Memory-enhanced Invariant Prompt Learning for Urban Flow Prediction under Distribution Shifts

Haiyang Jiang[1], Tong Chen[1] (✉), Wentao Zhang[2], Quoc Viet Hung Nguyen[3], Yuan Yuan[4], and Yong Li[4] Hongzhi Yin[1]

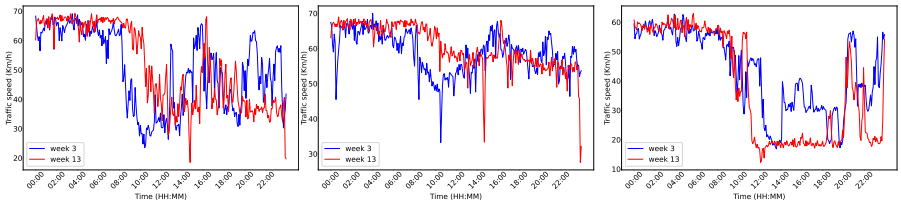
[1] The University of Queensland, Brisbane QLD, Australia
{haiyang.jiang,tong.chen}@uq.edu.au, db.hongzhi@gmail.com
[2] Peking University, Beijing, China wentao.zhang@pku.edu.cn
[3] Griffith University, Gold Coast, Australia quocviethung1@gmail.com
[4] Tsinghua University, Beijing, China y-yuan20@mails.tsinghua.edu.cn, liyong07@tsinghua.edu.cn

Abstract. While Spatial-Temporal Graph Neural Networks (STGNNs) excel at urban flow prediction, they struggle with distribution shifts caused by dynamic spatial-temporal environments. To improve generalizability to out-of-distribution (OOD) data, a typical solution is to disentangle invariant patterns that carry stable causal effects from variant ones that are environment-dependent. Existing OOD-robust methods attempt to model these environments but face challenges in quantifying dynamic changes and suffer from high computational costs. As a solution, we propose Memory-enhanced Invariant Prompt Learning (MIP), which enables environmental interventions directly within the latent space by learning a memory bank from the spatial-temporal urban flow graphs. Then, by performing spatial-temporal interventions on the variant prompts, diverse environments are constructed in the latent space to facilitate invariant learning. The invariant prompts, together with a memory-enhanced causal graph, are fed into an STGNN backbone to produce accurate predictions. Extensive experiments on two public urban flow datasets confirm MIP's effectiveness in improving robustness against OOD data.

Keywords: Spatial-temporal Graph Neural Networks · Out-of-distribution Generalization · Invariant Learning.

1 Introduction

Urban flow prediction, which forecasts traffic, pedestrian, and public transportation dynamics, is crucial for smart cities [18], public transit management [7, 33], and ride-sharing services [4, 38]. Typically modeled as a spatial-temporal graph, nodes (e.g., traffic sensors or geographical grids [29]) in urban flow are connected based on proximity, with the goal of predicting future traffic flow at each node. To effectively model these spatial-temporal dependencies, recent solutions are built upon deep learning-based approaches, particularly Spatial-Temporal Graph Neural Networks (STGNNs) [17, 33, 37]. These models leverage Graph Neural



(a) Traffic speed at node A.(b) Traffic speed at node B. (c) Traffic speed at node C.

Fig. 1: The sampled traffic speed recorded by three sensors in the METR-LA dataset, where nodes B and C are two closest sensors of Node A. The records correspond to two Wednesdays in the 3rd and 13th weeks of the dataset.

Networks (GNNs) [17, 33] to capture spatial correlations and sequential models like Recurrent Neural Networks (RNNs) [17, 20] and Temporal Convolution Networks (TCNs) [33, 37] for learning temporal dependencies. Some STGNNs further enhance predictions by incorporating dynamic graph structures based on temporal feature similarities [3, 26, 34] or modeling complex spatial-temporal interactions [6, 12].

Most STGNNs operate under the assumption that urban flow data adheres to the independent and identically distributed (I.I.D.) nature, which rarely holds true in real-world scenarios. In reality, once deployed, a trained model may need to perform inference on unseen data with patterns that are distinct significantly from the training data, a phenomenon referred to as distribution shift or out-of-distribution (OOD) during the test phase. Fig. 1 provides a real example from the METR-LA traffic dataset (see Section 5 for details), which demonstrates the traffic of three locations on two Wednesdays. Firstly, urban flow exhibits *continuous distribution shifts.* For each of the three geographic nodes, the two traffic records demonstrate entirely different patterns. Such continuous distribution shifts at each location disrupt long-term spatial-temporal patterns and hinder the generalizability of STGNNs. Secondly, there are *heterogeneous shifts across locations.* Although both nodes B and C are adjacent to node A, their patterns shift in distinct manners. This discrepancy complicates spatial correlations, as GNNs propagate noise from affected nodes. In urban flow prediction, spatial-temporal regularities can be easily disrupted by unexpected events such as traffic accidents or extreme weather. Moreover, during inference, it is generally impractical to assume prior knowledge about the occurrence of such perturbations that result in OOD data, thereby compromising prediction accuracy and diminishing the effectiveness of existing STGNNs. While frequent retraining can alleviate this issue, it is computationally prohibitive in such high-throughput applications. Thus, before entering the update cycle, an ideal STGNN should stay accurate for a reasonable period of time by generalizing to changed data distributions.

With the presence of distribution shifts in urban flow prediction, a key to enhancing the generalizability of STGNNs is to discover and leverage the invari-

ant (i.e., causal) patterns within spatial-temporal data. Many studies on OOD generalization [1, 25] point out that distribution shifts are driven by the dynamics of underlying environments, where invariant risk minimization [2, 22, 31] can be leveraged to optimize the model with augmented data drawn from diverse environments. As such, some methods [23, 24] decouple invariant patterns from variant ones learned from the data. For example, when handling graph-structured data, [5, 32, 39] learn two disentangled graph structures that contain either invariant or variant connections between nodes. Unfortunately, these models are misaligned with urban flow prediction tasks as they only focus on a static graph topology that does not assume the temporal evolution of node features.

To this end, we aim to build an OOD-robust STGNN that can distinguish invariant spatial-temporal patterns from urban flow data. Specifically, we propose Memory-enhanced Invariant Prompt learning (MIP), a novel solution to urban flow prediction. In MIP, we attach a memory bank to the STGNN architecture, which learns and memorizes the causal patterns from the dynamic node features. Based on the information stored in the memory bank, a new graph structure reflecting the semantic causality between different locations is built, providing a complementary graph view to the default, distance-based graph structure for node representation learning. Then, the prompt vectors carrying invariant or variant patterns are extracted respectively by attentively querying the memory bank with each node's features. Furthermore, to facilitate end-to-end optimization via invariant risk minimization and ensure disentanglement between invariant and variant patterns, we put forward an innovative intervention pipeline that directly operates on the extracted variant prompts. Different from existing invariant learning methods [35, 41], MIP bypasses the need for learning additional representations of different environments, and the designed intervention is a simple-yet-effective approach for implicitly mimicking the effect from data distribution shifts to node representations. The disentangled invariant patterns, along with both the geographical and causal graphs, are eventually fed into a spatial-temporal backbone model to make accurate urban flow predictions. To be concise, our contributions are summarized below:

- **New Challenge.** We highlight a largely overlooked challenge in urban flow prediction: the pervasive presence of OOD data that hinders model generalizability. To address this, we propose a new framework, namely MIP to mitigate distribution shifts in urban flow prediction.
- **New Method.** We extract invariant and variant features from a trainable memory bank and generate a supplementary graph structure. By implementing interventions on variant patterns and leveraging an invariant learning scheme, the invariant patterns are disentangled from the noisy data to facilitate accurate predictions.
- **State-of-the-art Performance.** Extensive experiments on two real-world benchmark datasets have demonstrated the superiority of our method over state-of-the-art baselines when faced with OOD urban flow data.

