Blue lines indicate the nodes and the edges connected, while the red nodes indicate the EVCSs' location.

On average, each city contributes approximately 110,000 nodes, 320,000 edges, and 290 EVCSs. A subset of these datasets is visualized in Figure 4.

The datasets used for the downstream tasks are as follows:

1) EVCS Load Prediction: A private dataset from Shenzhen, consisting of power usage data from 8 EVCSs over 3 days, recorded at a time interval of 1 minute.
2) Road Traffic Prediction: A public traffic dataset from Shenzhen, sourced from [58], with data recorded at a 15-minute time interval.
3) EVCS Location Planning: The Road-Power Graph of San Francisco is used for this task.

B. Implementation Details

The experiments were conducted on a high-performance server with 32GB of GPU memory and a computational capacity of approximately 15 TFLOPs.

1) Pretraining: The pretraining process takes approximately 15 mins with a total of 102,302,721 parameters. The parameters' configuration is shown in Table II.

Table II: Hyperparameter Settings for Pretraining.

Parameter	num_heads	num_layers	learning_rate	hidden_dim
Value	8	8	1e-3	512

2) Downstream Tasks: The parameters' configuration of downstream tasks is shown in Table III.

Table III: Hyperparameter Settings for Downstream Tasks.

Parameter	learning_rate	num_layers	batch_size	hidden_dim	num_heads
Range or Candidates	[1e-5, 1e-3]	{2, 4}	{32, 64}	{64, 100, 128, 256, 512}	{4, 8}

Across all downstream tasks we first perform an exhaustive hyper parameter search for each stand-alone baseline within the ranges listed in the above Table, and then lock this configuration when the same architecture appears inside its corresponding *+RPFM* variant; for example, in the EVCS Load Prediction task the LSTM attains its optimum with a learning rate of 1e-3, two layers, a batch size of 32, and a hidden dimension of 100, and these identical values are preserved for the LSTM component in *LSTM+RPFM* to guarantee a fair comparison.

For *EVCS Load Prediction* and *Road Traffic Prediction*, the dataset is divided 8:1:1 as training, validation, and test dataset. For *EVCS Location Planning*, the division of dataset can be referred in Figure 7.

C. Baselines

Spatial-Temporal Prediction Tasks: For EVCS Load Prediction and Road Traffic Prediction tasks, the proposed method is evaluated against baseline models designed for temporal modeling. These include the Gated Recurrent Unit (GRU) [59], the Long Short-Term Memory (LSTM) network [56], the Temporal Graph Convolution Network (T-GCN) [58], and the Spatial-Temporal Graph Attention Network (STGAT) [60].

Planning Task: For the EVCS Location Planning task, the proposed method is compared with graph-based baseline models, specifically the Graph Convolution Network (GCN) [61], GraphSAGE [62], and Graph Attention Network (GAT) [63], which focus on learning representations from graph-structured data.

D. Metrics

Spatial-Temporal Prediction Tasks: For the EVCS Load Prediction and Road Traffic Prediction tasks, performance is evaluated using three metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Accuracy.

Planning Task: For the EVCS Location Planning task, performance is assessed using three metrics: Precision, Recall, and F1 Score [64].

Calculations for these metrics are provided in Appendix A.

E. Results

1) EVCS Load Prediction: In this study, we utilize a private dataset from Shenzhen, comprising power data from eight EVCSs collected over three days at a 1-minute sampling interval. The input sequence consists of 60 timesteps, and the prediction target is the subsequent 15 timesteps. To evaluate the effectiveness, applicability, and flexibility of the proposed RPFM (Pretrained RPFM), we integrate it into various baseline models to assess performance improvements. The results on the test dataset are visualized in Figure 5, while a detailed

Table IV: Performance Comparison: EVCS Load Prediction (8 EVCSs × 4320 time steps).

