

Efficient Traffic Prediction through Spatio-Temporal Distillation

Anonymous submission

Abstract

Graph neural networks (GNNs) have gained considerable attention in recent years for traffic flow prediction due to their ability to learn spatio-temporal pattern representations through a graph-based message-passing framework. Although GNNs have shown great promise in handling traffic datasets, their deployment in real-life applications has been hindered by scalability constraints arising from high-order message passing. Additionally, the over-smoothing problem of GNNs may lead to indistinguishable region (node) representations as the number of layers increases, resulting in performance degradation. To address these challenges, we propose a new knowledge distillation paradigm termed LightST that transfers spatial and temporal knowledge from a high-capacity teacher to a lightweight student. Specifically, we introduce a dual-level spatio-temporal knowledge distillation framework that helps student MLPs capture graph-structured global spatio-temporal patterns while alleviating the over-smoothing effect with adaptive knowledge distillation. Extensive experiments verify that LightST significantly speeds up traffic flow predictions by $5\times$ to $40\times$ compared to state-of-the-art spatio-temporal GNNs, all while maintaining superior accuracy.

Introduction

Recent advancements in intelligent transportation systems have seen significant progress in traffic flow prediction (Wang et al. 2020; Pan et al. 2019; Liang et al. 2019) through the development of Graph Neural Networks (GNNs). GNN-based methods utilize the message-passing mechanism to propagate embeddings, enabling them to capture spatio-temporal traffic patterns. For example, STGCN (Yu, Yin, and Zhu 2018), AGCRN (Bai, Yao et al. 2020) and GWN (Shleifer, McCreery, and Chitters 2019) are built upon GNNs for traffic prediction. GCN-based models (Yu, Yin, and Zhu 2018; Shleifer, McCreery, and Chitters 2019) use convolutional operations on the graph structure to extract spatial features of traffic data, while GAT-based methods (Han and Gong 2022) employ attention mechanisms to weigh the importance of graph-based neighboring locations for each region, propagating information.

The effectiveness of spatio-temporal GNNs is largely attributed to the complex model structure and recursive message-passing architecture that encodes high-order region-wise connectives and learns region representations.

However, the increasing complexity of larger and deeper GNN model structures leads to the computationally intensive inference procedure, posing challenges for practical applications due to the scalability constraints. Therefore, a lightweight yet effective traffic prediction framework is required for practical settings of intelligent transportation systems. Additionally, propagating embeddings across multiple layers gradually makes node features more uniform, ultimately reducing the model’s ability to differentiate node features (Chen et al. 2020b; Zhou et al. 2020). This inherent recursive message-passing paradigm may fall short in encoding diverse spatio-temporal patterns, ultimately degrading traffic prediction performance.

To mitigate these challenges, existing methods (Izadi, Safayani, and Mirzaei 2024; Wang et al. 2024) leverage knowledge distillation (KD) to transfer knowledge from complex teacher models to smaller student models, helping to overcome limitations in spatio-temporal graph neural networks (ST-GNNs). However, they still face challenges and achieve sub-optimal performances. KD-pruning method (Izadi, Safayani, and Mirzaei 2024) calculates pruning scores via cost function and fine-tunes the student network decomposed with GNNs, but they do not fully address the over-smoothing issue inherent to GNNs. Furthermore, incorporating GNNs into the student network does not effectively resolve the high computational cost associated with training and deploying these models. A recent work (Wang et al. 2024) proposes a Spatial-Temporal Knowledge Distillation (STKD) algorithm framework for lightweight network traffic anomaly detection, integrating multi-scale 1D CNNs and LSTMs with identity mapping for performance enhancement. However, the individual components of STKD, namely 1D CNNs and LSTMs, may not be optimal for capturing temporal correlations, potentially limiting its effectiveness in spatio-temporal domains. Firstly, 1D CNNs function within fixed-size windows, potentially impeding their ability to capture essential long-range dependencies critical for modeling intricate temporal patterns within traffic data. Secondly, both 1D CNNs and LSTMs generate fixed-length representations, potentially compromising detailed temporal nuances essential for capturing subtle variations and complex temporal relationships in dynamic traffic environments.

In light of the motivations described above, this study

R#1vnE-W1

R#1vnE-W6

R#1vnE-W2

R#1vnE-W3

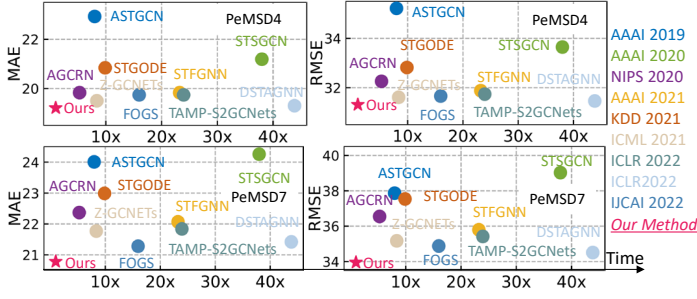


Figure 1: Model performance comparison in terms of both traffic flow prediction accuracy and inference time. Lower MAE and RMSE indicate better performance. The symbol 40× indicates that our LightST runs 40 times faster than a reference baseline method measured by inference time.