2 Related Work

2.1 Deep Learning for Urban Flow Prediction

Recently, spatial-temporal neural networks (STGNNs) have established themselves as state-of-the-art choices for urban flow prediction. STGNNs consist of GNN-based modules and sequential models that are alternately stacked, where typical variants include DCRNN [17], GWNet [33], STGCN [37] and ST-MGCN [9]. Furthermore, attention mechanisms, including multi-head attention, are additionally used in fusing spatial and temporal information, such as GMAN [40], ASTGCN [10], and PDFormer [13]. Moreover, introducing some trainable features can also improve the performance of STGNNs, even with naive backbone models, such as STID [27], STAEformer [21], and MegaCRN [14]. Besides, some physical theories can also guide spatial-temporal prediction, such as PGML [15] and STDEN [11]. However, these methods are designed based on the I.I.D assumption, making the extracted patterns solely dependent on the observed samples. Thus, these methods are prone to incorrect predictions when facing unobserved data with distribution shifts.

2.2 Handling Out-of-Distribution (OOD) Data in Prediction

There are some models [23, 35, 41] dedicated to overcoming distribution shifts in spatial-temporal data. For example, CaST [35] disentangles the environmental feature and the entity feature based on causal treatments [23], and it replaces the environment feature with the vector closest to it in the environment codebook, which contains vectors representing environments. CauSTG [41] designs a hierarchical invariance explorer, which merges the models trained across various environments. STONE [28] learns both spatial and temporal similarity matrices as adjacency matrices for STGNN to make predictions and implements intervention by masking these two adjacency matrices. By differentiating raw data from distinct environments, the aforementioned methods can capture data exhibiting distribution shifts, enabling STGNNs to learn features across diverse distributions. Consequently, STGNNs can achieve accurate predictions on OOD data. However, these methods are heavily dependent on specifically designed model mechanisms and exhibit high sensitivity to the number of virtual environments.

3 Preliminaries

3.1 Problem Formulation

In urban flow data, a geolocation graph can be defined as: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of N nodes and $\mathcal{E}$ is the set of edges. Correspondingly, $\mathbf{A} \in \{0, 1\}^{N \times N}$ is the derived adjacency matrix. In $\mathcal{G}$, a node is a spatial object like a traffic sensor, where the edges between nodes are commonly established by thresholding their physical distances [33, 37], thus $\mathbf{A}$ is constant in this task. At each time step $1 \leq t \leq T$, all nodes' dynamic features are represented via a matrix $\mathbf{X}^t \in \mathbb{R}^{N \times k}$,

with k representing the dimensionality of time-varying features. Following the commonly adopted setting [33], given the observed T historical observations $\{\mathbf{X}^t\}_{t=1}^{T}$ and the geolocation graph $\mathcal{G}$, the task objective is to train a model that predicts the next T urban flow signals $\{\mathbf{X}^{t'}\}_{t'=T+1}^{2T}$:

$$\mathbf{Y} \simeq f_\theta(\mathbf{X}, \mathcal{G}), \tag{1}$$

where $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{T\times N\times k}$ are respectively the tensorized versions of input $\{\mathbf{X}^t\}_{t=1}^{T}$ and output $\{\mathbf{X}^{t'}\}_{t'=T+1}^{2T}$, and $f_\theta(\cdot)$ is the prediction model parameterized by θ.

Usually, the optimization of θ is based on the I.I.D assumption, which means the training and test samples are drawn from the same distribution. In urban flow prediction, this assumption can hardly be guaranteed as the training and test data points are drawn from different environments, respectively denoted by E_{train} and E_{test}. Thus, the optimal model parameter θ^* should achieve minimal generalization error on an OOD test set, described as follows:

$$\begin{aligned} &\min \mathbb{E}_{(\mathbf{X}',\mathbf{Y}')\sim p(\mathbf{X}',\mathbf{Y}'|E_{test})}\mathcal{L}(f_\theta(\mathbf{X}',\mathcal{G}),\mathbf{Y}'),\\ s.t.\ \theta^* =&\arg\min_{\theta} \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim p(\mathbf{X},\mathbf{Y}|E_{train})}\mathcal{L}(f_\theta(\mathbf{X},\mathcal{G}),\mathbf{Y}),\ \ E_{train}\neq E_{test}, \end{aligned} \tag{2}$$

where $\mathcal{L}(\cdot)$ quantifies the prediction error.

3.2 Invariant Learning under Distribution Shifts

Let $\mathbf{H}_I, \mathbf{H}_V \in \mathbb{R}^{T\times N\times d}$ respectively denote all $T \times N$ variant and invariant patterns with dimensionality d extracted from input $\mathbf{X}$. Drawing on causality theory [23,24], there exists a prediction function $\text{pred}(\cdot)$, for which the invariant feature $\mathbf{H}_I$ is sufficiently predictive for $\mathbf{Y}$ and the variant feature $\mathbf{H}_V$ does not hold causation to $\mathbf{Y}$ [32, 39]. Given that, we can rewrite our objective below:

$$\begin{aligned} &\arg\min_{\theta,\psi} \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim p(\mathbf{X},\mathbf{Y}|E_{train})}\mathcal{L}(f_\theta(\mathbf{H}_I,\mathcal{G}),\mathbf{Y}),\\ s.t.\ &\mathbf{H}_I,\mathbf{H}_V = f_\psi(\mathbf{X}),\ \ \mathbf{Y}\perp\mathbf{H}_V|\mathbf{H}_I, \end{aligned} \tag{3}$$

where we use $\mathbf{H}_I, \mathbf{H}_V \in \mathbb{R}^{T\times N\times d}$ to respectively denote all $T \times N$ variant and invariant patterns with dimensionality d extracted from input $\mathbf{X}$. Here, $f_\psi(\cdot)$ is the invariant learning backbone model parameterized by ψ that disentangles invariant patterns with the variant ones from the dynamic node features. In this setup, the prediction model f_θ is only fed with the invariant patterns to derive final predictions. Based on the formulation, a key step is to train $f_\psi(\cdot)$ towards distinguishing invariant and variant patterns. To achieve this, based on the interventional distribution in causality theory [32], a common objective can be described following invariant learning loss:

