Tracking Topological Shifts: How Can Dynamic Graph Invariant Learning Enable Reliable Out-of-Time Spatio-Temporal Prediction?

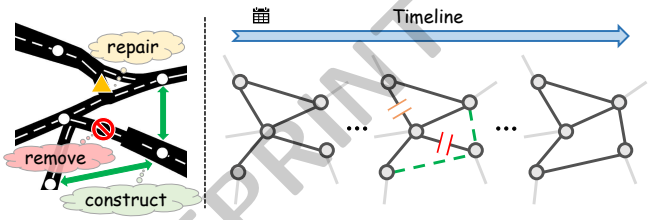
Xinyan Hao[1], **Huaiyu Wan**[1,2*], **Shengnan Guo**[1,2], **Shaojiang Wang**[3,4*] and **Youfang Lin**[1,2]
[1]School of Computer Science and Technology, Beijing Jiaotong University, Beijing, China
[2]Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, Beijing, China
[3]Institute of AI for Industries, Chinese Academy of Sciences, Nanjing, China
[4]Nanjing Institute of Software Technology, Nanjing, China
{xinyanhao, hywan, guoshn}@bjtu.edu.cn, wangshaojiang@iaii.ac.cn, yflin@bjtu.edu.cn

Abstract

Spatio-temporal graph networks form the foundation of modern traffic prediction, yet their deployment is fundamentally challenged by the pervasive reality of distribution shifts. While out-of-distribution (OOD) learning holds promise for robustness, existing methods rely on static graph structures, failing to capture the inherent topological dynamics of real-world traffic systems and thus limiting long-term deployment reliability. To bridge this gap, we propose DynaSTar, a Dynamic Spatio-Temporal Graph Invariant Learning model designed for reliable out-of-time (OOT) traffic prediction under evolving topologies. Our model employs a dynamic probabilistic graph structure, which is continuously refined through momentum-based updates and differentiable sparse sampling to model evolving inter-node dependencies. Besides, it utilizes node-level environment construction and modulated prediction to extract representations invariant to heterogeneous neighborhood fluctuations, enabling robust generalization. Comprehensive experiments on large-scale, long-term real-world datasets demonstrate that by effectively tracking evolving topological shifts, DynaSTar consistently outperforms state-of-the-art baselines across various OOT scenarios.

1 Introduction

With the development of smart cities, spatio-temporal prediction has become a critical pillar for urban management and planning [Lv *et al.*, 2014; Liang *et al.*, 2021]. However, underlying traffic generation mechanisms evolve continuously over time, driven by factors such as the opening of new subway lines or shifts in commercial hubs [Wang *et al.*, 2023]. This phenomenon directly violates the core assumption of independent and identically distributed (i.i.d.) training and test data in traditional spatio-temporal models, resulting in a decline in predictive performance when deployed in an evolved real-world environment. Therefore, developing robust methods capable of adapting to such distribution shifts is essential for ensuring the long-term reliability of models in dynamic urban systems.



(a) Various incidents of road network (b) Dynamic evolution of graph topology

Figure 1: Evolution of graph topology driven by traffic incidents

To address this problem, researchers have increasingly focused on out-of-distribution (OOD) learning paradigms, which enhance model robustness across diverse and shifting data distributions [Li *et al.*, 2025]. These methods aim to uncover underlying stable causal mechanisms or invariant relationships from observed data under varying conditions. Strategies include manually constructing causal models [Xia *et al.*, 2023; Zhou *et al.*, 2023], using environments as self-supervision signals to disentangle confounding factors [Ji *et al.*, 2025], or augmenting environment diversity via graph perturbations [Wang *et al.*, 2024]. By extracting robust knowledge, these approaches improve model generalization to new distributions without requiring additional training.

Although previous work has achieved significant progress in spatio-temporal prediction under distribution shift, they typically assume shift factors are independent to the passage of time, such as weather or accidents. However, in non-stationary systems like traffic, shifts often occur systematically as time progresses, for instance, the persistent topological dynamics (see Fig. 1(b)) induced by real-world changes (see Fig. 1(a)), which continuously introduce new and unforeseen environments associated with the progression of time. This represents a temporal evolution of topology. Therefore, addressing such shifts that are inherent to the passage of time, referred to as *out-of-time (OOT) generalization*, is essential for a model's ability to maintain performance when deployed across future unseen time periods. Existing methods typically learn from static graph structures, overlooking gradual, long-term topological evolution. **To overcome this, a model must explicitly capture topological dy-**

*Corresponding authors.

namics and track the graph's own evolution.

However, two major challenges must be addressed to achieve this goal. **First, how to construct a dynamic graph adaptable to continuous evolution and OOT generalization?** Existing work often rely on ahistorical, fully-connected, and deterministic graph structures. However, graph evolution is gradual, not instantaneous, such as the strengthened correlations caused by the opening of a new subway line. Moreover, fully-connected graphs introduce spurious dependencies that obscure robust spatial patterns [Ju *et al.*, 2024], while deterministic graphs cannot adequately model the uncertainty and temporariness of structural changes caused by unforeseen events. **Second, how to learn node-level knowledge that is invariant to dynamic topological shifts?** Existing methods often apply random interventions to the global graph to create training environments, ignoring the heterogeneous and node-specific nature of topological change. For example, connections at an accident-prone intersection are more likely to toggle abruptly than those along a stable one-way arterial. Additionally, using a globally shared prediction function can hinder the learning of node-specific invariant patterns, ultimately limiting OOT generalization.

To address the aforementioned challenges, this paper introduces ***DynaSTar***, a Dynamic Spatio Temporal Graph Invariant learning model for robust traffic prediction. To track dynamic topology, we employ a dynamic probabilistic graph structure, where each edge is parameterized as a Bernoulli distribution whose parameters evolve via a momentum-based update rule. This design ensures the graph reflects gradual, historically-grounded structural changes. A sparse sampling mechanism is then applied to instantiate a clear, sparse adjacency matrix at each step, reducing spurious edges and enhancing robustness. Next, to extract representations robust to heterogeneous node conditions, we propose neighborhood environment set construction to simulates diverse, plausible topological shift scenarios for each node. For prediction, we use a modulated predictor that fuses node-conditioned vectors with a global predictor, allowing the model to learn both locally stable and globally transferable patterns. Finally, guided by multi-objective optimization, the model jointly integrates accurate prediction, and node-heterogeneous invariant knowledge, improving its OOT generalization. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, this work presents the first formal investigation that explicitly models topological dynamics to address OOT generalization in spatio-temporal traffic prediction, distinguishing it from conventional OOD scenarios.
- We propose **DynaSTar**, a novel framework that integrates dynamic topology tracking with node-heterogeneous invariant learning for robust OOT spatio-temporal prediction. Our model employs a evolving dynamic graph coupled with a node-specific environment construction and modulation mechanism to capture topological shifts and extract heterogeneous invariant representations.
- Extensive experiments on two large-scale, long-term real-world datasets demonstrate DynaSTar's superior OOT generalization performance across various scenarios with dynamic topologies. The code is made publicly available at https://github.com/ShaunHao/dynastar.