aims to: i) develop a traffic prediction model that is highly scalable while effectively capturing complex spatio-temporal dependency patterns across various locations and time periods; and ii) enhance the spatio-temporal encoding function to mitigate the issue of over-smoothing. To achieve this, we propose a dual-level spatio-temporal knowledge distillation paradigm that effectively transfers complex dynamic spatio-temporal dependency knowledge into a compact and fast-to-execute student model. Specifically, in the distillation procedure, the soft prediction labels from the teacher GNN guide the learning of the student model. This transfer of knowledge effectively incorporates structural information from both spatial and temporal aspects. Furthermore, to avoid using a uniform alignment factor for all region pairs, we propose an adaptive embedding-level distillation framework that enhances the knowledge transfer while mitigating over-smoothing effects. Our empirical studies, as shown in Figure 1, suggest that our designed spatio-temporal knowledge distillation substantially benefits traffic prediction performance in terms of both forecasting accuracy and efficiency. Our key contributions are listed as follows:

- To overcome the computational and oversmoothing challenges in state-of-the-art GNN-based traffic prediction models, we propose distilling the complex spatio-temporal GNN architecture into a streamlined MLP model, enhancing both efficiency and robustness in traffic prediction.
- We design a new spatio-temporal knowledge distillation paradigm with two model alignment levels, enabling the transfer of knowledge related to spatial and temporal dynamics while enhancing the student model with a global context. Additionally, we propose an adaptive contrastive distillation scheme to further enhance the robustness of the prediction model against the over-smoothing issue.
- We empirically validate our new framework on 5 real-world traffic datasets. Evaluation results demonstrate that our model achieves state-of-the-art traffic prediction accuracy, with a 5× to 40× inference speedup compared to existing baselines. Our codes are available at: <https://anonymous.4open.science/r/LightST-43EB/README.md>. We also attached the reproducibility

checklist.

Methodology

Spatio-temporal Graph and Traffic Data

Following established practices (Lan et al. 2022; Chen, Segovia, and Gel 2021), we define our spatial graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the geographically-adjacent relationships between traffic sensing regions. We represent the traffic volume data across both spatial and temporal dimensions using a matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$, where N denotes the number of regions and T represents the number of time slots. Each element $x_{n,t}$ within \mathbf{X} corresponds to the traffic volume information of region n during time slot t .

Problem Formulation. Our goal is to develop a function \mathcal{F} that predicts future traffic flow $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times H}$ based on observed traffic flow data $\mathbf{X} = (x_1, \dots, x_T) \in \mathbb{R}^{N \times T}$. The ground truth is $\mathbf{Y} = (x_{T+1}, \dots, x_{T+H}) \in \mathbb{R}^{N \times H}$. Here, \mathbf{X} represents the observed traffic flow from N sensing regions within a sensor graph \mathcal{G} over the preceding T time slots, and $\hat{\mathbf{Y}}$ denotes the predicted traffic flow for the subsequent H time steps. Our approach aims to learn \mathcal{F} while preserving both spatial and temporal dependencies within the data. This is shown as $\hat{\mathbf{Y}} = \mathcal{F}(\mathbf{X}; \mathcal{G})$.

Spatio-Temporal Graph Neural Networks

Data Scale. Graph Neural Networks (GNNs) have demonstrated significant potential in learning from spatio-temporal data (Li et al. 2018; Lan et al. 2022; Rao et al. 2022). Inspired by this, our teacher model leverages a graph-based message passing framework that harnesses the power of GNNs to capture region-wise dependencies. We begin by mapping traffic data into a latent representation space. Each element $\mathbf{X}_{n,t} \in \mathbf{X}$ is encoded into an embedding $\mathbf{E}_{n,t}^{(d)} \in \mathbb{R}^d$ as follows:

$$\mathbf{E}_{n,t}^{(d)} = \text{Z-Score}(x_{n,t}) \cdot \mathbf{e} = \frac{x_{n,t} - \mu}{\sigma} \cdot \mathbf{e}. \quad (1)$$

We normalize the value of $\mathbf{X}_{n,t}$ using the Z-Score function, which centers the data around zero with a standard deviation of one. This normalized value is then multiplied by a base embedding vector $\mathbf{e} \in \mathbb{R}^d$, resulting in the final embedding $\mathbf{E}_{n,t}^{(d)} \in \mathbb{R}^d$. μ represents the average traffic flow value of node n over the last 12 time steps. And σ denotes the standard deviation of the traffic flow values of node n over the previous 12 time steps. The base embedding vector \mathbf{e} acts as a template, and the Z-Score normalization scales the traffic flow value, effectively creating a unique embedding for each data point based on its relative position within the normalized distribution.

Time-aware Spatial Message Passing. To capture time-aware spatial dependencies and learn representations specific to each time slot t , we employ a time-specific message passing mechanism among traffic sensing regions. This process involves aggregating information from neighboring nodes within the sensor graph. The embedding of region n at the l -th GNN layer is denoted by $\mathbf{E}_n^{(l)}$, and is shown as follows:

R#nHf6-Q3

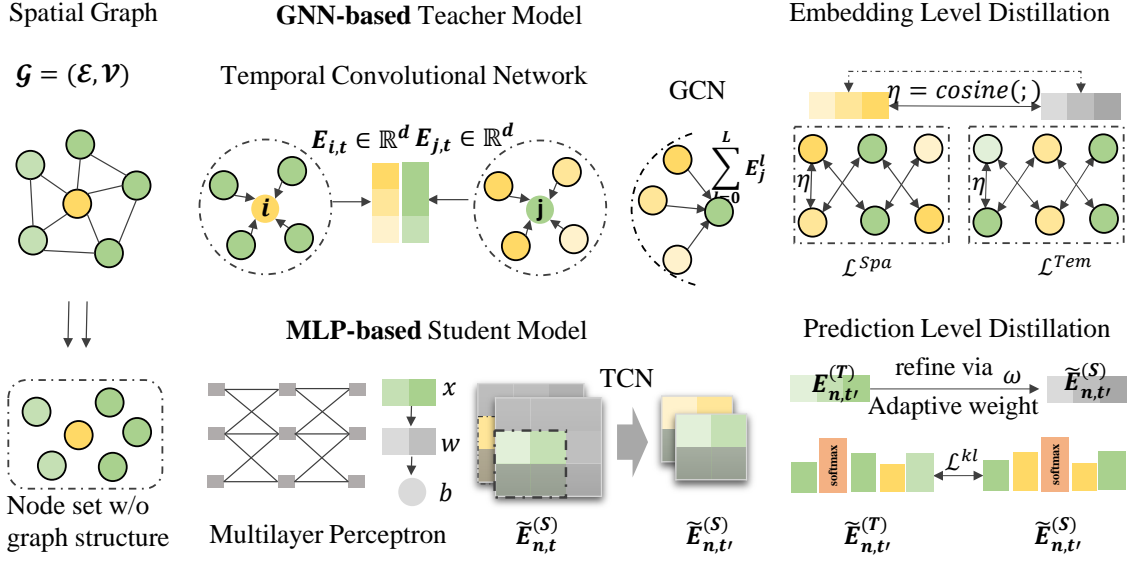


Figure 2: Our proposed spatio-temporal knowledge distillation framework, LightST, comprises two main components: a GNN-based teacher model and an MLP-based student model. The distillation paradigm itself is structured in two parts: embedding-level distillation and prediction-level distillation.

$$\mathbf{E}_n^{(l)} = \sigma^{(1)}\left(\sum_{j \in \mathcal{N}_n} \alpha_{n,j} \mathbf{W}^{(l-1)} \mathbf{E}_j^{(l-1)}\right), \quad (2)$$

$$\alpha_{n,j} = \frac{1}{\sqrt{|\mathcal{N}_n| |\mathcal{N}_j|}}.$$

Here, \mathcal{N}_n and \mathcal{N}_j represent the sets of neighboring nodes for regions n and j , respectively. The embedding propagation mechanism aggregates information from these neighboring nodes. $\sigma^{(1)}(\cdot)$ represents the ReLU activation function. The normalization weight $\alpha_{n,j} \in \mathbb{R}$ for a node pair (n, j) is calculated based on the degrees of the nodes. This cross-layer information propagation and aggregation can be formalized in matrix form using the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ of our spatial graph \mathcal{G} :

$$\mathbf{E}^{(l)} = \sigma^{(1)}(\mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{E}^{(l-1)} \mathbf{W}^{(l-1)\top}). \quad (3)$$

The embedding matrix $\mathbf{E} = \sum_{l=0}^L \mathbf{E}^{(l)} \in \mathbb{R}^{N \times d}$ contains node embeddings, where each row represents individual traffic sensing point \mathbf{E}_n ($1 \leq n \leq N$). $\mathbf{D} \in \mathbb{R}^{N \times N}$ denotes the diagonal degree matrix, and L represents the number of graph neural iterations. The final aggregated region representations are given by $\mathbf{E} \in \mathbb{R}^{N \times d}$.

Temporal Encoder Layer. To capture temporal dependencies across all sensor regions, we employ a two-layer Temporal Convolutional Network (TCN) to model temporal correlations. This can be represented as follows:

$$\tilde{\mathbf{E}}_n = \sigma^{(2)}(\delta(\mathbf{W} * \mathbf{E}_n + \mathbf{b}) + \mathbf{E}_t), t \in [1, T]. \quad (4)$$

Here, the embedding matrix of all N regions at the time slot of t is represented by $\mathbf{E}_t \in \mathbb{R}^{N \times d}$. For a specific region n , the embedding matrix is represented by $\mathbf{E}_n \in \mathbb{R}^{T \times d}$, which captures the embedding of region n across the previous T time slots. The temporal convolution kernel and bias

are represented by $\mathbf{W} \in \mathbb{R}^{f \times d}$ and $\mathbf{b} \in \mathbb{R}^d$, respectively. These are learnable transformation parameters. Here, f denotes the kernel size. The operation $*$ denotes the temporal convolution, while the dropout $\delta(\cdot)$ and LeakyReLU activation function $\sigma^{(2)}(\cdot)$ are used as well. The output representation $\tilde{\mathbf{E}} \in \mathbb{R}^{N \times T \times d}$ is generated with two-layer TCNs that serve as the temporal encoder in the teacher model, so as to capture the time-evolving traffic patterns.