$$\begin{aligned} \min_{\psi}\ &\mathbb{E}_{(\hat{\mathbf{X}},\mathbf{Y})\sim p(\hat{\mathbf{X}},\mathbf{Y}|\hat{E})}\mathcal{L}(\text{pred}(f_\psi(\hat{\mathbf{X}}),\mathcal{G}),\mathbf{Y})\\ &+\lambda Var_{(\hat{\mathbf{X}},\mathbf{Y})\sim p(\hat{\mathbf{X}},\mathbf{Y}|\hat{E})}\mathcal{L}(\text{pred}(f_\psi(\hat{\mathbf{X}}),\mathcal{G}),\mathbf{Y}), \end{aligned} \tag{4}$$

where $\hat{E} \neq E_{train}$ denotes an intervention sampled from an intervention set, which imposes changes on the original environment and leads to features $\hat{\mathbf{X}} \neq \mathbf{X}$ with shifted distributions. Ideally, $\hat{E}$ only performs intervention on variant patterns, leaving the invariant features unaffected. pred$(\cdot)$ denotes a predictor that uses both invariant and variant patterns emitted by $f_\psi(\hat{\mathbf{X}})$ to predict the ground truth label. Note that as pred$(\cdot)$ is not responsible for generating the final predictions, it does not necessarily share the same structure or parameterization with $f_\theta(\cdot)$. The first term minimizes the prediction loss, whereas in the second term, Var denotes the variance and λ is a balancing hyperparameter. As the variant patterns are unrelated to the label $\mathbf{Y}$, the prediction should remain stable regardless of the variant patterns introduced by $\hat{E}$, translating into a lower variance. As such, $f_\psi(\cdot)$ is trained to differentiate invariant patterns $\mathbf{H}_I$ and variant patterns $\mathbf{H}_V$ amid distribution shifts in urban flow.

4 MIP: The Proposed Method

In this section, we introduce a universal framework named Memory-enhanced Invariant Prompt learning (MIP) for urban flow prediction under OOD scenarios, whose main components are depicted in Figure 1. In what follows, we unfold the design of MIP by introducing the design of the memory bank, as well as the backbones for invariant learning (i.e., $f_\psi(\cdot)$) and spatial-temporal prediction (i.e., $f_\theta(\cdot)$) backbone model.

4.1 Memory-enhanced Invariant Prompt Learning

A key advantage of MIP is that rather than generating intervened environments $\hat{E}$ and simulating changes in latent patterns, it directly intervenes in the latent space to mimic representation changes after intervention. To do this, we first mine latent patterns correlated with predicted labels from time-varying node features. Therefore, we extract and store these representative causal features with a memory bank [14]. The memory bank can be represented as $\mathbf{\Phi} \in \mathbb{R}^{M \times d}$, where M and d represent the number of virtual nodes and their dimensions, respectively. Essentially, each of the M virtual nodes in the memory bank is assigned a d-dimensional prototype vector $\mathbf{\Phi}[m] \in \mathbb{R}^d$ $(m \leq M)$ that summarizes a part of the latent, invariant features within the spatial-temporal node features $\mathbf{X}$. The memory bank $\mathbf{\Phi}$ supports two subsequent computations: generating variant and invariant prompts through a querying process as described below, and providing a causal graph to supplement the geographical graph for node representation learning as described in Section 4.3.

Learning Invariant and Variant Prompts. As a core part of OOD generalization, given all nodes' temporal features $\mathbf{X}^t \in \mathbb{R}^{N \times k}$ at time t, MIP extracts both causal and spurious patterns from them – which we term invariant and variant prompts in this work. To do this, we firstly project $\mathbf{X}^t$ into a query matrix $\mathbf{Q}_t \in \mathbb{R}^{N \times d}$:

$$\mathbf{Q}_t = \mathbf{X}_t \mathbf{W}_Q + \mathbf{b}_Q, \tag{5}$$

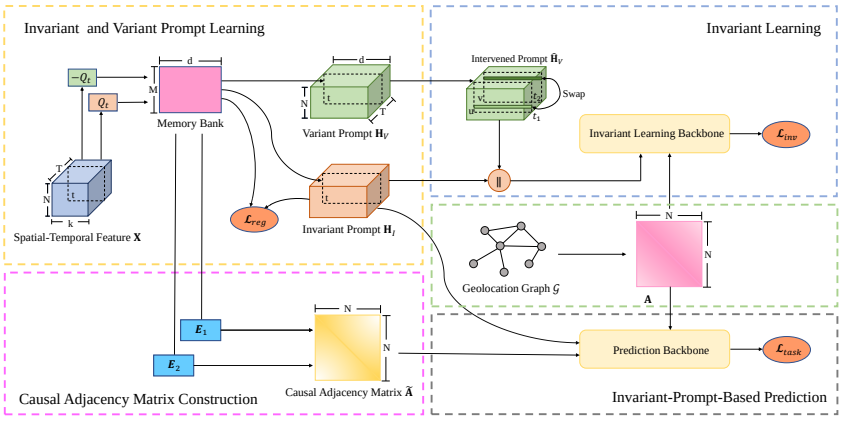


Fig. 2: The main components of MIP.

where $\mathbf{W}_Q \in \mathbb{R}^{k\times d}$ and $\mathbf{b}_Q \in \mathbb{R}^d$ are trainable parameters of the linear layer. As $\mathbf{Q}_t$ contains both invariant and variant patterns, we further disentangle them by querying the invariant memory bank. To obtain invariant prompts, we multiply the query matrix $\mathbf{Q}_t$ with the memory bank $\mathbf{\Phi}$ to obtain an affinity matrix $\mathbf{S}_I^t$, based on which the invariant prompt can be drawn from the memory bank in a self-attentive fashion. This process can be formulated as:

$$\mathbf{H}_I^t = \mathbf{S}_I^t \mathbf{\Phi}, \;\; \mathbf{S}_I^t = \text{softmax}\left(\mathbf{Q}_t \mathbf{\Phi}^\top\right), \tag{6}$$

where $\mathbf{H}_I^t \in \mathbb{R}^{N\times d}$ is the computed invariant prompt. Similarly, the variant prompt can be extracted in an analogous process, with a minor modification:

$$\mathbf{H}_V^t = \mathbf{S}_V^t \mathbf{\Phi}, \;\; \mathbf{S}_V^t = \text{softmax}\left(-1 \cdot \mathbf{Q}_t \mathbf{\Phi}^\top\right), \tag{7}$$

where negation is applied before the softmax function, so as to flip the score distribution and assign higher weights to invariant patterns that are less relevant to the memory $\mathbf{\Phi}$. As this is executed for all time steps, we can obtain a sequence of T prompts $\{\mathbf{H}_I^1, \mathbf{H}_I^2, ..., \mathbf{H}_I^T\}$ and $\{\mathbf{H}_V^1, \mathbf{H}_V^2, ..., \mathbf{H}_V^T\}$. By respectively concatenating invariant and variant prompts across time, we can obtain two prompt tensors $\mathbf{H}_I, \mathbf{H}_V \in \mathbf{R}^{T\times N\times d}$ for subsequent computations.