2 Related Work

Accurate spatio-temporal prediction is crucial for intelligent transportation systems in various tasks, such as urban flow [Gong *et al.*, 2022; Qu *et al.*, 2023; Lv *et al.*, 2026], traffic speed [Weng *et al.*, 2023; Weng *et al.*, 2025a; Huang *et al.*, 2025], and travel time [Shen *et al.*, 2025]. However, distributional inconsistency in urban systems poses a cross-scenario generalization challenge for models trained under the i.i.d. assumption. To enhance models' generalization, research focuses on learning cross-scenario invariant knowledge. Knowledge transfer methods [Jin *et al.*, 2022; Yuan *et al.*, 2024; Li *et al.*, 2024; Hao *et al.*, 2025] learn city-level generalizable patterns, yet overlook temporal distribution shifts within cities. OOD learning methods, which use prior knowledge to simulate test distributions during training, are gaining traction. Temporal OOD methods [Du *et al.*, 2021; Lu *et al.*, 2022] employ unsupervised domain generalization, while spatio-temporal OOD techniques leverage structural causal models to learn generalizable representations. Among them, [Xia *et al.*, 2023] for causal data generation, [Zhou *et al.*, 2023] for invariant relations, [Ji *et al.*, 2025] for disentangling confounders, and [Wang *et al.*, 2024] for semantic graph interventions. However, current research primarily models distribution shifts using static adjacency relations, overlooking inherent topological dynamics and structural evolution. This causes learned knowledge to degrade over time, impairing OOT generalization.

3 Preliminaries

Definition 1 (Temporal Phase)**.** *To analyze distribution shifts after model deployments, we partition the continuous timeline into non-overlapping intervals (e.g., several months or a year) called temporal phases. Let $\mathcal{P}_k$ denote the k-th phase, containing timestamps $\{t_1^k, \ldots, t_{L_k}^k\}$. Data distributions and topologies may shift across phases.*

Definition 2 (Spatio-Temporal Graph)**.** *At time t, the traffic network is represented as a graph $\mathcal{G}_t = (\mathcal{V}, \mathbf{X}_t, \mathbf{A}_t)$. $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the node set. $\mathbf{X}_t \in \mathbb{R}^{N \times C}$ denotes node features (e.g., traffic flow). $\mathbf{A}_t \in \mathbb{R}^{N \times N}$ is a adjacency matrix encoding inter-node relationships.*

Definition 3 (Spatio-Temporal Prediction under Dynamic Topology)**.** *Given a historical window $\mathbf{X}_{[t-T_h+1:t]}$ and corresponding adjacency matrices $\mathbf{A}_{[t-T_h+1:t]}$, the goal is to learn a predition function F_Θ to predict future states:*

$$\hat{\mathbf{Y}}_{[t+1:t+T_f]} = F_\Theta\left(\mathbf{X}_{[t-T_h+1:t]}, \mathbf{A}_{[t-T_h+1:t]}\right).$$

Definition 4 (Long-Term Deployment Spatio-Temporal Prediction Learning)**.** *In real-world deployment, the model encounters unseen temporal phases $\{\mathcal{P}_{M+1}, \ldots\}$. Both the data distribution $P(\mathbf{X}, \mathbf{Y})$ and the graph structure $\mathbf{A}_t$ may shift across phases. The objective is to train a model on historical phases $\{\mathcal{P}_1, \ldots, \mathcal{P}_M\}$ that minimizes the expected prediction*

error on all future phases, thereby requiring OOT generalization under topological dynamics.

4 Methodology

This section details our DynaSTar model depicted in Fig. 2, which first extracts an evolving dynamic graph from traffic data, then performs node-specific future prediction. Joint training enables learning invariant predictive knowledge for topological shifts. The key designs are described below.

4.1 Dynamic Topological Shifts Tracking

Modeling a graph structure that reflects the continuous evolution of topologies is a prerequisite for reliable long-term deployment of traffic prediction. To achieve this, we first compute a dynamic correlation graph to quantify the real-time linkage strength between nodes. Next, we introduce a momentum mechanism to update the graph, smooth the transition of topological changes, and then transformed it into a dynamic probabilistic graph prototype. Finally, a sparse binary adjacency matrix is sampled from the updated graph via reparameterization trick for prediction learning.

Memory-Enhanced Dynamic Correlation Graph Construction

To capture the time-varying patterns of each node, we first encode raw traffic data into dynamic representations. Given input $\mathbf{X}_{[t-T_h+1:t]} \in \mathbb{R}^{N\times T_h\times C}$ at time t, we use L stacked spatio-temporal attention blocks, which can be formulated as

$$\begin{aligned}\mathbf{H}^{(0)} &= \mathbf{X}_{[t-T_h+1:t]},\\ \mathbf{H}^{(l)} &= \mathrm{SA}\big(\mathrm{TA}\big(\mathbf{H}^{(l-1)}\big), \mathbf{A}^{\mathrm{ori}}\big)(l=1,\dots,L),\end{aligned} \tag{1}$$

where SA denotes a graph attention with the original adjacency matrix [Veličković *et al.*, 2018] and TA denotes a windowed temporal attention [Cirstea *et al.*, 2022]. Each block's output is projected by a fully-connected (FC) layer, then concatenated to form the final dynamic representation $\mathbf{Z}_{(t)} \in \mathbb{R}^{N\times d}$, synthesizing multi-scale information for robust graph inference

$$\mathbf{Z}_{(t)} = \mathrm{Concat}\left(\mathbf{W}^{(1)}\mathbf{H}^{(1)}, \mathbf{W}^{(2)}\mathbf{H}^{(2)}, \dots, \mathbf{W}^{(L)}\mathbf{H}^{(L)}\right). \tag{2}$$