Dual-Level Spatio-Temporal (ST) Distillation

This research aims to develop a *lightweight* and *efficient* traffic predictor that leverages spatial and temporal knowledge extracted from a Graph Neural Network (GNN) teacher model to achieve accurate traffic predictions. This is achieved through both explicit prediction-level and implicit embedding-level knowledge transfer from the GNN teacher model to an MLP-based student model. We provide details of dual-level ST distillation as follows:

Prediction-Level Distillation. During this stage, we aim to align the probability distributions of the teacher and student models, thereby enhancing the predictive performance of the student model. This is achieved through knowledge distillation, where the output results of the teacher model are used as soft labels to guide the student model. We minimize the difference between the probability distributions of the teacher and student models by employing an MSE-based loss function (Wu et al. 2020). The MSE-based loss functions for the teacher and student models in traffic prediction

are defined as follows:

$$\begin{aligned}\mathcal{L}^{(T)} &= \frac{1}{N} \sum_{n=1}^N \sum_{t'=T+1}^{T+H} (\hat{\mathbf{Y}}_{n,t'}^{(T)} - \mathbf{Y}_{n,t'})^2, \\ \mathcal{L}^{(S)} &= \frac{1}{N} \sum_{i=1}^N \sum_{t'=T+1}^{T+H} (\hat{\mathbf{Y}}_{n,t'}^{(S)} - \mathbf{Y}_{n,t'})^2,\end{aligned}\quad (5)$$

Here, $\mathcal{L}^{(T)}$ represents the loss of the GNN-based teacher model, denoted by the superscript (T). Similarly, $\mathcal{L}^{(S)}$ represents the loss of the MLP-based student model, denoted by the superscript (S). N corresponds to the number of sensing regions within the spatio-temporal graph \mathcal{G} , and H denotes the number of predicted time steps. $\hat{\mathbf{Y}}_{n,t'}^{(T)}$ represents the predicted traffic volume of node n at time step t' by the teacher model, while $\hat{\mathbf{Y}}_{n,t'}^{(S)}$ represents the predicted traffic volume of the same node and time step by the student model. $\mathbf{Y}_{n,t'}$ denotes the ground truth traffic volume of node n at time step t' . To further measure the distribution difference of the GNN-based teacher model and that of the MLP-based student model, we adopt Kullback Leibler (KL) (Hu and Hong 2013) divergence into the prediction-level loss $\mathcal{L}^{(KL)}$:

$$\mathcal{L}^{(KL)} = \sum_{n=1}^N \sum_{t'=T+1}^{T+H} \tilde{\mathbf{E}}_{n,t'}^{(T)} \cdot \log(\text{Softmax}(\tilde{\mathbf{E}}_{n,t'}^{(S)})). \quad (6)$$

Here, $\mathcal{L}^{(KL)}$ represents the Kullback-Leibler (KL) divergence loss, denoted by the superscript (KL). $\tilde{\mathbf{E}}_{n,t'}^{(T)}$ denotes the output of the Temporal Convolutional Network (TCN) for node n at time step t' from the teacher model, while $\tilde{\mathbf{E}}_{n,t'}^{(S)}$ represents the corresponding output from the student model. Minimizing $\mathcal{L}^{(KL)}$ aims to align the predicted probability distribution of the student model with that of the teacher model. This process effectively incorporates spatio-temporal knowledge from the teacher into the student's predictions.

Embedding-Level Distillation. To further transfer spatio-temporal knowledge from the representation space, we introduce embedding-level distillation. Our model aims to achieve consistency in the embeddings of the same region from both the teacher and student models. To account for variations in the spatio-temporal context within the latent semantic space, we assign different consistency strengths to embeddings of different region pairs. This is achieved using a similarity function denoted by $\eta(\cdot)$, where we leverage embedding cosine similarity.

Our approach, which involves embedding-level distillation from both the spatial (with loss $\mathcal{L}^{(P)}$) and temporal (with loss $\mathcal{L}^{(E)}$) dimensions using contrastive learning, is formally presented as follows:

$$\begin{aligned}\mathcal{L}^{(P)} &= \sum_{n=1}^N \sum_{t'=T+1}^{T+H} -\log \frac{\exp(\frac{\eta(\tilde{\mathbf{E}}_{n,t'}^{(S)}, \mathbf{E}_{n,t'}^{(T)})}{\tau_2})}{\sum_{n' \neq n} \exp(\frac{\eta(\tilde{\mathbf{E}}_{n',t'}^{(S)}, \mathbf{E}_{n,t'}^{(T)})}{\tau_2})}, \\ \mathcal{L}^{(E)} &= \sum_{n=1}^N \sum_{t'=T+1}^{T+H} -\log \frac{\exp(\frac{\eta(\tilde{\mathbf{E}}_{n,t'}^{(S)}, \tilde{\mathbf{E}}_{n,t'}^{(T)})}{\tau_3})}{\sum_{n' \neq n} \exp(\frac{\eta(\tilde{\mathbf{E}}_{n,t'}^{(S)}, \tilde{\mathbf{E}}_{n',t'}^{(T)})}{\tau_3})}.\end{aligned}\quad (7)$$

The parameters τ_2 and τ_3 control the temperature of the softmax function used in the contrastive loss during training. $\mathcal{L}^{(P)}$ and $\mathcal{L}^{(E)}$ represent the spatial loss and temporal loss, respectively, denoted by the superscripts (P) and (E). $\mathbf{E}_{n,t'}^{(T)}$ represents the output embedding from the GCN layers of the teacher model for region n at time step t' . Similarly, $\tilde{\mathbf{E}}_{n,t'}^{(T)}$ represents the output embedding from the TCN layers of the teacher model for the same region and time step. Finally, $\tilde{\mathbf{E}}_{n,t'}^{(S)}$ denotes the output embedding from TCN layers of the student model for region n at time step t' .