4.2 Invariant Learning with Latent Intervention

As per our discussions earlier, the variant prompt $\mathbf{H}_V$ is environment-dependent but unrelated to the label $\mathbf{Y}$, and the invariant prompt $\mathbf{H}_I$ is causally linked to $\mathbf{Y}$. To distill $\mathbf{H}_I$ from $\mathbf{X}$, a common approach as described in Section 3.2 is to perform invariant learning with intervened raw data $\hat{\mathbf{X}}$. However, directly intervening the raw data is a less favorable option for urban flow prediction tasks due to the risk of introducing additional noises, while some workarounds [30, 32, 35] need to additionally parameterize and learn the underlying environments

Algorithm 1 INTERVENE($\mathbf{H}_V, r$)

Input: Variant prompt tensor $\mathbf{H}_V \in \mathbb{R}^{T\times N\times d}$, intervention rate r
Output: Intervened variant prompt $\hat{\mathbf{H}}_V \in \mathbb{R}^{T\times N\times d}$
$\hat{\mathbf{H}}_V \leftarrow \mathbf{H}_V$
for s=1,2,...,$\lfloor \frac{rN}{2} \rfloor$ **do**
    Randomly sample a node pair (w, v) s.t. $w, v \in [1, N]$;
    Randomly select a time step pair (i, j) s.t. $i, j \in [1, T]$;
    $\hat{\mathbf{H}}_V[i, w] \leftarrow \mathbf{H}_V[j, v]$, $\hat{\mathbf{H}}_V[j, v] \leftarrow \mathbf{H}_V[i, w]$;
end for
return $\hat{\mathbf{H}}_V$

in order to alter the raw data distribution. Also, when STGNNs are used as the predictor pred$(\cdot)$, multiple complex forward passes are required to optimize Eq.(4), which is computationally impractical considering the large spatial and temporal spans N and T in urban flow graphs.

Latent Intervention Mechanism. In this paper, we innovatively propose to generate spatial-temporal interventions in the latent space, which more effectively mimics the changes in the learnable patterns after possible distribution shifts within the input data. Specifically, given the extracted variant prompts $\mathbf{H}_V$, we exchange features in $\mathbf{H}_V$ between different nodes and time points with a predefined rate r. The details of the latent intervention mechanism are described in Algorithm 1. To be succinct, we simplify this process into the following:

$$\hat{\mathbf{H}}_V = \text{INTERVENE}(\mathbf{H}_V, r), \tag{8}$$

where $\hat{\mathbf{H}}_V$ denotes the intervened variant prompts after rN feature exchanges between nodes have taken place. Note that, the swap is not constrained to node features at the same time step, so as to account for the spatial-temporal fluctuations within the variant patterns. Also, by producing intervened variant prompts with representations learned from the original input data, the generated $\hat{\mathbf{H}}_V$ remains plausible and challenging for refining the invariant prompts $\hat{\mathbf{H}}_I$.

Invariant Learning. After obtaining the intervened variant prompts $\hat{\mathbf{H}}_V$, we are able to train the prompt extractor described in Section 4.1 via invariant learning, so as to distinguish the invariant and variant prompts. To achieve this, the invariant and the intervened variant prompts are concatenated and then input into a supplementary predictor pred$(\cdot)$ to generate predictions:

$$\tilde{\mathbf{Y}} = \text{pred}(\mathbf{H}_I || \hat{\mathbf{H}}_V, \mathbf{A}), \tag{9}$$

where $||$ denotes tensor concatenation along the last dimension, $\tilde{\mathbf{Y}} \in \mathbb{R}^{T\times N\times k}$ is the predicted urban flow at all locations and time steps. The choice of pred$(\cdot)$ is flexible with most STGNNs. Since pred$(\cdot)$ is only responsible for differentiating invariant and variant prompts and will not be used for computing the final predictions, we adopt GWNet [33], a simple yet effective STGNN as pred$(\cdot)$. For

training, we first define the loss for a single node n at one time step t:

$$l(t,n) = \frac{1}{k}\sum_{k'=1}^{k} |\tilde{\mathbf{Y}}[t,n,k'] - \mathbf{Y}[t,n,k']|, \tag{10}$$

based on which the invariant learning loss is defined:

$$\mathcal{L}_{inv} = \mathbb{E}_{(t,n)} l(t,n) + \lambda_1 Var_{(t,n)} l(t,n), \;\; t \in [1,T], \;\; n \in [1,N], \tag{11}$$

where the first and second terms respectively reduce the mean and variance of the prediction error across locations and time steps. More specifically, the first term ensures that pred$(\cdot)$ is optimized towards correctly predicting the urban flow in different environments with $\mathbf{H}_I$, while the second term enforces that when the predictions are conditioned on $\mathbf{H}_I$, there are minimal performance fluctuations despite the presence of noisy signals from intervened variant prompts $\mathbf{H}_V$.

4.3 Urban Flow Prediction with Causal Graph Generation

Once the invariant features $\mathbf{H}_I$ are extracted, a spatial-temporal backbone model is in place for producing the final predictions. In MIP, our backbone model consists of alternately stacked GNN layers and temporal Transformer layers, which take invariant prompts $\mathbf{H}_I$ and adjacency matrix $\mathbf{A}$ as its input. However, $\mathbf{A}$ is normally constructed solely based on the physical distances between nodes. As a result, this can introduce biases during GNN's information propagation, because geographic proximity does not necessarily imply similar temporal patterns, especially in OOD scenarios. In this section, we introduce a memory-based approach for generating a causal graph as a supplement to $\mathbf{A}$, followed by details of the backbone STGNN. **Causal Graph Generation.** To address the limitation of the geolocation graph that is purely distance-based, we introduce an auxiliary graph based on the semantic distance between causal node representations, which are constructed from highly invariant features from the memory bank $\mathbf{\Phi}$. This causal graph complements the geolocation-based graph, thus providing additional predictive signals. This memory bank-based causal graph is constructed as the following:

$$\tilde{\mathbf{A}} = \text{softmax}\left(\mathbf{E}_1 \mathbf{E}_2^\top\right), \;\; \mathbf{E}_1 = \mathbf{W}_A \mathbf{\Phi}, \;\; \mathbf{E}_2 = \mathbf{W}_B \mathbf{\Phi}, \tag{12}$$

where $\mathbf{W}_A, \mathbf{W}_B \in \mathbb{R}^{N \times M}$ are trainable projection matrices that map the M prototype vectors in the memory bank into N node representations $\mathbf{E}_1$ and $\mathbf{E}_2$. As the memory bank already encapsulates critical information about the urban flow, the newly developed causal adjacency matrix provides additional information propagation channels between nodes.

GNN Layer. The GNN layer is fed with both the geographical and causal adjacency matrices $\mathbf{A}$, $\tilde{\mathbf{A}}$ and the invariant prompts $\mathbf{H}_I$ to learn node representations with information propagation. Since the geographical adjacency matrix $\mathbf{A}$ is symmetric and hardly captures the directed nature of interactions in urban

flow data, we derive forward and backward transition matrices from $\mathbf{A}$ through a bidirectional, degree-weighted random walk [17]:

$$\mathbf{P}_f = \mathbf{D}^{-1}\mathbf{A},\ \mathbf{P}_b = (\mathbf{D}^\top)^{-1}\mathbf{A}^\top, \tag{13}$$

where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$. It is worth mentioning that, the GNN layer processes each time step t separately. By incorporating the causal adjacency matrix, for each time step t, the propagation process in the GNN from layer l to $l+1$ is summarized as follows:

$$\mathbf{G}_{l+1}^t = \sum_{z=0}^{Z}(\mathbf{P}_f^z\mathbf{G}_l^t\mathbf{W}_1^z + \mathbf{P}_b^z\mathbf{G}_l^t\mathbf{W}_2^z + \tilde{\mathbf{A}}^z\mathbf{G}_l^t\mathbf{W}_3^z), \tag{14}$$

where $z \leq Z$ controls the order of the information propagation, and $\mathbf{W}^{(k)} \in \mathbb{R}^{d\times d}$ denotes the learnable weights. Note that the initial node embeddings are set to $\mathbf{G}_0^t = \mathbf{H}_I^t$ when $l = 0$.