Method	MAE ↓	RMSE ↓	Accuracy ↑	Improvement ↑
STGAT	94.10	138.89	0.7052	–
+ RPFM	**84.64**	**122.55**	**0.7497**	**9.38%**
T-GCN	97.49	142.16	0.6895	–
+ RPFM	**85.29**	**122.81**	**0.7468**	**11.48%**
LSTM	101.45	146.01	0.6784	–
+ RPFM	**92.43**	**124.53**	**0.7279**	**10.30%**
GRU	112.25	164.85	0.6279	–
+ RPFM	**96.47**	**142.97**	**0.6903**	**12.42%**

numerical comparison between our method and other baselines is presented in Table IV.

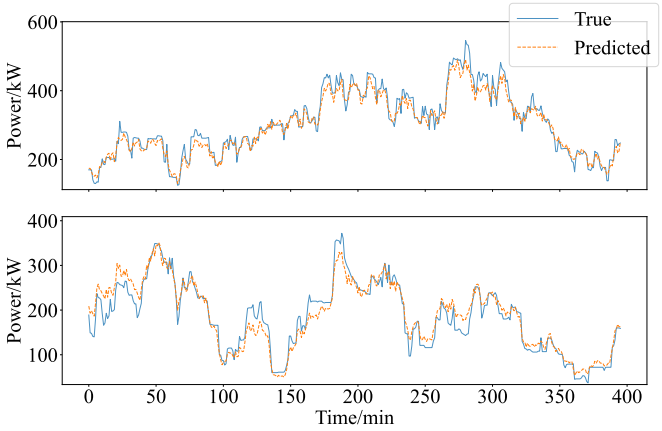

Figure 5: Prediction results for EVCS load using *STGAT + RPFM*.

Based on the numerical results and visualizations, we find that the integration of the proposed RPFM into baseline models, such as STGAT, T-GCN, LSTM, and GRU, consistently improves their performance across all metrics. For instance, STGAT with RPFM achieves a significant reduction in MAE (from 94.10 to 84.64) and RMSE (from 138.89 to 122.55), accompanied by an improvement in accuracy from 0.7052 to 0.7497, representing an 9.38% relative improvement on average. Similarly, T-GCN, LSTM, and GRU show notable performance enhancements, with relative accuracy improvements of 11.48%, 10.30%, and 12.42%, respectively. Among baseline models, STGAT achieves the best performance due to its attention mechanism in identifying important connections in the graph. However, this also contributes to RPFM's relatively low improvement on this baseline model since it has already gained knowledge from the similar way Graphormer does. Yet the RPFM still enhances the STGAT's performance with extra knowledge learned from pretraining.

The consistent performance gains across different architectures highlight the RPFM's generalizability and robustness in modeling EVCS load data, with improvements in accuracy and error metrics underscoring its practical utility in forecasting. These findings suggest that integrating pretrained models like RPFM can significantly boost the performance of predictive frameworks, especially in EVCS management, where accurate short-term load forecasting is essential for optimizing energy allocation and improving operational efficiency.

2) Road Traffic Prediction: In this task, we utilize a traffic speed dataset from Shenzhen, as introduced in [58]. The task involves using data from four consecutive time steps as input to predict the traffic speed for the subsequent time step, adhering to standard practices commonly in traffic prediction research [58, 65]. A comprehensive comparison of our proposed method with other baseline approaches is presented in Table V and Figure 6.

Table V: Performance Comparison: Road Traffic Prediction (156 roads × 2976 time steps).

Method	MAE ↓	RMSE ↓	Accuracy ↑	Improvement ↑
STGAT	2.80	4.27	0.7089	–
+ RPFM	**2.71**	**4.12**	**0.7178**	**2.66%**
T-GCN	2.83	4.42	0.7074	–
+ RPFM	**2.72**	**4.12**	**0.7175**	**4.03%**
LSTM	2.81	4.50	0.7024	–
+ RPFM	**2.79**	**4.19**	**0.7097**	**2.88%**
GRU	2.84	4.51	0.7006	–
+ RPFM	**2.80**	**4.20**	**0.7092**	**3.17%**

As another spatial-temporal prediction task, the numerical results demonstrate trends similar to those observed in the EVCS load prediction task, with consistent improvements in all performance metrics after integrating the pretrained RPFM. Notably, the reduction in RMSE is particularly significant, indicating the model's ability to better capture and predict variations in traffic patterns.