Model Optimization. Following the learning paradigm of knowledge distillation, we first train the GNN-based teacher model of LightST until convergence using the loss function $\mathcal{L}^{(T)}$ from Equation 5. This involves feeding mini-batches of traffic observation tensors into the model and optimizing it. We then perform joint training to optimize both the MLP-based student model and the teacher model together. The overall objective function is an integration of the optimized objectives, which is shown as follows:

$$\mathcal{L} = \mathcal{L}^{(S)} + \lambda_1 \cdot \mathcal{L}^{(KL)} + \lambda_2 \cdot (\mathcal{L}^{(P)} + \mathcal{L}^{(E)}), \quad (8)$$

where λ_1 and λ_2 are loss weights. The training process of our LightST is elaborated in Algorithm 1 in Appendix A.1.

In-Depth Model Discussion

Model Complexity. To evaluate the efficiency improvement of the proposed MLP-based student model, we analyzed its time complexity in comparison to the GNN-based teacher model. The teacher model has a higher time complexity due to its graph information propagation in the encoder, which requires $\mathcal{O}(|\mathcal{E}| \times L \times d)$, where $|\mathcal{E}|$ is the number of edges, and L is the number of graph layers. In contrast, the proposed student model only requires $\mathcal{O}(B \times L' \times d^2)$ for CL-enhanced embedding-level distillation, where B is the number of samples in each batch, and L' is the number of MLP layers. In summary, our analysis shows that the proposed MLP-based student model is significantly more efficient than the GNN-based traffic prediction methods, making it a promising framework for large-scale traffic data in practical spatio-temporal data mining scenario.

Model Theoretical Analysis. We discuss how adaptive spatio-temporal distillation can alleviate the over-smoothing effects in spatio-temporal GNNs. We will begin by introducing the message passing schema that is used to propagate information along the graph-structured path in our spatial

graph \mathcal{G} :

$$\mathbf{E}_n^{(T)(L)} = \sum_{j \in \mathcal{N}_n} \left(\sum_{\mathcal{P}_{n,j}^L} \prod_{n_k, n_h \in \mathcal{P}_{n,j}^L} \frac{1}{\sqrt{d_k d_h}} \right) \cdot \mathbf{E}_j^{(T)(0)}, \quad (9)$$

where $\mathcal{P}_{n,j}^L$ represents the maximum length L of a possible path from the n -th region node and the j -th region node, with the intermediate connection nodes k and h . The variables d_k and d_h refer to the degrees of these intermediate nodes. From the above Eq 9, it is important to note that the weight of $\mathbf{E}_j^{(T)(0)}$ is non-learnable, which means it cannot be adjusted during the message passing process over noisy graph structures when generating the region embedding $\mathbf{E}_n^{(T)(L)}$. \mathcal{N}_n denotes the set of neighbour nodes of region n . Our framework provides a solution to this issue by introducing a learnable and adaptive knowledge distillation approach. Here, we analyze the gradients of our knowledge distillation with KL divergence alignment objective \mathcal{L}^{KL} given the corresponding embeddings $\tilde{\mathbf{E}}_n^{(S)}$ of region n via the student model as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}^{\text{KL}}}{\partial \tilde{\mathbf{E}}_n^{(S)}} &= \sum_{n=1}^N \sum_{t'=T+1}^{T+H} \omega \cdot \frac{\partial (\tilde{\mathbf{E}}_{n,t'}^{(T)}, \tilde{\mathbf{E}}_{n,t'}^{(S)})}{\tilde{\mathbf{E}}_{n,t'}^{(S)}}, \\ \omega &= \frac{1}{\text{softmax}(\tilde{\mathbf{E}}_{n,t'}^{(S)})} \left(-\frac{e^{\tilde{\mathbf{E}}_{j,t'}^{(S)}}}{\sum_{n'=1}^N e^{\tilde{\mathbf{E}}_{n',t'}^{(S)}}} \right) \left(-\frac{e^{\tilde{\mathbf{E}}_{n,t'}^{(S)}}}{\sum_{n'=1}^N e^{\tilde{\mathbf{E}}_{n',t'}^{(S)}}} \right). \end{aligned} \quad (10)$$

Here, $\tilde{\mathbf{E}}_{n',t'}^{(S)}$ represents the region embedding obtained from the TCN layers of the student model, denoted by the superscript (S) . The subscripts n', t' indicate node n' at time step t' . Similarly, $\tilde{\mathbf{E}}_{n,t'}^{(T)}$ represents the region embedding of node n at time step t' obtained from the TCN layers of the teacher model. The derivations in Equation 10 demonstrate that the region embeddings are refined through the transferred knowledge from the teacher model using the derived weight ω . While recursive message passing can lead to over-smoothing of the representations, our framework automatically adapts the knowledge transfer process, mitigating these over-smoothing effects. Due to space constraints, the detailed derivations are provided in the Appendix A.3.

Evaluation

Experimental Setting

Datasets. In this study, we conduct a series of experiments using real-life traffic flow datasets from California, specifically the PEMS3, PEMS4, PEMS7, PEMS8 and PeMS-Bay datasets released by (Song et al. 2020). The traffic data is aggregated into 5-minute time intervals, resulting in 12 points of data per hour. Additionally, we construct the spatial graph of our traffic sensing regions based on the road network, which models the relationships between traffic flow patterns in different regions of the city. More details about the experimental datasets are provided in Table 5 in Appendix A.2.