Temporal Transformer Layer. Once the GNN layer processes all graphs at all time steps, we can collect T feature matrices produced by the final graph propagation layer, denoted by $\mathbf{G}^1, \mathbf{G}^2, ..., \mathbf{G}^T \in \mathbb{R}^{N\times d}$. For a certain node n, we can stack all its d-dimensional, time-sensitive features across T steps into a matrix, denoted by $\mathbf{G}_n \in \mathbb{R}^{T\times d}$. With that, we learn the dependencies across all temporal features of a node through a transformer layer as:

$$\begin{aligned}\mathbf{G}_n' &= \text{softmax}\left(\frac{\mathbf{G}_n\mathbf{W}_Q\left(\mathbf{G}_n\mathbf{W}_K\right)^\top}{\sqrt{d}}\right)\left(\mathbf{G}_n\mathbf{W}_V\right),\\ \mathbf{Z}_n &= \text{MLP}(\mathbf{G}_n'),\end{aligned} \tag{15}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d}$ are trainable query, key and value projection weights, and $\text{MLP}(\cdot)$ denotes a feedforward multilayer perceptron.

Prediction Layer. After obtaining N outputs for all nodes $\mathbf{H}_1', \mathbf{H}_2', ..., \mathbf{H}_N' \in \mathbb{R}^{T\times d}$, we can stack all feature matrices into $\mathbf{Z} \in \mathbb{R}^{T\times N\times d}$. Then, we generate the final predictions with an MLP:

$$\hat{\mathbf{Y}} = \text{MLP}\left(\mathbf{Z}\right), \tag{16}$$

where the MLP projects $\mathbf{Z}$ into $\hat{\mathbf{Y}} \in \mathbb{R}^{T\times N\times k}$ that carries the predicted urban flow per time step per location.

4.4 Model Optimization

Now, we detail the optimization strategy for MIP. Firstly, based on the prediction $\hat{\mathbf{Y}} \in \mathbb{R}^{L\times N\times k}$ generated by the backbone model, the prediction error is as follows:

$$\mathcal{L}_{task} = \frac{1}{TN}\sum_{t,n,k'=1}^{T,N,k}|\hat{\mathbf{Y}}[t,n,k'] - \mathbf{Y}[t,n,k']|. \tag{17}$$

In addition, as we extract invariant prompts from a memory bank, we use an auxiliary regularization loss to enhance the quality of features stored within the memory bank:

$$\mathcal{L}_{reg} = \sum_{t,n}^{T,N} \max\left\{\|\mathbf{H}_I^t[n] - \mathbf{\Phi}[a]\|^2 - \|\mathbf{H}_I^t[n] - \mathbf{\Phi}[b]\|^2 + \kappa, 0\right\} + \sum_{t,n}^{T,N} \|\mathbf{H}_I^t[n] - \mathbf{\Phi}[a]\|^2, \tag{18}$$

where κ is a distance margin, a, b are the indices of the most and second similar virtual nodes w.r.t. node n based on the affinity score $\mathbf{S}_I^t$ computed in Eq.(6). As such, $\mathcal{L}_{reg}$ encourages diversity within the information encoded by different virtual node prototype vectors in the memory bank. Finally, the optimization objective aims to minimize the following overall loss:

$$\mathcal{L} = \mathcal{L}_{task} + \mathcal{L}_{inv} + \lambda_2 \mathcal{L}_{reg}, \tag{19}$$

with a balancing hyperparameter λ_2. As MIP is being trained towards convergence, the intervention on variant patterns, i.e., $\hat{\mathbf{H}}_V = \text{Intervene}(\mathbf{H}_V, r)$ is re-executed in every training epoch, so as to inject more variations in the supervision signals. It is worth noting that, once MIP is trained, only the spatial-temporal backbone model described in Section 4.3 is activated for making predictions in the inference stage.

5 Experiments

5.1 Experimental Settings

We evaluate our model on two well-established benchmarks, namely METR-LA [17] and NYCBike [38]. METR-LA [17] is a traffic speed prediction dataset collected with 207 sensors across Los Angeles, from 1st March 2012 to 30th June 2012, the data points are sampled with 5 5-minute time interval. NYCBike1 [38] is a dataset of bike rental records from 1st April 2014 to 30th September in New York City, where the city is divided into 8×16 equally-sized grids, and the data points are sampled with a 1-hour time interval. As NYCBike1 records both in and out flows of bikes, we treat them as two prediction tasks and respectively denote them as NYCBike1 (In) and NYCBike1 (Out).

We split both datasets chronologically: the first 60% is for training, the following 10% for validation, and three test sets are constructed by evenly slicing the remaining data (10% for each). This is to fully mimic real-world application scenarios where a trained model is expected to provide predictions for multiple consecutive time periods with varying distributions. For convenience, we number the tree test sets with 0, 1, and 2. Generally, as test sets 0-2 become farther apart from the training set in time, their distribution shifts tend to become stronger. Based on the number of time steps available, we predict the next 12 time steps based on the past 12 on METR-LA and predict the next 6 time steps based on the past 6 on NYCBike. We compare MIP with

Table 1: Performance comparison results. The best results are marked in bold and the second-best results are underlined.