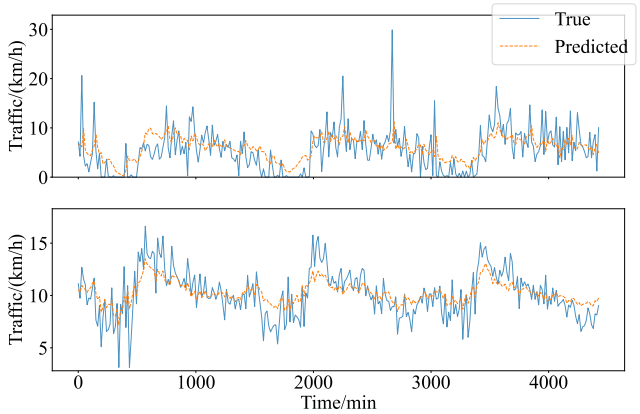

Figure 6: Prediction results of Road Traffic Prediction *STGAT + RPFM*.

The qualitative result shown in Figure 6 indicate that the predicted values, while generally following the trends of the actual values, exhibit deviation in time steps with high traffic variability. Potential explanations for this discrepancy could include insufficient representation of rare or abrupt traffic events in the training data or challenges in modeling highly nonlinear

traffic patterns within short prediction intervals. Addressing these issues through methods such as incorporating additional external features, leveraging more advanced architectures, or employing multi-scale temporal modeling may facilitate the development of more advanced spatial-temporal forecasting models. Nevertheless, these limitations do not undermine the effectiveness of the proposed RPFM, as evidenced by the significant reduction in RMSE and improvements across all metrics, which demonstrate its robust contribution to enhancing baseline model performance. In conclusion, the integration of pretrained RPFM provides measurable performance gains across all baseline models, reaffirming its generalizability and utility in spatial-temporal prediction tasks.

3) EVCS Location Planning: In this task, we utilize an RPCN dataset of Seattle, which is not included in the pretraining dataset. A circular region with an 8 km radius is used as the training dataset, while a ring-shaped region with a radius of 8–10 km serves as the test dataset, as depicted in Figure 7.

Table VI: Performance Comparison: EVCS Location Planning (456 EVCSs, 130K nodes, 400K edges).

Method	Precision ↑	Recall ↑	F1 Score ↑	Improvement ↑
RPFM (0-shot)	**0.7327**	**0.8661**	**0.7938**	–
GCN	0.8729	0.8667	0.8739	–
+ RPFM	**0.9600**	**0.9105**	**0.9346**	**7.33%**
GAT	0.9553	0.8549	0.8941	–
+ RPFM	**0.9899**	**0.9623**	**0.9759**	**8.44%**
GraphSAGE	1.0000	0.8305	0.9074	–
+ RPFM	**1.0000**	**0.9914**	**0.9957**	**9.70%**

Base on the results in Table VI and Figure 7, a key finding is that the pretrained RPFM demonstrates remarkable zero-shot capabilities, achieving an F1 score of 0.7938 without requiring any fine-tuning on the new dataset. This performance highlights the effectiveness of the RPFM's pretraining process, where the model learns generalized relationships between EVCS placements and graph structures from extensive pretraining on massive datasets. When integrated with baseline models such as GCN, GAT, and GraphSAGE, the pretrained RPFM significantly improves performance, improving results by 7.33% over GCN, 8.44% over GAT, and 9.70% over GraphSAGE on average. Notably, the addition of RPFM also enhances precision and recall, with GraphSAGE+RPFM achieving perfect precision (1.0000) and a near-perfect recall of 0.9914.

These findings underscore the versatility and generalizability of the RPFM. The model's ability to transfer knowledge learned from diverse pretraining data and apply it effectively to new graph structures demonstrates its robustness in capturing complex and heterogeneous spatial relationships.