Baselines. We conduct a comprehensive evaluation of LightST by comparing it against 18 baselines, including

many state-of-the-art GNN-based traffic prediction models which are categorized into six groups: 1) **GCN-based methods**: STGCN (Yu, Yin, and Zhu 2018), DCRNN (Li et al. 2018), GWN (Shleifer, McCreery, and Chitters 2019); 2) **GAT-based approaches**: ASTGCN (Zhu et al. 2021), LSTGCN (Han and Gong 2022), DSTAGNN (Lan et al. 2022); 3) **Differential GNNs**: STG-ODE (Fang et al. 2021); 4) **GNNs enhanced with Zigzag Persistence**: Z-GCNETs (Chen, Segovia, and Gel 2021) and TAMP (Chen, Segovia-Dominguez et al. 2021); 5) **Hybrid spatio-temporal GNNs**: FOGS (Rao et al. 2022), AGCRN (Bai, Yao et al. 2020), STSGCN (Song et al. 2020), STFGNN (Li and Zhu 2021). 6) **Distillation methods**: KD-Pruning (Izadi, Safayani, and Mirzaei 2024) and STKD (Wang et al. 2024). 7) **MLP-based methods**: STID (Shao et al. 2022) and ST-MLP (Wang et al. 2023). 8) **Transformer-based methods**: STAEformer (Liu et al. 2023). Details of the baseline methods in Appendix A.2. We also provide evaluation platform and protocols in Appendix A.4.

Effectiveness Evaluation

We evaluate the effectiveness of our method, LightST, and the baselines on 5 datasets in terms of MAE, MAPE, and RMSE metrics, as shown in Table 1. Based on results, we have following observations:

Superior Prediction Accuracy. Our proposed method has consistently outperforms other baselines across all four datasets, in most evaluation cases. This can be attributed to several key factors that contribute to the effectiveness of our approach. *Firstly*, we are able to successfully distill spatial and temporal dynamics from the teacher model. This enables the student to capture time-evolving traffic dependencies across geographical regions and time slots without relying on cumbersome message passing frameworks. *Secondly*, our adaptive knowledge distillation method, realized through our dual-level knowledge transfer, guides student learning with appropriate knowledge to alleviate the over-smoothing effects of the GNN architecture. *Third*, by enabling cross-region and cross-time dependency modeling in an adaptive manner, our spatio-temporal knowledge distillation alleviates the effects of noisy adjacent relationships, contributing to the robustness of traffic flow prediction.

Performance Comparison among Baselines. Among the various baselines, we observe that methods such as STSGCN and STFGNN, which incorporate time-aware spatial dependency, perform better than approaches such as STGCN and DCRNN, which only consider stationary spatial correlations among regions. This highlights the importance of capturing temporal dynamics when encoding spatial dependency relationships among regions. In contrast to distillation methods like STKD and KD-pruning, which incorporate LSTM or GNNs into student models, potentially leading to over-smoothing and suboptimal performance, TCNs demonstrate superior efficacy in capturing temporal dynamics compared to LSTMs. Our proposed spatio-temporal knowledge distillation paradigm is designed to transfer time-aware spatial dependency knowledge from the teacher model to the MLP student model. By doing so, the student model is able to capture complex spatio-temporal patterns of traffic flow,

Table 1: Overall performance of traffic prediction on PeMS-Bay, PeMSD4, PeMSD8, PeMSD3 and PeMSD7

Models	PeMS-Bay			PeMSD4			PeMSD8			PeMSD3			PeMSD7		
	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MPAE ↓	MAE ↓	RMSE ↓	MPAE ↓	MAE ↓	RMSE ↓	MAPE ↓	MAE ↓	RMSE ↓	MAPE ↓
HA	2.88	5.59	6.82%	38.03	59.24	27.88%	34.86	52.04	24.07%	31.58	52.39	33.78%	45.12	65.64	24.51%
VAR	2.32	5.25	5.61%	24.54	38.61	17.24%	19.19	29.80	13.10%	23.65	38.26	24.51%	50.22	75.63	32.22%
DSANet	2.16	4.97	5.54%	22.79	35.77	17.12%	17.14	26.96	11.32%	21.29	34.55	23.21%	31.36	49.11	14.43%
DCRNN	2.07	4.74	4.90%	24.70	38.12	14.17%	17.86	27.83	11.45%	17.99	30.31	18.34%	25.22	38.61	11.82%
STGCN	2.42	5.33	5.58%	22.70	35.55	14.59%	18.02	27.83	11.40%	17.55	30.42	17.34%	25.33	39.34	11.21%
GWN	1.95	4.52	4.63%	25.45	39.70	17.29%	19.13	31.05	12.68%	19.12	32.77	18.89%	26.39	41.50	11.97%
ASTGCN	2.10	4.77	5.30%	22.93	35.22	16.56%	18.25	28.06	11.64%	17.34	29.56	17.21%	24.01	37.87	10.73%
LSGCN	2.13	4.82	5.18%	21.53	33.86	13.18%	17.73	26.76	11.30%	17.94	29.85	16.98%	27.31	41.16	11.98%
STSGCN	2.10	4.74	5.28%	21.19	33.65	13.90%	17.13	26.86	10.96%	17.48	29.21	16.78%	24.26	39.03	10.21%
AGCRN	1.96	4.57	4.69%	19.83	32.26	12.97%	15.95	25.22	10.09%	15.98	28.25	15.23%	22.37	36.55	9.12%
STFGNN	1.83	4.33	4.19%	19.83	31.88	13.02%	16.64	26.22	10.60%	16.77	28.34	16.30%	22.07	35.80	9.21%
STGODE	2.02	4.40	4.72%	20.84	32.82	13.77%	16.81	25.97	10.62%	16.50	27.84	16.69%	22.99	37.54	10.14%
Z-GCNETs	2.03	4.38	4.71%	19.50	31.61	12.78%	15.76	25.11	10.01%	16.64	28.15	16.39%	21.77	35.17	9.25%
TAMP	2.04	4.45	4.76%	19.74	31.74	13.22%	16.36	25.98	10.15%	16.46	28.44	15.37%	21.84	35.42	9.24%
FOGS	2.07	4.51	4.80%	19.74	31.66	13.05%	15.73	24.92	9.88%	15.89	25.74	15.13%	21.28	34.88	8.95%
DSTAGNN	2.13	4.79	5.32%	19.30	31.46	12.72%	15.67	24.77	9.94%	15.57	27.21	14.68%	21.42	34.51	9.01%
STKD	2.08	4.56	4.82%	19.86	31.93	13.18%	15.81	25.07	10.02%	16.03	25.95	15.76%	21.64	34.96	9.03%
KD-Pruning	2.23	4.97	5.34%	21.22	34.63	14.15%	17.46	27.09	11.74%	17.12	29.87	17.06%	24.55	38.17	11.90%
STID	1.81	3.96	4.37 %	19.58	31.79	13.38%	15.58	25.89	10.33 %	15.45	26.84	16.25%	21.03	35.88	9.06 %
ST-MLP	1.90	4.34	4.45 %	19.23	31.54	12.89%	15.47	25.48	10.17 %	15.37	26.48	15.84%	21.43	35.86	9.19 %
STAEformer	1.80	3.92	4.33 %	19.23	31.48	12.83 %	15.45	25.01	9.92%	15.35	27.55	15.18 %	19.69	33.09	8.38%
LightST(Ours)	1.78	3.88	4.15%	19.21	31.31	12.70%	15.43	24.52	9.84%	15.11	24.74	14.41%	20.78	33.95	8.98%