METR-LA	test set 0			test set 1			test set 2			overall results		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	3.33	7.15	10.14%	3.63	7.47	10.90%	3.63	7.58	10.21%	3.53	7.40	10.42%
DCRNN	3.33	7.28	10.01%	3.58	7.49	10.80%	3.69	7.87	10.41%	3.53	7.55	10.41%
STNorm	3.33	7.17	10.09%	3.65	7.57	11.16%	3.63	7.61	10.23%	3.53	7.45	10.49%
GMSDR	3.27	6.99	9.75%	**3.49**	7.36	10.83%	**3.50**	7.47	10.01%	**3.42**	7.27	10.20%
MegaCRN	**3.22**	7.05	9.69%	3.64	7.65	11.04%	3.79	8.00	10.75%	3.55	7.57	10.49%
CauSTG	3.33	7.08	9.86%	3.64	7.44	10.81%	3.66	7.55	10.10%	3.55	7.36	10.26%
TESTAM	3.36	7.33	9.56%	3.62	7.58	**10.26%**	3.69	7.89	9.98%	3.56	7.60	9.93%
MIP	3.28	**6.87**	**9.52%**	3.55	**7.19**	10.40%	3.57	**7.28**	**9.73%**	3.46	**7.11**	**9.88%**
NYCBike1 (In)	test set 0			test set 1			test set 2			overall results		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	4.90	8.41	50.38%	4.69	7.55	51.11%	5.33	9.42	63.05%	4.97	8.46	54.84%
DCRNN	6.24	10.04	80.58%	5.90	9.18	77.03%	6.47	10.74	92.59%	6.20	9.99	83.40%
STNorm	4.83	8.52	46.49%	**4.53**	8.29	50.51%	5.28	9.37	56.21%	4.88	8.38	49.35%
GMSDR	5.10	8.96	48.70%	4.86	8.14	49.14%	5.41	9.68	61.02%	5.12	8.92	52.95%
MegaCRN	**4.62**	**7.96**	46.65%	5.15	8.87	55.35%	5.62	9.51	65.81%	5.13	8.78	55.93%
CauSTG	4.95	8.51	49.21%	4.83	7.91	49.47%	5.37	9.45	59.73%	5.05	8.63	52.80%
TESTAM	5.06	8.51	47.69%	5.18	8.38	49.23%	5.83	9.93	59.42%	5.04	8.66	51.13%
MIP	4.74	8.13	**45.10%**	4.56	**7.27**	**43.32%**	**5.26**	**9.18**	**55.27%**	**4.87**	**8.16**	**47.56%**
NYCBike1 (Out)	test set 0			test set 1			test set 2			overall results		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	5.04	8.84	46.64%	4.78	7.83	47.81%	5.51	9.56	62.02%	5.11	8.74	52.16%
DCRNN	5.38	9.04	54.46%	4.97	7.84	52.51%	5.78	10.15	65.90%	5.38	9.01	57.62%
STNorm	5.19	9.23	44.80%	4.83	7.90	43.63%	5.57	9.70	58.47%	5.20	8.94	48.97%
GMSDR	4.98	8.75	45.79%	4.73	7.74	44.74%	5.55	9.56	60.67%	5.09	8.68	50.40%
MegaCRN	5.53	9.58	49.97%	5.13	8.49	48.76%	5.82	10.48	64.23%	5.49	9.51	54.32%
CauSTG	5.04	8.73	46.13%	4.95	8.04	46.68%	5.60	9.71	**56.65%**	5.20	8.83	49.82%
TESTAM	**4.88**	**8.45**	**43.63%**	5.08	8.59	46.03%	5.54	9.59	56.81%	5.16	8.88	48.82%
MIP	4.94	8.68	44.72%	**4.66**	**7.56**	**43.22%**	**5.51**	**9.45**	57.60%	**5.03**	**8.56**	**48.51%**

the following state-of-the-art baselines: STGCN [37], DCRNN [17],STNorm [8], GMSDR [19],MegaCRN [14], CauSTG [41], TESTAM [16]. Similar to previous studies [19, 28, 33], we evaluate all methods in terms of Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). Hyperparameters and implementation notes are available in our released code: https://github.com/Ocean-Jiang0729/MIP.

5.2 Performance Comparison with Baselines

We compare MIP with SOTA baselines, recording the final horizon matrices in Table 1. MIP consistently outperforms all baselines across the three test sets, demonstrating strong generalization, versatility, and adaptability in urban flow prediction. On the METR-LA dataset, as test set 0 is the closest to the training set, its distribution changes less than the other two test sets. Thus, the models achieve similar performance

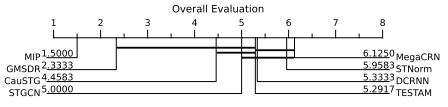
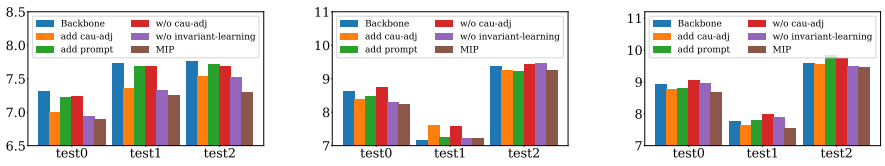


Fig. 3: Critical difference w.r.t. performance on all 9 test sets. Smaller scores indicate better performance.



(a) METR-LA dataset. (b) NYCBike1 dataset(In). (c) NYCBike1 dataset(Out).

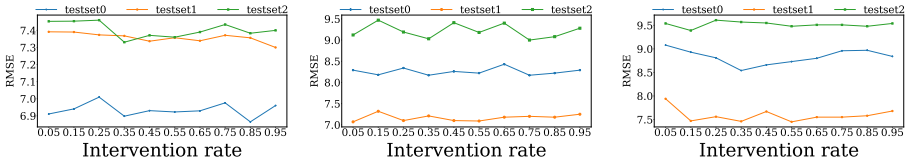
Fig. 4: Ablation study: RMSE of MIP and its variants.

on test 0, and some baselines get the best result, such as the MAPE of MegaCRN on test set 0. However, the distribution shift happens more on test 1 and test 2, and the performance of all the models becomes worse. Some baseline models, such as MegaCRN, TESTAM, and GMSDR, get similar RMSE on test set 0, while their RMSE scores on test set 1 and test set 2 increase largely. On the NYCBike1 dataset, all the models perform well on test set 1 on both bikes' in and out tasks. A reasonable explanation is that the distribution of this test set is more similar to that of the training set. Although TESTAM achieved the best performance on all three evaluation matrices on test set 0, it performed worse on test sets 1 and 2, even with the biggest RMSE on test set 1. Notably, while some models outperform MIP on test set 0, they struggle with distribution shifts and consequently exhibit performance degradation on test set 2. Moreover, when evaluating these models across all test sets (referred to as overall results), our model consistently delivers superior performance across all evaluation metrics, with the sole exception of achieving second place in terms of MAE on the METR-LA dataset. In Fig 3, we also calculate the critical difference diagram of all the models on all the datasets and evaluation matrices. By obtaining the highest rank among all 9 test sets, MIP demonstrates the ability to provide stable predictions across test sets that exhibit a variety of distribution shifts.

5.3 Ablation Study

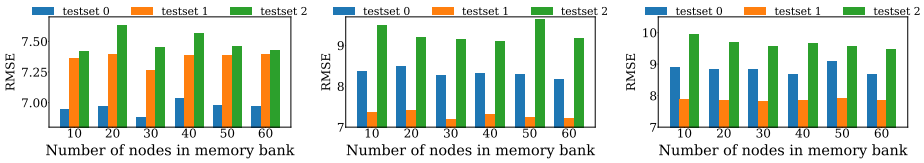
To explore the significance of each core component in MIP, we carry out an ablation study with the following variants: **Backbone** is the backbone model alone; **add cau-adj** only adds the causal adjacency matrix based on the backbone model; **add prompt** only feeds the prompt learned from the memory bank into the backbone model and omits the causal adjacency matrix and the invariant learning loss; **w/o cau-adj** removes the causal adjacency matrix; and **w/o invariant learning** removes the invariant loss.

The results are presented in Fig. 4. We can see that MIP beats all the variants on both datasets. The backbone model gets the worst results on most of the test sets, as the naive GNNs and temporal Transformer layers cannot capture the distribution shift and the heterophily of the node features. **add cau-adj** performs much better than the backbone model, even better than **add prompt**



(a) METR-LA. (b) NYCBike1 (In). (c) NYCBike1 (Out).