In conclusion, the results highlight the effectiveness of the RPFM in EVCS location planning, both as a standalone zero-shot model and as an enhancement to the existing graph-based frameworks.

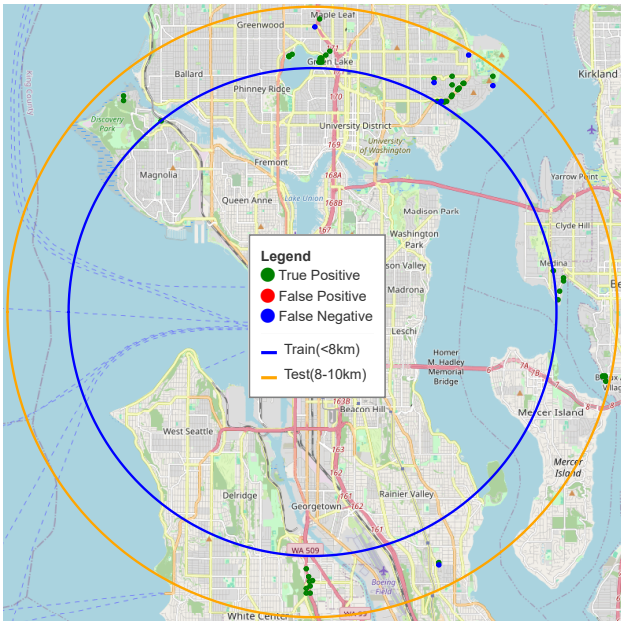

Figure 7: Prediction results of EVCS Location Planning *GraphSAGE + RPFM*.

F. Discussion

The results from our experiments highlight the significance and implications of the proposed Road Power Foundation Model (RPFM). Conceptually, RPFM's success across multiple tasks demonstrates the viability of foundation models in RPCN research. By delivering consistent gains in EVCS load prediction, road traffic prediction, and EVCS location planning, it highlights the value of a unified design framework that addresses interrelated RPCN dynamics. These findings show how a shared foundation model can promote consistency and transferability in spatial-temporal prediction and planning tasks. Methodologically, RPFM's pretraining approach illuminates the link between graph structure and EVCS placements. In spatiotemporal prediction tasks, pretraining ensures that the model can generalize power consumption or traffic patterns based on graph connectivity and localized dependencies. In EVCS location planning, pretraining facilitates the model's ability to predict optimal placements by leveraging its understanding of how graph structure influences network functionality. This foundational framework learns correlations among different graph structures and various EVCS features within RPCNs, enabling the model to adapt effectively to diverse RPCN tasks. Practically, RPFM's adaptability enhances performance across diverse RPCN applications and provides a foundation for broader integration. Its consistent gains in various tasks highlight its ability to address varied challenges while remaining scalable. Moreover, its versatility suggests potential integration with other RPCN tasks, positioning RPFM as a robust solution for both prediction and optimization.

V. Conclusion

This paper introduces **RoadPowerFM (RPFM)**, a foundation model for Road-Power Coupling Networks (RPCNs) based on the Graphormer and JEPA architectures. RPFM

leverages pretraining on large-scale graphs to capture both global and local dependencies, enabling generalization across diverse tasks. Extensive experiments demonstrate the model's strong performance in EVCS load prediction, road traffic prediction, and EVCS location planning. This work reinforces the significance of foundation models in RPCN research. First, RPFM validates the viability of using foundation models as a unified backbone to address prediction, optimization, and planning tasks in interconnected RPCNs. Second, the pretraining methodology ensures highly transferable graph representations, significantly reducing the reliance on task-specific data and enhancing the applicability of the model to scenarios with limited labeled data. Third, the incorporation of community-based graph partitioning enables RPFM to scale effectively to large graphs, ensuring the model captures both localized dependencies and global structural insights. This dual capability is crucial for accurate spatial-temporal modeling, demonstrating the model's versatility and practical relevance for diverse applications.