To distill a more purified signal for graph generation from $\mathbf{Z}_{(t)}$, we use a memory-enhanced module that projects nodes into a latent space of relational prototypes [Weng *et al.*, 2025b]. Using a learnable meta memory bank $\mathcal{M} \in \mathbb{R}^{M\times d_m}$, each node's representation $\mathbf{z}_i^t$ queries it via cross-attention, yielding a topology-refined graph-generating embedding $\mathbf{g}_i$:

$$\mathrm{g}_i^t = \textstyle\sum_{p=1}^{M} \alpha_{i,p}^t (\mathbf{m}_p \mathbf{W}_V), \alpha_{i,p}^t = \frac{\exp(\mathbf{z}_i^t \mathbf{W}_Q (\mathbf{m}_p \mathbf{W}_K)^\top/\sqrt{d})}{\sum_{k=1}^{M} \exp(\mathbf{z}_i^t \mathbf{W}_Q (\mathbf{m}_k \mathbf{W}_K)^\top/\sqrt{d})}, \tag{3}$$

where $\mathbf{m}_p$ is the p-th memory vector, and $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are projection matrices. Finally, the dynamic correlation score between node i and node j at time t is computed from their graph generation embeddings:

$$s_{ij}^t = (\mathbf{g}_i^t \mathbf{W}_\phi)(\mathbf{g}_j^t \mathbf{W}_\psi)^\top, \tag{4}$$

where $\mathbf{W}_\phi, \mathbf{W}_\psi$ are two FC operation. The collection of s_{ij}^t for all node pairs forms a dynamic correlation graph $\mathcal{G}_p^t$, which reflects the global real-time linkage strength.

Momentum-based Graph Updating and Probabilistic Modeling

The instantaneous graph $\mathcal{G}_p^t$ lacks historical memory, making it noise-sensitive. Inspired by [Ioffe, 2015], We maintain a momentum-updated graph prototype $\mathcal{G}_{proto}$ to ensure smooth and historically grounded evolution, accumulating knowledge from previous graphs:

$$\mathcal{G}_{proto}^t \leftarrow \beta \cdot \mathcal{G}_{proto}^{t-1} + (1-\beta)\cdot \mathcal{G}_p^t, \tag{5}$$

where $\beta \in (0,1)$ is a momentum coefficient. We train on sequential data partitions, updating the graph prototype per batch and resetting it only after the final partition to simulate continuous learning. During inference, the prototype updates after each prediction, integrating current dependencies with long-term trends for stable, adaptive deployment.

To model the fluctuations of inter-node dependencies, each edge is treated as a random variable. Specifically, the graph prototype $\mathcal{G}_{proto}^t$ is transformed into a probabilistic one $\widetilde{\mathcal{A}}^t$ via sigmoid:

$$\widetilde{\mathcal{A}}_{ij}^t = \sigma(\mathcal{G}_{proto,ij}^t), \tag{6}$$

where $\widetilde{\mathcal{A}}_{ij}^t \in (0,1)$ is the probability of an edge from node i to j. The probabilistic graph $\widetilde{\mathcal{A}}^t$ models temporary structural fluctuations, providing an dynamic topological foundation for node-level invariant learning.

Differentiable Dynamic Sparse Graph Generation

From the fully-connected probabilistic prototype $\widetilde{\mathcal{A}}$ (omit the time index t for simplicity), we sample a sparse graph instance using Straight-Through Gumbel-Softmax reparameterization [Liu *et al.*, 2022], enabling efficient and robust prediction within gradient-based learning. Firstly, we apply the Gumbel-Softmax function [Jang *et al.*, 2016] to each edge, obtaining a continuous and differentiable relaxation $\bar{A}_{ij}$ of it:

$$\bar{A}_{ij} = \frac{\exp((\log(\pi_{ij}^1) + \epsilon_{ij}^1)/\tau_g)}{\exp((\log(\pi_{ij}^0) + \epsilon_{ij}^0)/\tau_g) + \exp((\log(\pi_{ij}^1) + \epsilon_{ij}^1)/\tau_g)}. \tag{7}$$

where $\pi_{ij}^1 = \widetilde{\mathcal{A}}_{ij}$, $\pi_{ij}^0 = 1 - \widetilde{\mathcal{A}}_{ij}$, $\epsilon_{ij}^0, \epsilon_{ij}^1 \sim \mathrm{Gumbel}(0,1)$ are i.i.d. Gumbel noise samples, and τ_g is a temperature coefficient. As $\tau \to 0$, $\bar{A}_{ij}$ converges to a one-hot vector where category 1's selection probability equals $\widetilde{\mathcal{A}}_{ij}$, effectively sampling from the Bernoulli distribution. Next, the discrete edge $\hat{A}_{ij}$ is obtained by an *argmax* operation:

$$\hat{A}_{ij} = \mathbb{I}\left[(\log(\pi_{ij}^1) + \epsilon_{ij}^1) > (\log(\pi_{ij}^0) + \epsilon_{ij}^0)\right], \tag{8}$$

where $\mathbb{I}[\cdot]$ is the indicator function. Finally, To allow gradients to flow through this operation, we use the straight-through estimator to obtain a discrete graph $\hat{\mathbf{A}}$ that remains differentiable for backpropagation:

$$\hat{\mathbf{A}} := \bar{\mathbf{A}} + \mathrm{detach}(\hat{\mathbf{A}} - \bar{\mathbf{A}}), \tag{9}$$

where $\mathrm{detach}(\cdot)$ is the stop-gradient operation. This enables gradient-based optimization of the underlying probabilistic graph parameters $\widetilde{\mathcal{A}}$. During inference, we directly threshold the edge probability, yielding a sparse and binary adjacency matrix $\mathbf{A}^{\mathrm{inf}}$ for stable and deterministic predictions:

$$\mathbf{A}_{ij}^{\mathrm{inf}} = \mathbb{I}[\log(\widetilde{\mathcal{A}}_{ij}) > \log(1-\widetilde{\mathcal{A}}_{ij})] = \mathbb{I}[\widetilde{\mathcal{A}}_{ij} > 0.5]. \tag{10}$$