Fig. 5: RMSE of MIP with different intervention rates.



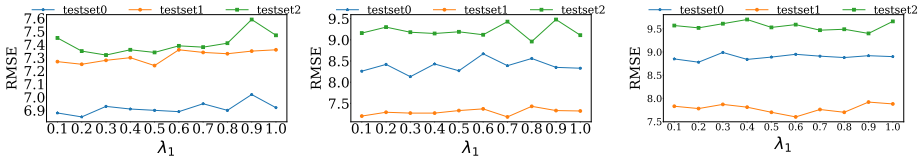
(a) METR-LA. (b) NYCBike1 (In). (c) NYCBike1 (Out).

Fig. 6: RMSE of MIP with different virtual node numbers in the memory bank.

and **w/o cau-adj** sometimes, as the causal adjacency matrix connects nodes with similar urban flow data, even if they are far away from each other in topology. **add prompt** performs better than the backbone model, as it learns some useful features from the memory bank. The performance of **w/o cau-adj** decreases much more than **MIP**. Even though it distinguishes the invariant and variant prompts with the invariant learning loss, its RMSE is bigger than the **w/o invariant learning**. As the invariant prompts are propagated to their distance-based neighbours rather than their semantic-based neighbours, some nodes receive the opposite features from their own features. **w/o invariant learning** gets worse results than **MIP** as it mixes the invariant and variant prompts together for the absence of invariant learning.

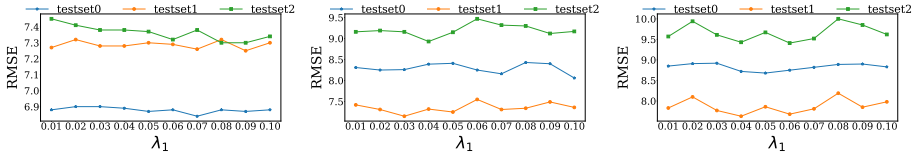
5.4 Parameter Sensitivity Analysis

Intervention rate: The intervention rate is closely related to the ability to separate the invariant and variant features. We set this parameter from 0.05 to 0.95 with an interval of 0.1, and evaluate our model on both datasets. In Fig. 5, we record the RMSE of the final horizon. The MIP demonstrates insensitivity to varying intervention rates, as evidenced by the small fluctuating RMSE across different levels of intervention. In the spatial-temporal model, the intervened variant prompts will propagate to all the nodes in an urban graph, even a small intervention rate will make all the nodes contain variant patterns before the prediction layer. Thus, the change in intervention rate does not influence the prediction RMSE.



(a) METR-LA dataset. (b) NYCBike1 dataset(In). (c) NYCBike1 dataset(Out).

Fig. 7: RMSE of MIP with different settings of λ_1.

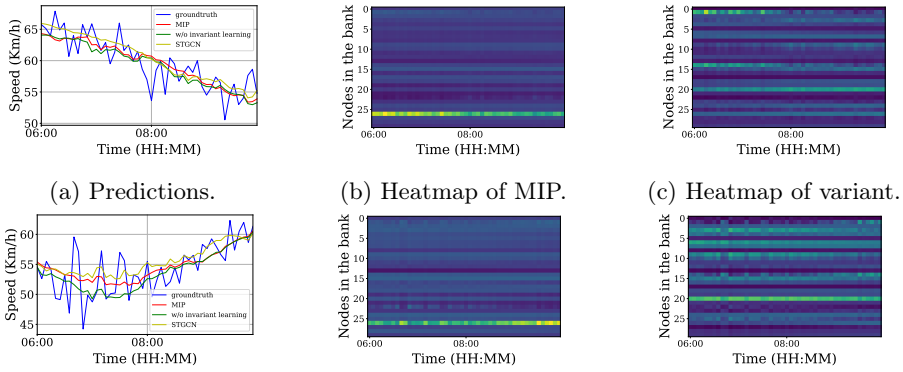


(a) METR-LA dataset. (b) NYCBike1 dataset(In). (c) NYCBike1 dataset(Out).

Fig. 8: RMSE of MIP with different settings of λ_2.

Number of nodes in memory bank: We investigate the sensitivity of the number of nodes in the memory bank and show the results in Fig. 6. The MIP performs well on all the datasets with 30 nodes in the memory bank. With a small number of nodes, MIP cannot extract high-quality invariant features due to the limited diversity. On the contrary, with more nodes in the memory bank, diverse invariant features lead to the increasing training difficulty of both the prediction model and the invariant learning backbone model.

The Weight Coefficients in The Loss Function: In Eq.(19), the loss function consists of task loss, invariant loss, and auxiliary loss. The composition ratio of the last two losses is controlled with hyperparameters λ_1 and λ_2, and we implement an experiment to investigate which one works better. Firstly, we set $\lambda_2 = 0.01$, and λ_1 from 0.1 to 1.0, with a step as 0.1, and record the RMSE of the last horizon on both datasets in Fig. 7. On all the datasets, the RMSE of the test set 0 is stable, and it fluctuates on test sets 1 and 2. Concretely, on the METR-LA dataset, the model gets better generalization ability at λ_1 is 0.3, as the RMSE on the test set 2 is the smallest. As for the NYCBike1 dataset, the RMSE also fluctuates on test set 1 and test set 2, which indicates that the MIP is not sensitive to this hyper-parameter on this dataset. Furthermore, we set $\lambda_1 = 0.1$ and λ_2 from 0.01 to 0.1, with the step as 0.01, and the results are shown in Fig. 8. On all the datasets, the RMSE on test sets 0 and 1 fluctuates. On the METR-LA dataset, the RMSE on test set 2 gradually decreases as λ_2 rises, as there are more nodes in this dataset, the proportion of $\mathcal{L}_1$ and $\mathcal{L}_2$ should be larger to make sure the invariant prompt are diversity enough for all the nodes. While on the NYCBike1 dataset, the RMSE on test set 2 reaches the low point at about 0.04 or 0.05, as the number of nodes in this dataset is less than it in METR-LA, a low proportion of $\mathcal{L}_1$ and $\mathcal{L}_2$ can make the invariant



(a) Predictions. (b) Heatmap of MIP. (c) Heatmap of variant.

(d) Predictions. (e) Heatmap of MIP. (f) Heatmap of variant.

Fig. 9: Case study on MIP's prediction performance under distribution shifts.

prompt to be diversity enough for NYCBike1 dataset. All these experiments are carried out with 30 nodes in the memory bank.

5.5 Case Study

We further conduct a case study in Fig. 9, where we randomly select a sensor in the METR-LA dataset and visualize its real and predicted traffic speeds in test sets 0 and 2. To ensure a fair comparison, the same time period on Tuesday is used for this sensor. In Fig. 9a and 9d, the tendencies of ground truths are totally different, which means distribution shift happens. Moreover, the prediction of MIP is closest to the ground truth, while the variant model and the baseline make biased predictions. Furthermore, the heatmap of two prompt scores corresponding to each sample(Fig. 9b and 9c, Fig. 9e and 9f) show completely different distributions, the prompt score of MIP tend to select more feature from a certain memory node, while the variant tends to combine features from various memory nodes. This phenomenon demonstrates that invariant learning can help the model extract invariant prompts and overcome the OOD problem.