resulting in state-of-the-art traffic prediction performance. We also provide multi-step results in Appendix A.4.

Model Scalability Investigation

To evaluate the efficiency of our proposed LightST, we conduct experiments on the large PeMSD7 dataset competing with several state-of-the-art baselines. We conduct the experiments on a server with 10 cores of Intel(R) Core(TM) i9-9820X CPU @ 3.30GHz, 64.0GB RAM, and 4 Nvidia GeForce RTX 3090 GPU. The results for inference time and forecasting accuracy are shown in Table 2. Our analysis yields two key observations. First, our LightST achieves competitive performance in terms of accuracy metrics, *i.e.*, MAE, MAPE, and RMSE. This is particularly noteworthy given the potential for over-smoothing on large-scale spatial region graphs, which our framework avoids by not explicitly introducing graph message passing and instead distilling denoised spatio-temporal knowledge into graph-less designations. Second, our LightST achieves much faster inference time than the compared baseline models, which is attributed to the fact that LightST does not require recursive graph-based information propagation operations during inference phase. While our traffic flow predictor is a simple graph-less neural network, its achieved superior performance suggests the effectiveness of our knowledge distillation paradigm in injecting complex global spatio-temporal dependencies across high-order region and time connections into the student. The ability to achieve high accuracy with fast inference time is particularly important in practical applications, where traffic forecasting models need to operate in real-time urban sensing.

Table 2: Model Efficiency Study

Datasets			PeMSD7		
Method	MAE ↓	RMSE ↓	MAPE ↓	Inference ↓	Faster x ↑
ASTGCN	24.01	37.87	10.73%	20.06s	7.99 ×
STFGNN	22.07	35.80	9.21%	53.81s	21.44 ×
STGODE	22.99	37.54	10.14%	24.79s	9.88 ×
DSTAGNN	21.42	34.51	9.01%	110.06s	43.85 ×
STID	21.03	35.88	9.06%	12.35s	4.92 ×
ST-MLP	21.43	35.86	9.19%	5.23s	2.08 ×
STAEformer	19.69	33.09	8.38%	53.26s	21.22 ×
Ours	20.78	33.95	<u>8.98%</u>	2.51s	-

Datasets			PeMS-Bay		
Method	MAE ↓	RMSE ↓	MAPE ↓	Inference ↓	Faster x ↑
ASTGCN	2.10	4.77	5.30	40.08s	7.82 ×
STFGNN	1.83	4.33	4.19%	98.18s	19.18 ×
STGODE	2.02	4.40	4.72%	208.51s	40.72 ×
DSTAGNN	2.13	4.79	5.32%	86.97s	16.99 ×
STID	1.81	3.96	4.37%	22.18s	4.33 ×
ST-MLP	1.90	4.34	4.45%	13.52s	2.64 ×
STAEformer	1.80	3.92	4.33%	28.48s	5.56 ×
Ours	1.78	3.88	4.15%	5.12s	-

Ablation Study and Effectiveness Analyses

To assess the impact of each component in our knowledge distillation framework on prediction results and speed, we conducted an ablation study across four traffic datasets using model variants (see Figures 3 and 6 in the Appendix A.4). These variants include: 1) *w/o E-KD*, which disables embedding-level knowledge distillation for transferring spatio-temporal signals from the latent representation space; 2) *w/o E-S*, which omits adaptive embedding alignment with spatial information by removing $\mathcal{L}^{(P)}$ from the