In summary, RPFM is a robust and flexible framework for addressing the challenges of RPCNs. Its transferability, adaptability, and scalability bridge the gap between spatial-temporal forecasting and strategic optimization, enabling scalable, generalizable solutions for complex urban energy and transportation systems. Future research could explore advanced pretraining methods incorporating multimodal data from power and urban systems, refine fine-tuning approaches, and extend RPFM's application to areas like renewable energy systems and urban resilience planning.

APPENDIX A
METRICS

(1) **RMSE**

$$RMSE = \sqrt{\frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}(l_i^j - \hat{l}_i^j)^2} \tag{A1}$$

(2) **MAE**

$$MAE = \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}\left|l_i^j - \hat{l}_i^j\right| \tag{A2}$$

(3) **Accuracy**

$$Accuracy = 1 - \frac{\|L - \hat{L}\|_F}{\|L\|_F} \tag{A3}$$

where l_i^j and $\hat{l}_i^j$ represent the ground truth and the prediction of jth time sample in the ith node or edge, M is the number of time samples, N is the number of nodes or edges, and L and $\hat{L}$ represent the set of l_i^j and $\hat{l}_i^j$, respectively.

(4) **Precision**

$$\text{Precision} = \frac{TP}{TP + FP} \tag{A4}$$

(5) **Recall**

$$\text{Recall} = \frac{TP}{TP + FN} \tag{A5}$$

(6) **F1 Score**

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{A6}$$

Their representation can be explained by Table AI.

Table AI: **Confusion Matrix**

	Positive_Pred	Negative_Pred
Positive_True	TP	FN
Negative_True	FP	TN

ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China 2021YFB2401203.

REFERENCES

[1] X. He, Q. Ai, J. Wang, F. Tao, B. Pan, R. Qiu, and B. Yang, "Situation awareness of energy internet of things in smart city based on digital twin: From digitization to informatization," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7439–7458, 2023.

[2] X. He, Y. Tang, S. Ma, Q. Ai, F. Tao, and R. Qiu, "Redefinition of digital twin and its situation awareness framework designing toward fourth paradigm for energy internet of things," *IEEE Trans. Syst., Man, Cybern.*, vol. 54, no. 11, pp. 6873–6888, 2024.

[3] L. N. Moghanlou, F. Di Maio, and E. Zio, "Probabilistic scenario analysis of integrated road-power infrastructures with hybrid fleets of evs and icvs," *Reliab. Eng. Syst. Saf.*, vol. 242, p. 109712, 2024.

[4] M. Nour, J. P. Chaves-Ávila, G. Magdy, and Á. Sánchez-Miralles, "Review of positive and negative impacts of electric vehicles charging on electric power systems," *Energies*, vol. 13, no. 18, p. 4675, 2020.

[5] W. Wei, W. Danman, W. Qiuwei, M. Shafie-Khah, and J. P. Catalão, "Interdependence between transportation system and power distribution system: A comprehensive review on models and applications," *J. Mod. Power Syst. Clean Energy*, vol. 7, no. 3, pp. 433–448, 2019.

[6] N. B. M. Shariff, M. Al Essa, and L. Cipcigan, "Probabilistic analysis of electric vehicles charging load impact on residential distributions networks," in *2016 IEEE Int. Energy Conf. (ENERGYCON)*. IEEE, 2016, pp. 1–6.

[7] S. Pourazarm and C. G. Cassandras, "Optimal routing of energy-aware vehicles in transportation networks with inhomogeneous charging nodes," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2515–2527, 2017.

[8] H. Yao, Y. Xiang, C. Gu, and J. Liu, "Optimal planning of distribution systems and charging stations considering pv-grid-ev transactions," *IEEE Trans. Smart Grid*, 2024.

[9] H. Pourvaziri, H. Sarhadi, N. Azad, H. Afshari, and M. Taghavi, "Planning of electric vehicle charging stations: An integrated deep learning and queueing theory approach," *Transp. Res. Part E: Logist. Transp. Rev.*, vol. 186, p. 103568, 2024.