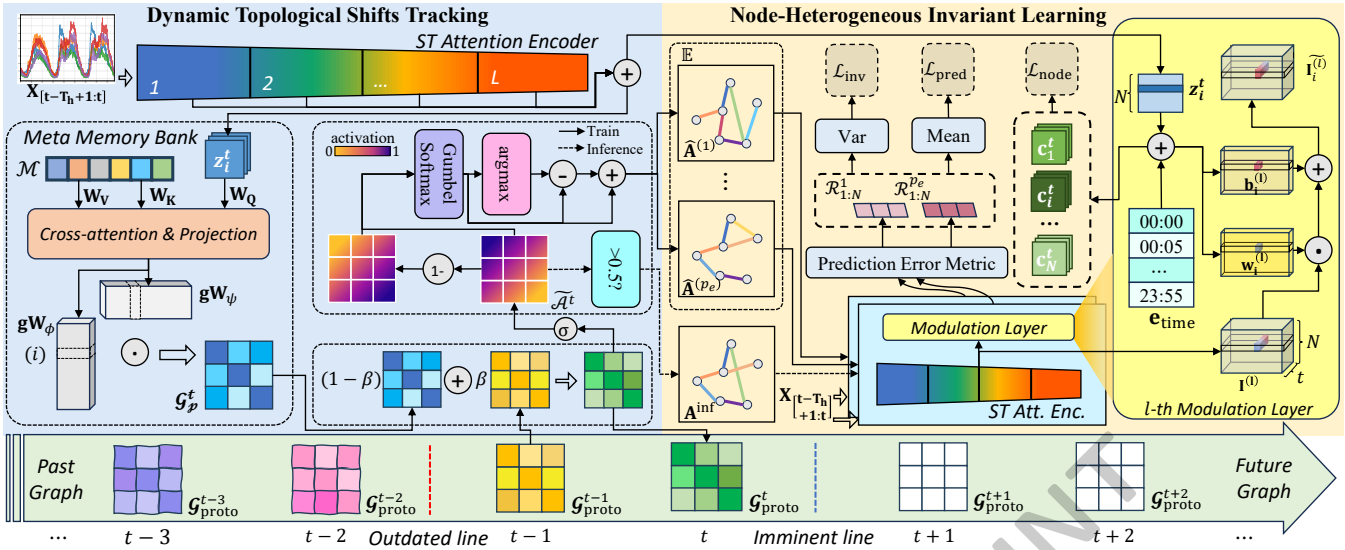


Figure 2: The overview of our DynaSTar model

4.2 Node-Heterogeneous Invariant Learning

Adapting to individual node patterns and extracting their topological-invariant knowledge is critical for accurate prediction in OOT scenarios. To accomplish this, we first construct multiple neighborhood environments per node through dynamic probabilistic graph sampling. Next, we learn node-specific dynamic condition vectors to modulate prediction, customizing unique inference per node while maintaining globally prediction knowledge. Finally, under multi-loss optimization, the model learns stable prediction for OOT generalization.

Node Neighborhood Environment Set Construction

The OOT challenge posed by real-world topological shifts (i.e., random neighborhood compositions) requires modeling edge-level heterogeneity and dynamics. To address this, our dynamic probabilistic graph prototype $\widetilde{\mathcal{A}}^t$ directly learns these properties, rather than relying on global random perturbations. For node i at time t, its neighborhood environment is parameterized by a multivariate Bernoulli distribution as

$$P(\mathcal{N}_i^t) = P\left(\{a_{i1}^t, \ldots, a_{iN}^t, a_{1i}^t, \ldots, a_{Ni}^t\}\right) = \prod_{\substack{j=1 \\ j\neq i}}^{N} (\widetilde{\mathcal{A}}_{ij}^t)^{a_{ij}^t} (1-\widetilde{\mathcal{A}}_{ij}^t)^{1-a_{ij}^t} \cdot \prod_{\substack{j=1 \\ j\neq i}}^{N} (\widetilde{\mathcal{A}}_{ji}^t)^{a_{ji}^t} (1-\widetilde{\mathcal{A}}_{ji}^t)^{1-a_{ji}^t}, \tag{11}$$

where $a_{ij}^t \in \{0,1\}$ is the directed edge from i to j at time t.

Leveraging this time-varying distribution, we generate diverse neighborhood environments per node. For each training batch, we sample the sparse adjacency matrix p_e times via Straight-Through Gumbel-Softmax, yielding a global environment set $\mathbb{E} = \{\hat{\mathbf{A}}^{(1)}, ..., \hat{\mathbf{A}}^{(p_e)}\}$. From node i's perspective, this corresponds to sampling p_e times from its own distribution $P(\mathcal{N}_i^t)$, producing a neighborhood set $\{\mathcal{N}_i^{(1)}, ..., \mathcal{N}_i^{(p_e)}\}$. Training across these diverse sampled environments simulates heterogeneous topological shifts per node, forcing the model to extract node-level invariant predictive patterns across varying neighborhood contexts.

Node-Specific Modulation for Prediction

Conventional models use a globally shared prediction mapping, learning an average pattern that suppresses node-specific signals. To address heterogeneous shifts, we design a node-specific modulation mechanism that disentangles and leverages personalized knowledge for robust prediction.

For each node i, we first derive a dynamic condition vector $\mathbf{c}_i^t$ by fusing its dynamic representation $\mathbf{z}_i^t$ obtained from Section 4.1 with a learnable time-of-day embedding via a Multi-Layer Perceptron (MLP):

$$\mathbf{c}_i^t = \text{MLP}\left(\mathbf{z}_i^t + \mathbf{e}_{\text{time}}(t)\right), \tag{12}$$

where $\mathbf{e}_{\text{time}}(t)$ is the time-of-day embedding. This vector serves as an adaptive code for node-specific prediction conditioning. Next, a modulated predictor integrates node-specific knowledge with globally shared patterns via Feature-wise Linear Modulation [Perez *et al.*, 2018]. Specifically, a modulation generator MLP transforms the condition vector $\mathbf{c}_i^t$ into layer-specific affine parameters $\mathbf{w}_i^{(l)}, \mathbf{b}_i^{(l)}$ for each predictor layer l:

$$\mathbf{w}_i^{(l)}, \mathbf{b}_i^{(l)} = \text{MLP}_{\text{mod}}^{(l)}(\mathbf{c}_i^t). \tag{13}$$

Notably, the predictor shares the same attention block architecture as in Section 4.1, but its intermediate representations $\mathbf{I}^{(l)} = \text{SA}(\text{TA}(\mathbf{I}^{(l-1)}), \hat{\mathbf{A}})$ are modulated by node-specific affine parameters:

$$\widetilde{\mathbf{I}}_i^{(l)} = \mathbf{w}_i^{(l)} \odot \mathbf{I}^{(l)} + \mathbf{b}_i^{(l)}, \tag{14}$$

where $\odot$ denotes element-wise multiplication. Finally, the modulated outputs from all layers are projected and aggregated to produce the final prediction for node i:

$$\hat{\mathbf{Y}}_i = \mathbf{W}_O \left(\text{Concat}(\mathbf{W}^{(1)}\widetilde{\mathbf{I}}_i^{(1)}, \ldots, \mathbf{W}^{(L)}\widetilde{\mathbf{I}}_i^{(L)})\right). \tag{15}$$

This design decouples predictor into shared and node-specific components, enabling conditional mapping per node to handle heterogeneous shifts.