6 Conclusion

In this paper, we introduce a new framework named MIP to solve the distribution shift problem in urban flow prediction. MIP stores the most important informative signal during the training process in a memory bank. Then, a memory-based causal graph structure is generated based on the memory bank. Furthermore, the invariant and variant prompts are extracted from the memory bank and we design a spatial-temporal intervention mechanism to create diverse distribution and propose an invariance learning regularization to help the prompt extractor

separate the invariant and variant prompts. Extensive experiments on two real-world datasets demonstrate that our method can better handle spatial-temporal distribution shifts than state-of-the-art baselines.

Acknowledgments. This work was partially supported by the Australian Research Council, under the streams of Future Fellowship (Grant No. FT210100624), Discovery Early Career Researcher Award (Grants No. DE230101033), Discovery Project (Grant No. DP240101108 and DP240101814), and Linkage Project (Grant No. LP230200892 and LP240200546).

References

1. Ahuja, K., Shanmugam, K., Varshney, K., Dhurandhar, A.: Invariant risk minimization games. In: ICML. pp. 145–155. PMLR (2020)
2. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019)
3. Bai, L., Yao, L., Li, C., Wang, X., Wang, C.: Adaptive graph convolutional recurrent network for traffic forecasting. NeurIPS **33**, 17804–17815 (2020)
4. Chen, C., Yao, F., Mo, D., Zhu, J., Chen, X.M.: Spatial-temporal pricing for ride-sourcing platform with reinforcement learning. Transportation Research Part C: Emerging Technologies **130**, 103272 (2021)
5. Chen, G., et al.: Causality and independence enhancement for biased node classification. In: CIKM. pp. 203–212 (2023)
6. Cini, A., Marisca, I., Zambon, D., Alippi, C.: Taming local effects in graph-based spatiotemporal forecasting. NeurIPS **36** (2024)
7. Cui, Y., et al.: Roi-demand traffic prediction: A pre-train, query and fine-tune framework. In: ICDE. pp. 1340–1352 (2023)
8. Deng, J., et al.: St-norm: Spatial and temporal normalization for multi-variate time series forecasting. In: SIGKDD. pp. 269–278 (2021)
9. Geng, X., et al.: Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In: AAAI. vol. 33, pp. 3656–3663 (2019)
10. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: AAAI. vol. 33, pp. 922–929 (2019)
11. Ji, J., Wang, J., Jiang, Z., Jiang, J., Zhang, H.: Stden: Towards physics-guided neural networks for traffic flow prediction. In: AAAI. vol. 36, pp. 4048–4056 (2022)
12. Ji, J., et al.: Spatio-temporal self-supervised learning for traffic flow prediction. In: AAAI. vol. 37, pp. 4356–4364 (2023)
13. Jiang, J., et al.: Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In: AAAI. AAAI Press (2023)
14. Jiang, R., et al.: Spatio-temporal meta-graph learning for traffic forecasting. In: AAAI. vol. 37, pp. 8078–8086 (2023)
15. Jiang, W., et al.: Physics-guided active sample reweighting for urban flow prediction. arXiv preprint arXiv:2407.13605 (2024)
16. Lee, H., Ko, S.: Testam: a time-enhanced spatio-temporal attention model with mixture of experts. arXiv preprint arXiv:2403.02600 (2024)
17. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)

18. Li, Z., Huang, C., Xia, L., Xu, Y., Pei, J.: Spatial-temporal hypergraph self-supervised learning for crime prediction. In: ICDE. pp. 2984–2996 (2022). https://doi.org/10.1109/ICDE53745.2022.00269
19. Liu, D., Wang, J., Shang, S., Han, P.: Msdr: Multi-step dependency relation networks for spatial temporal forecasting. In: SIGKDD. pp. 1042–1050 (2022)
20. Liu, H., Zhu, C., Zhang, D., Li, Q.: Attention-based spatial-temporal graph convolutional recurrent networks for traffic forecasting. In: International Conference on Advanced Data Mining and Applications. pp. 630–645. Springer (2023)
21. Liu, H., et al.: Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In: CIKM. pp. 4125–4129 (2023)
22. Liu, Y., et al.: Flood: A flexible invariant learning framework for out-of-distribution generalization on graphs. In: SIGKDD. pp. 1548–1558 (2023)
23. Neuberg, L.G.: Causality: models, reasoning, and inference, by judea pearl, cambridge university press, 2000. Econometric Theory **19**(4), 675–685 (2003)
24. Pearl, J.: Causal inference in statistics: a primer. John Wiley & Sons (2016)
25. Rojas-Carulla, M., Schölkopf, B., Turner, R., Peters, J.: Invariant models for causal transfer learning. Journal of Machine Learning Research **19**(36), 1–34 (2018)
26. Shang, C., Chen, J., Bi, J.: Discrete graph structure learning for forecasting multiple time series. arXiv preprint arXiv:2101.06861 (2021)
27. Shao, Z., Zhang, Z., Wang, F., Wei, W., Xu, Y.: Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In: CIKM. pp. 4454–4458 (2022)
28. Wang, B., et al.: Stone: A spatio-temporal ood learning framework kills both spatial and temporal shifts. In: SIGKDD. pp. 2948–2959 (2024)
29. Wang, Y., et al.: Gallat: A spatiotemporal graph attention network for passenger demand prediction. In: 2021 ICDE. pp. 2129–2134. IEEE (2021)
30. Wu, Q., Nie, F., Yang, C., Bao, T., Yan, J.: Graph out-of-distribution generalization via causal intervention. In: WWW. pp. 850–860 (2024)
31. Wu, Q., Zhang, H., Yan, J., Wipf, D.: Handling distribution shifts on graphs: An invariance perspective. arXiv preprint arXiv:2202.02466 (2022)
32. Wu, Y.X., Wang, X., Zhang, A., He, X., Chua, T.S.: Discovering invariant rationales for graph neural networks. arXiv preprint arXiv:2201.12872 (2022)
33. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121 (2019)
34. Wu, Z., et al.: Connecting the dots: Multivariate time series forecasting with graph neural networks. In: SIGKDD. pp. 753–763 (2020)
35. Xia, Y., et al.: Deciphering spatio-temporal graph forecasting: A causal lens and treatment. NeurIPS **36** (2024)
36. Yang, C., Wu, Q., Wen, Q., Zhou, Z., Sun, L., Yan, J.: Towards out-of-distribution sequential event prediction: A causal treatment. NeurIPS **35**, 22656–22670 (2022)
37. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. IJCAI (2018)
38. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: AAAI. vol. 31 (2017)
39. Zhang, Z., Wang, X., Zhang, Z., Li, H., Qin, Z., Zhu, W.: Dynamic graph neural networks under spatio-temporal distribution shift. NeurIPS **35**, 6074–6089 (2022)
40. Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attention network for traffic prediction. In: AAAI. vol. 34, pp. 1234–1241 (2020)
41. Zhou, Z., et al.: Maintaining the status quo: Capturing invariant relations for ood spatiotemporal learning. In: SIGKDD. pp. 3603–3614 (2023)