[10] S. Çelik and Ş. Ok, "Electric vehicle charging stations: Model, algorithm, simulation, location, and capacity planning," *Heliyon*, vol. 10, no. 7, 2024.

[11] K. Nareshkumar and D. Das, "Optimal location and sizing of electric vehicles charging stations and renewable sources in a coupled transportation-power distribution network," *Renew. Sustain. Energy Rev.*, vol. 203, p. 114767, 2024.

[12] J. Li, X. Xu, Z. Yan, H. Wang, M. Shahidehpour, B. Xie, and X. Luo, "Resilient resource allocations for multi-stage transportation-power distribution system operations in hurricanes," *IEEE Trans. Smart Grid*, 2024.

[13] H. Zhang, S. J. Moura, Z. Hu, and Y. Song, "Pev fast-charging station siting and sizing on coupled transportation and power networks," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2595–2605, 2016.

[14] T. Qian, C. Shao, X. Li, X. Wang, and M. Shahidehpour, "Enhanced coordinated operations of electric power and transportation networks via ev charging services," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3019–3030, 2020.

[15] Q. Yuan, Y. Ye, Y. Tang, X. Liu, and Q. Tian, "Low carbon electric vehicle charging coordination in coupled transportation and power networks," *IEEE Trans. Ind. Appl.*, vol. 59, no. 2, pp. 2162–2172, 2022.

[16] Q.-C. Lu, S. Wang, P.-C. Xu, J. Li, X. Meng, and A. Hussain, "Modeling the dependency relationship of coupled power and transportation networks," *Energy*, p. 135330, 2025.

[17] T. Zhao, H. Yan, X. Liu, and Z. Ding, "Congestion-aware dynamic optimal traffic power flow in coupled transportation power systems," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1833–1843, 2022.

[18] S. Lv, Z. Wei, S. Chen, G. Sun, and D. Wang, "Integrated demand response for congestion alleviation in coupled power and transportation networks," *Applied Energy*, vol. 283, p. 116206, 2021.

[19] W. Wei, L. Wu, J. Wang, and S. Mei, "Network equilibrium of coupled transportation and power distribution systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6764–6779, 2017.

[20] K. Park and I. Moon, "Multi-agent deep reinforcement learning approach for ev charging scheduling in a smart grid," *Appl. Energy*, vol. 328, p. 120111, 2022.

[21] J. Dong, A. Yassine, A. Armitage, and M. S. Hossain, "Multi-agent reinforcement learning for intelligent v2g integration in future transportation systems," *IEEE Trans. Intell. Transp. Syst.*, 2023.

[22] W. Pan, X. Yu, Z. Guo, T. Qian, and Y. Li, "Online evs vehicle-to-grid scheduling coordinated with multi-energy microgrids: A deep reinforcement learning-based approach," *Energies*, vol. 17, no. 11, p. 2491, 2024.

[23] Y. Li, S. Su, M. Zhang, Q. Liu, X. Nie, M. Xia, and D. D. Micu, "Multi-agent graph reinforcement learning method for electric vehicle on-route charging guidance in coupled transportation electrification," *IEEE Trans. Sustainable Energy*, 2023.

[24] Y. Wang, D. Qiu, Y. He, Q. Zhou, and G. Strbac, "Multi-agent reinforcement learning for electric vehicle decarbonized routing and scheduling," *Energy*, vol. 284, p. 129335, 2023.

[25] J. Topić, B. Škugor, and J. Deur, "Neural network-based modeling of electric vehicle energy demand and all electric range," *Energies*, vol. 12, no. 7, p. 1396, 2019.

[26] S. Su, Y. Li, Q. Chen, M. Xia, K. Yamashita, and J. Jurasz, "Operating status prediction model at ev charging stations with fusing spatiotemporal graph convolutional network," *IEEE Trans. Transp. Electrification*, vol. 9, no. 1, pp. 114–129, 2022.

[27] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[28] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, pp. 1–45, 2024.

[29] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting, "Large language models in medicine," *Nature Medicine*, vol. 29, no. 8, pp. 1930–1940, 2023.