Node-Heterogeneous Model Optimization

To refine condition vectors and enhance their discriminative power, we introduce a node-heterogeneity contrastive loss. This enforces semantic similarity for vectors from the same node across time, and distinctiveness for vectors from different nodes. Formally, within a batch, condition vector pairs from the same node at different times are positive samples, whereas pairs from different nodes are negatives. We optimize an InfoNCE-based [Oord *et al.*, 2018] loss:

$$\mathcal{L}_{\text{node}} = \frac{1}{N|\mathcal{T}_i|} \sum_{i=1, t\in\mathcal{T}_i}^{N} \mathcal{L}_{\text{node}}^{(i)}, \tag{16}$$

where $\mathcal{L}_{\text{node}}^{(i)} = \log \frac{-\sum_{t'\in\mathcal{T}_i\setminus\{t\}} \exp(\text{sim}(\mathbf{c}_i^t, \mathbf{c}_i^{t'})/\tau)}{\sum_{j=1,t''\in\mathcal{T}_j}^{N} \mathbb{I}_{[j\neq i \text{ or } t''\neq t]} \exp(\text{sim}(\mathbf{c}_i^t, \mathbf{c}_j^{t''})/\tau)}$, $\text{sim}(\cdot,\cdot)$ is cosine similarity, τ is temperature, and $\mathcal{T}_i$ are timestamps for node i. This loss distills stable node-specific vectors for effective personalized modulation. Furthermore, to ensure robustness against heterogeneous topological shifts, we enforce a node-level invariance constraint with the simulated neighborhood environments $\mathcal{E}_i = \{\mathcal{N}_i^{(1)}, \ldots, \mathcal{N}_i^{(p_e)}\}$ [Arjovsky *et al.*, 2019]. Let Φ represent the modulated predictor, the predictive risk for node i in a specific sampled environment $e \in \mathcal{E}_i$ is:

$$\mathcal{R}_i^e(\Phi) = \ell\left(\Phi(\mathbf{X}; \mathbf{c}_i; e), \mathbf{Y}\right), \tag{17}$$

where $\ell(\cdot)$ is a prediction error metric (e.g., MAE). The node-level invariant loss aims to penalize variance in predictive risk across environments:

$$\mathcal{L}_{\text{inv}} = \frac{1}{N} \sum_{i=1}^{N} \text{Var}_{e\in\mathcal{E}_i}\left(\mathcal{R}_i^e(\Phi)\right), \tag{18}$$

where $\text{Var}(\cdot)$ computes risk variance. This loss drives the model to extract stable and invariant predictive patterns from each node's traffic, regardless of fluctuating connectivity.

4.3 Model Training

The training process integrates the previously described components and objectives into a coherent end-to-end learning framework. The primary optimization target is the mean prediction loss, aggregated over all p_e sampled graph environments as

$$\mathcal{L}_{\text{pred}} = \frac{1}{N \cdot p_e} \sum_{i=1, e\in\mathcal{E}_i}^{N} \mathcal{R}_i^e(\Phi). \tag{19}$$

The overall training objective integrates the prediction loss with the proposed auxiliary losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pred}} + \gamma_1 \mathcal{L}_{\text{node}} + \gamma_2 \mathcal{L}_{\text{inv}}, \tag{20}$$

where coefficients γ_1, γ_2 control the contributions of each loss term. This multi-objective optimization jointly ensures accurate prediction, distinctive node representation, and robustness to topological shifts.

5 Experiments

5.1 Experiment Settings

Datasets

We employ two real-world traffic datasets as benchmarks: the San Diego (SD) and the South Greater Bay Area (SGBA) datasets, to evaluate the robustness of DynaSTar model in dynamic and evolving traffic prediction scenarios. These datasets are subsets of the LargeST dataset [Liu *et al.*, 2023], recording traffic flow data collected at 5-minute intervals by thousands of sensors over a two-year period from 2019 to 2020. The SD dataset comprises 716 sensors, while the SGBA dataset includes 1278 sensors from several adjacent counties in the southern part of the Greater Bay Area (i.e., Alameda, San Francisco, Santa Clara, San Mateo and San Benito). Their static spatial graphs are constructed based on the travel distance between sensors.

Task Settings

Due to the distribution shifts introduced by continuously evolving topology, the characteristics of the shift differ across temporal phases. For prediction models deployed long-term, performance across various phases is crucial, not just during a single specific period. Therefore, we adopt two task settings, near-term and long-term, to comprehensively assess the stability of our model's performance in OOT scenarios at different stages post-deployment. For both the SD and SGBA datasets, we use the 2019 data for model training, splitting it into training and validation sets with a 7:3 ratio. The test set is drawn from the 2020 data of each dataset. Predictions for the first half of the year (January 1st to June 30th) represent the near-term OOT scenario after deployment, while the second half (July 1st to December 31st) is used to test the performance after long-term evolution. Additionally, Z-Score normalization is applied for data preprocessing.

Baselines

We compare the performance of DynaSTar with a series of prediction models, which can be categorized into three distinct groups: i) GNN-based spatio-temporal models: STGCN [Yu *et al.*, 2017], GWNet [Wu *et al.*, 2019], AGCRN [Bai *et al.*, 2020]; ii) Temporal OOD prediction models: AdaRNN [Du *et al.*, 2021], Diversify [Lu *et al.*, 2022], RevIN [Kim *et al.*, 2021]; iii) Spatio-temporal OOD prediction models: CaST [Xia *et al.*, 2023], CauSTG [Zhou *et al.*, 2023], STONE [Wang *et al.*, 2024], STEVE [Ji *et al.*, 2025]. We evaluate the models using the official code released in the respective papers and the model parameters that achieve the best performance on the validation set.

Implementation Details

DynaSTar is implemented using PyTorch. We stack three spatio-temporal attention blocks to extract spatio-temporal representations. Within each block, the hidden dimensions for the temporal and spatial attention modules are 32, and the output FC layer has a dimension of 256. For memory-enhanced graph generation, we employ a 32×32 memory bank, with each projection having a dimension of 32. In the node-specific modulated predictor, an MLP with a hidden dimension of 8 generates a 64-dimensional dynamic condition-

Dataset	Baselines	Near-Term Deployment						Long-Term Deployment					
		MAE			MAPE			MAE			MAPE		
		15min	30min	60min	15min	30min	60min	15min	30min	60min	15min	30min	60min
SD	STGCN	50.80	50.51	51.52	54.37	54.12	54.53	52.56	52.54	53.56	55.57	55.46	55.89
	GWNet	48.34	48.87	49.67	53.13	52.86	60.29	49.41	50.76	51.85	54.22	55.95	59.90
	AGCRN	58.94	59.79	66.00	65.95	70.81	80.98	60.58	61.16	66.74	97.29	91.44	98.39
	AdaRNN	64.59	66.29	69.83	113.60	119.56	130.16	74.09	72.12	75.45	114.36	118.51	135.26
	Diversify	70.57	72.97	76.95	84.92	83.77	89.39	74.08	75.26	77.21	92.64	96.87	100.13
	RevIN	71.90	72.86	75.27	78.06	76.94	90.24	79.97	81.64	83.15	91.34	96.32	99.67
	CaST	39.67	39.40	41.29	38.27	40.92	41.35	39.81	40.37	41.06	40.99	40.24	41.71
	CauSTG	41.96	42.71	43.23	39.01	39.10	40.62	42.43	42.98	43.84	43.72	43.40	45.53
	STONE	29.24	31.51	37.20	26.12	28.90	38.22	29.15	31.00	36.06	23.87	26.26	34.34
	STEVE	37.16	37.05	37.63	36.50	36.97	37.54	36.80	36.92	38.24	32.79	32.97	33.61
	DynaSTar (Ours)	**23.62**	**24.78**	**27.08**	**25.58**	**26.32**	**27.94**	**23.79**	**24.86**	**26.87**	**22.95**	**23.58**	**24.78**
SGBA	STGCN	70.28	72.20	74.43	41.46	41.84	43.01	75.91	78.07	81.53	42.49	43.13	44.57
	GWNet	55.49	56.20	58.03	66.56	66.40	68.78	54.57	56.07	58.93	67.95	68.82	69.48
	AGCRN	65.19	68.24	75.32	80.87	82.17	90.67	66.92	67.77	74.35	91.09	97.63	103.17
	AdaRNN	71.93	73.36	76.27	125.69	133.01	142.70	83.18	87.51	90.22	129.80	136.69	145.31
	Diversify	78.02	78.26	78.82	94.99	97.19	100.37	79.50	80.26	84.64	96.43	96.97	99.20
	RevIN	73.76	74.49	75.73	92.37	96.32	98.05	82.80	82.72	85.34	93.18	99.50	104.22
	CaST	31.54	32.49	34.42	29.99	30.32	33.82	31.59	33.15	36.54	29.79	29.64	31.28
	CauSTG	39.57	40.51	42.06	39.98	40.59	40.82	37.36	38.76	42.63	36.36	37.62	39.81
	STONE	27.80	30.20	35.60	24.49	27.06	33.76	27.11	29.40	34.25	22.56	24.64	30.01
	STEVE	30.74	30.88	32.42	27.08	26.85	29.46	30.74	30.92	32.45	24.27	24.03	25.52
	DynaSTar (Ours)	**25.72**	**27.34**	**30.34**	**24.10**	**25.08**	**27.48**	**25.18**	**26.63**	**29.23**	**20.85**	**21.62**	**23.13**

Table 1: Performance of all methods on near- and long-term deployment OOT generalization, with the best results highlighted in bold and the suboptimal results underlined

ing vector, and its prediction output is projected by a 512-dimensional hidden layer. During training, the sparse graph generation samples 2 graphs with a temperature of 10, the temperature for the node-heterogeneity contrastive loss is 1, and the loss weights γ_1 and γ_2 are set to 0.01 and 1, respectively. We configure the model to predict the next 12 steps (i.e., 1 hour) based on the previous 12 steps of observations. Model performance is evaluated using the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) averaged over forecasting horizons of 3, 6, and 12 steps.

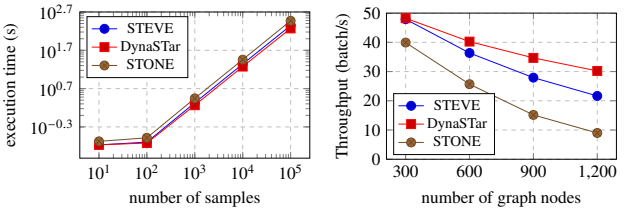
5.2 Evaluations

Multi-Phase Performance across Unseen Time Periods

We evaluate the model's performance across two critical scenarios: the near-term deployment phase and the long-term deployment phase, and report the average results of 5 independent runs of each method in Table 1. We make the following obeservations. (1) Most methods show higher prediction errors in long-term versus near-term scenarios. GNN-based and temporal OOD models degrade significantly over time, indicating OOT distribution shifts are more unpredictable and challenging. (2) GNN-based spatio-temporal models outperform specialized temporal OOD models, demonstrating that effective spatial aggregation is more critical than merely addressing temporal shifts. Although real-time neighbor features can partially mitigate temporal shifts, static or learnable static graphs that ignore potential spurious correlations still underperform in long-term deployment. Notably, AGCRN's high errors suggest severe overfitting of its node-adaptive parameters, hindering OOT generalization. (3) Spatio-temporal OOD prediction models consistently outperform others, highlighting the necessity of addressing distribution shifts. While methods like CauSTG and CaST learn effects of fixed confounders, they overlook the need for modeling of the constantly changed interference factors. STEVE and STONE partially consider confounders or spatial-temporal changes but neglect the continuous evolution of topology, which complicates learning invariant relationships for OOT prediction. (4) By modeling topological dynamics and continuously updating the graph structure, DynaSTar achieves state-of-the-art results in both near-term and long-term scenarios. This confirms our dynamic graph structure's effectiveness in perceiving topological evolution, providing consistent, superior adaptability across diverse scenarios without explicit interference modeling. Furthermore, our predictor learns heterogeneous yet invariant node-specific patterns, retaining node uniqueness while avoiding overfitting to spurious correlations, ensuring robust predictions against distribution shifts.