[30] Y. Tang, Z. Wang, A. Qu, Y. Yan, Z. Wu, D. Zhuang, J. Kai, K. Hou, X. Guo, J. Zhao *et al.*, "Itinera: Integrating spatial optimization with large language models for open-domain urban itinerary planning," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2024, pp. 1413–1432.

[31] M. Moor, O. Banerjee, Z. S. H. Abad, H. M. Krumholz, J. Leskovec, E. J. Topol, and P. Rajpurkar, "Foundation models for generalist medical artificial intelligence," *Nature*, vol. 616, no. 7956, pp. 259–265, 2023.

[32] M. Awais, M. Naseer, S. Khan, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, and F. S. Khan, "Foundation models defining a new era in vision: a survey and outlook," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025.

[33] X. Zhu, X. Yang, Z. Wang, H. Li, W. Dou, J. Ge, L. Lu, Y. Qiao, and J. Dai, "Parameter-inverted image pyramid networks," in *NeurIPS*, 2024.

[34] C. Huang, S. Li, R. Liu, H. Wang, and Y. Chen, "Large foundation models for power systems," in *2024 IEEE Power & Energy Soc. Gen. Meet. (PESGM)*. IEEE, 2024, pp. 1–5.

[35] A. Vaswani, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, 2017.

[36] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, pp. 111–132, 2022.

[37] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, 2020, pp. 2704–2710.

[38] D. Cai and W. Lam, "Graph transformer for graph-to-sequence learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 05, 2020, pp. 7464–7471.

[39] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for

graph representation?" *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 28 877–28 888, 2021.

[40] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12 939–12 948.

[41] X. Chang, J. Wang, M. Wen, Y. Wang, and Y. Huang, "M-graphormer: Multi-channel graph transformer for node representation learning," *IEEE Trans. Big Data*, 2024.

[42] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, 2021.

[43] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 15 619–15 629.

[44] G. Skenderi, H. Li, J. Tang, and M. Cristani, "Graph-level representation learning with joint-embedding predictive architectures," *arXiv preprint arXiv:2309.16014*, 2023.

[45] J. Xu, X. Sun, Z. Zhang, G. Zhao, and J. Lin, "Understanding and improving layer normalization," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[47] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183–197, 1991.

[48] M. M. Keikha, M. Rahgozar, and M. Asadpour, "Community aware random walk for network embedding," *Knowl. Based Syst.*, vol. 148, pp. 47–54, 2018.

[49] Z. Huang, A. Silva, and A. Singh, "A broader picture of random-walk based graph embedding," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 685–695.

[50] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theor. Exp.*, vol. 2008, no. 10, p. P10008, 2008.

[51] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[52] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[53] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2147–2154.

[54] S. Mo and P. Tong, "Connecting joint-embedding predictive architecture with contrastive self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 2348–2377, 2024.

[55] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *ICLR*, 2022.

[56] A. Graves and A. Graves, "Long short-term memory," *Supervised Sequence Labelling with Recurrent Neural Networks*, pp. 37–45, 2012.

[57] Z. Bai, J. Wang, X.-L. Zhang, and J. Chen, "End-to-end speaker verification via curriculum bipartite ranking weighted binary cross-entropy," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 1330–1344, 2022.

[58] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, 2019.

[59] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (gru) neural networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2017, pp. 1597–1600.

[60] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, and W. Lu, "Stgat: Spatial-temporal graph attention networks for traffic flow forecasting," *IEEE Access*, vol. 8, pp. 134 363–134 372, 2020.

[61] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Int. Conf. Mach. Learn.* PMLR, 2019, pp. 6861–6871.

[62] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[63] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[64] D. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[65] Y. Tang, A. Qu, A. H. Chow, W. H. Lam, S. C. Wong, and W. Ma, "Domain adversarial spatial-temporal network: A transferable framework for short-term traffic forecasting across cities," in *Proc. 31st ACM Int. Conf. Inf. & Knowledge Management*, 2022, pp. 1905–1915.