Impact of Components

We conduct an ablation study across two types of variants of DynaSTar: (1) Removing the memory-enhanced module (w/o ME), replacing the momentum-updated graph with an instantaneous graph (w/o MG), and disabling sparse graph generation (w/o SG); (2) Removing node-specific modulation (w/o NM) and removing neighborhood environment construction (w/o NE). As shown in Table 2, all ablated variants exhibit performance degradation. Among (1), w/o MG shows the most significant decline, underscoring the importance of modeling an evolving and stable graph for long-term deployment. Similarly, both (2) reduce accuracy, highlighting the necessity of modeling node-level heterogeneity and learning neighborhood-invariant representations to handle distribution shifts under topological evolution.



(a) Time vs. num. of samples (b) Throughput vs. graph size

Figure 3: Efficiency comparison at varying scales

Dataset	Scenario	w/o ME	w/o MG	w/o SG	w/o NM	w/o NE	**DynaSTar**
SD	NT	27.20	27.54	27.37	28.13	27.71	**27.08**
	LT	27.80	28.49	28.13	28.25	27.96	**26.87**
SGBA	NT	30.48	30.82	30.76	31.92	31.41	**30.34**
	LT	30.49	31.27	30.95	31.10	30.54	**29.23**

Table 2: Ablation results on MAE of 60-minute horizon

Efficiency Study

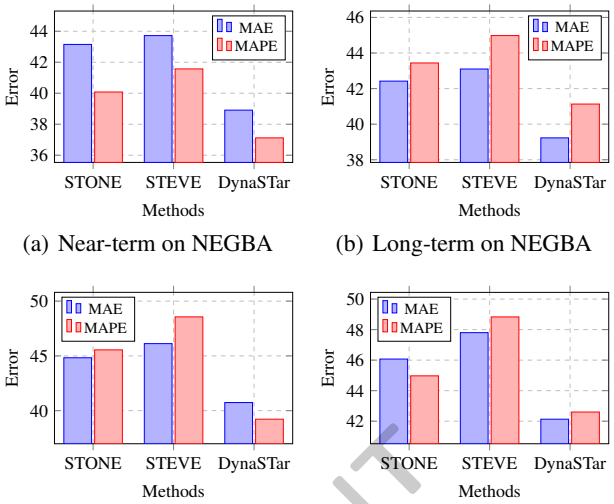
To evaluate computational efficiency, we compare the execution time and throughput of DynaSTar against the two leading baselines, STEVE and STONE, across varying problem scales. As shown in Fig. 3(a), execution time increases with sample size for all methods, but DynaSTar consistently achieves the lowest runtime. This advantage grows with graph size. Fig. 3(b) illustrates that DynaSTar maintains the highest throughput as the number of nodes increases. In contrast, STEVE and STONE exhibit steeper drops. This efficiency stems from DynaSTar's aggregation on sparse graphs, which avoids the overhead of dense-graph operations and ensures scalable performance under increased complexity.

Generalization Performance on Unseen Graphs

To evaluate generalization to unseen graph structures, we partition sensors in the GBA dataset (excluding SGBA) into two sub-graphs, forming the NWGBA and NEGBA test sets. Models trained exclusively on SGBA are tested on these datasets in a zero-shot setting. As shown in Fig. 4, DynaSTar achieves more accurate predictions than the competitive baselines STONE and STEVE, even on unseen topologies. This robustness stems from its node-heterogeneous invariant learning, which captures specific yet transferable patterns during training. Consequently, DynaSTar can adapt to new graphs with distinct topologies, demonstrating scalable generalization to previously unseen scenarios.

Analysis of Dynamic Graph Structure Learning

DynaSTar can generate meaningful dynamic edges by evaluating real-time dependency scores of nodes and updating them. To verify this, we visualize the 40-day traffic flow sequences of a node pair from the SD dataset alongside a heatmap showing the corresponding fluctuations of the edge probability scores output by our model in Fig. 5. Two key observations emerge. (1) The edge probability score exhibits two significant declines, corresponding to periods of approximately 4 and 9 days, respectively, during which the patterns of the two node's sequences differ significantly from each



(a) Near-term on NEGBA (b) Long-term on NEGBA

(c) Near-term on NWGBA (d) Long-term on NWGBA

Figure 4: Performance of different methods on unseen graphs

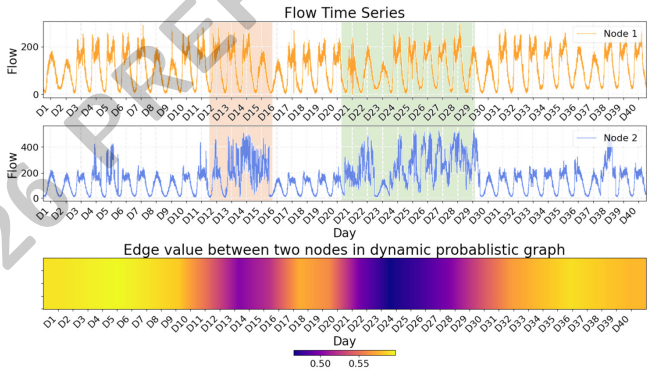


Figure 5: Visualization of the dynamic edge probability generated by our graph structure learning module for a node pair in SD

other. This indicates our model successfully captures changes in the topological relationship across different temporal segments. (2) The green-shaded period's high-frequency noise causes a more significant drop in correlation than the orange period's, highlighting our momentum-based update mechanism's sensitivity to varying topological change intensities.

6 Conclusion

This paper proposes a topological shifts-aware dynamic spatio-temporal graph invariant learning model (DynaSTar), a novel paradigm designed to enhance the robustness and generalization capability of OOT traffic prediction in long-term deployment scenarios. By explicitly capturing evolving graph structures and incorporating node-level invariant learning, DynaSTar achieves superior, stable performance in both near-term and long-term deployment. Future work could extend DynaSTar's principles to other spatio-temporal tasks (e.g., grid-based and irregular point process data modeling), laying a foundation for adaptive urban intelligent systems.

Ethical Statement

There are no ethical issues.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62272033), Frontier Technologies R&D Program of Jiangsu (Grant No. BF2024052), Nanjing Municipal Science and Technology Bureau (Grant No.202512136).

References

[Arjovsky *et al.*, 2019] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[Bai *et al.*, 2020] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[Cirstea *et al.*, 2022] Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2900–2913. IEEE, 2022.

[Du *et al.*, 2021] Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 402–411, 2021.

[Gong *et al.*, 2022] Yongshun Gong, Zhibin Li, Jian Zhang, Wei Liu, and Yu Zheng. Online spatio-temporal crowd flow distribution prediction for complex metro system. *IEEE Transactions on Knowledge and Data Engineering*, 34(2):865–880, 2022.

[Hao *et al.*, 2025] Xinyan Hao, Huaiyu Wan, Shengnan Guo, and Youfang Lin. Balancing imbalance: Data-scarce urban flow prediction via spatio-temporal balanced transfer learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 2865–2873, 2025.

[Huang *et al.*, 2025] Yiheng Huang, Xiaowei Mao, Shengnan Guo, Yubin Chen, Junfeng Shen, Tiankuo Li, Youfang Lin, and Huaiyu Wan. Std-plm: understanding both spatial and temporal properties of spatial-temporal data with plm. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025.

[Ioffe, 2015] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[Jang *et al.*, 2016] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[Ji *et al.*, 2025] Jiahao Ji, Wentao Zhang, Jingyuan Wang, and Chao Huang. Seeing the unseen: Learning basis confounder representations for robust traffic prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, page 577–588, New York, NY, USA, 2025. Association for Computing Machinery.

[Jin *et al.*, 2022] Yilun Jin, Kai Chen, and Qiang Yang. Selective cross-city transfer learning for traffic prediction via source city region re-weighting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 731–741, 2022.

[Ju *et al.*, 2024] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. A comprehensive survey on deep graph representation learning. *Neural Networks*, 173:106207, 2024.

[Kim *et al.*, 2021] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

[Li *et al.*, 2024] Zhonghang Li, Lianghao Xia, Yong Xu, and Chao Huang. FlashST: A simple and universal prompt-tuning framework for traffic prediction. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 28978–28988. PMLR, 21–27 Jul 2024.

[Li *et al.*, 2025] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[Liang *et al.*, 2021] Yuxuan Liang, Kun Ouyang, Junkai Sun, Yiwei Wang, Junbo Zhang, Yu Zheng, David Rosenblum, and Roger Zimmermann. Fine-Grained Urban Flow Prediction. In *Proceedings of the Web Conference 2021*, pages 1833–1845, Ljubljana Slovenia, April 2021. ACM.

[Liu *et al.*, 2022] Zechun Liu, Kwang-Ting Cheng, Dong Huang, Eric P Xing, and Zhiqiang Shen. Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4942–4952, 2022.

[Liu *et al.*, 2023] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in Neural Information Processing Systems*, 36:75354–75371, 2023.

[Lu *et al.*, 2022] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. Out-of-distribution representation learning for time series classification. *arXiv preprint arXiv:2209.07027*, 2022.

[Lv *et al.*, 2014] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2014.

[Lv *et al.*, 2026] Haochen Lv, Yan Lin, Shengnan Guo, Xiaowei Mao, Hong Nie, Letian Gong, Youfang Lin, and Huaiyu Wan. Ripcn: A road impedance principal component network for probabilistic traffic flow forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '26, page 1042–1053, New York, NY, USA, 2026. Association for Computing Machinery.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Perez *et al.*, 2018] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[Qu *et al.*, 2023] Hao Qu, Yongshun Gong, Meng Chen, Junbo Zhang, Yu Zheng, and Yilong Yin. Forecasting fine-grained urban flows via spatio-temporal contrastive self-supervision. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8008–8023, 2023.

[Shen *et al.*, 2025] Zekai Shen, Haitao Yuan, Xiaowei Mao, Congkang Lv, Shengnan Guo, Youfang Lin, and Huaiyu Wan. Towards an efficient and effective en route travel time estimation framework. In *International Conference on Database Systems for Advanced Applications*, pages 604–619. Springer, 2025.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wang *et al.*, 2023] Kun Wang, Hao Wu, Yifan Duan, Guibin Zhang, Kai Wang, Xiaojiang Peng, Yu Zheng, Yuxuan Liang, and Yang Wang. NuwaDynamics: Discovering and Updating in Causal Spatio-Temporal Modeling. In *The Twelfth International Conference on Learning Representations*, October 2023.

[Wang *et al.*, 2024] Binwu Wang, Jiaming Ma, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang. Stone: A spatio-temporal ood learning framework kills both spatial and temporal shifts. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 2948–2959, New York, NY, USA, 2024. Association for Computing Machinery.

[Weng *et al.*, 2023] Wenchao Weng, Jin Fan, Huifeng Wu, Yujie Hu, Hao Tian, Fu Zhu, and Jia Wu. A decomposition dynamic graph convolutional recurrent network for traffic forecasting. *Pattern Recognition*, 142:109670, 2023.

[Weng *et al.*, 2025a] Wenchao Weng, Hanyu Jiang, Mei Wu, Xiao Han, Haidong Gao, Guojiang Shen, and Xiangjie Kong. Let's group: a plug-and-play subgraph learning method for memory-efficient spatio-temporal graph modeling. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 3471–3479, 2025.

[Weng *et al.*, 2025b] Wenchao Weng, Mei Wu, Hanyu Jiang, Wanzeng Kong, Xiangjie Kong, and Feng Xia. Pattern-matching dynamic memory network for dual-mode traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 1907–1913, Macao, China, August 2019. International Joint Conferences on Artificial Intelligence Organization.

[Xia *et al.*, 2023] Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 37068–37088. Curran Associates, Inc., 2023.

[Yu *et al.*, 2017] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[Yuan *et al.*, 2024] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4095–4106, Barcelona Spain, August 2024. ACM.

[Zhou *et al.*, 2023] Zhengyang Zhou, Qihe Huang, Kuo Yang, Kun Wang, Xu Wang, Yudong Zhang, Yuxuan Liang, and Yang Wang. Maintaining the Status Quo: Capturing Invariant Relations for OOD Spatiotemporal Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pages 3603–3614, New York, NY, USA, August 2023. Association for Computing Machinery.