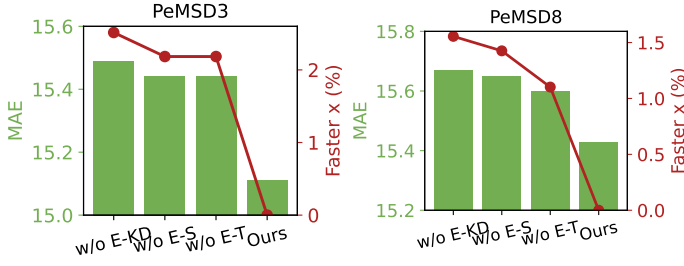


Figure 3: Ablation study of sub-modules in our spatio-temporal knowledge distillation paradigm.

joint loss \mathcal{L} ; and 3) *w/o E-T*, which excludes $\mathcal{L}^{(E)}$ from \mathcal{L} to avoid capturing temporal information during embedding-level knowledge distillation. The results, presented in Figure 3, reveal key observations. Firstly, the *w/o E-KD* variant performs significantly worse than our full model, highlighting the crucial role of embedding-level knowledge distillation in transferring spatio-temporal signals. Notably, KL divergence is the most computationally demanding component of our model. Secondly, the superior performance of our model compared to *w/o E-S* and *w/o E-T* demonstrates the effectiveness of adaptive embedding alignment across both spatial and temporal domains in capturing complex cross-location and cross-time traffic dependencies.

Table 3: Multi-backbone Comparison

	PeMSD4				
	MAE	RMSE	MAPE	Inference Time	Faster x
AGCRN	19.76	32.53	13.15%	25.72s	10.08x
AGCRN+ours	19.35	31.57	12.93%	2.55s	-
MTGNN	19.55	31.78	13.03%	23.81s	13.68x
MTGNN+ours	19.21	31.31	12.70%	1.74s	-

Study of Over-Smoothing Alleviation

In this section, our aim is to evaluate the effectiveness of our framework, LightST, in addressing the issue of over-smoothing. We use Mean Average Distance (MAD) to test the smoothness of node representations (Chen et al. 2020a). To compare the results of our framework with state-of-the-art methods, we provide the comparison of 5 methods on four datasets in Table 4. From the table, it is evident that our method, LightST, outperforms several state-of-the-art baselines in terms of MAD values on the PeMSD4 and PeMSD8 datasets. However, GNN-based methods, especially those with deeper message passing layers, *e.g.*, differential GNNs-STGODE, GAT-based methods-ASTGCN & DSTAGNN, are likely to lead to severe over-smoothing issues after recursively propagating region embeddings. In contrast, LightST is resilient to over-smoothing due to our dual-level knowledge transfer. This results in more distinguishable region embeddings, which preserve uniform spatial and temporal dependent signals.

Hyperparameter Study

The aim of this section is to evaluate the effects of key hyperparameters on the performance of our framework, LightST. We present our results on PeMSD8 and PeMSD3 datasets

Table 4: Study of over-smoothing alleviation on PeMSD4 and PeMSD8 in terms of MAD (higher scores indicates less over-smoothing effect).

Data	ASTGCN	AGCRN	STFGNN	STGODE	DSTAGNN	LightST(Ours)
PeMSD4	0.30	0.51	0.53	0.32	0.22	0.66
PeMSD8	0.31	0.52	0.55	0.34	0.23	0.68

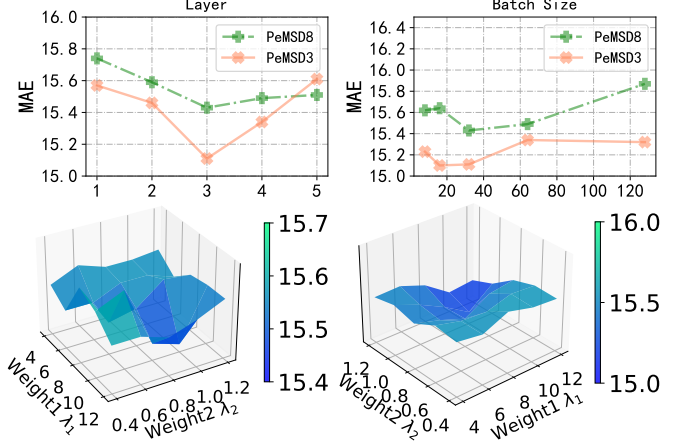


Figure 4: Hyperparameter study on PeMSD8 and PeMSD3 in terms of MAE.

in terms of MAE and RMSE in Figure 4 (More results in Figure 7 shown in Appendix A.4). We summarize our observations as follows: 1) Figure 4 show the effect of the number of MLP layers (ranging from $\{1, 2, 3, 4, 5\}$) and varying batch size (ranging from $\{2^3, 2^4, 2^5, 2^6, 2^7\}$) on performance. Our framework, LightST, achieves the best performance on PeMSD8 and PeMSD3 when the number of layers is 3 and the batch size is 32. Even when LightST achieves the worst performance, it still outperforms most of the baselines. These results suggest that the performance of our LightST is not sensitive to the MLP depth and batch size. 2) λ_1, λ_2 serve as loss weights to control how strongly our prediction-level and embedding-level restrict the joint model training. Figure 4 show that λ_1 and λ_2 jointly affect the strength of the optimization of knowledge distillation. **And the temperature is set as 0.4 when our method obtains the best performance.** We find that a larger weight of distillation causes performance maintenance, enabling MLP to learn sufficient knowledge.

Related Work

Traffic Flow Prediction. Numerous neural network architectures have been proposed for traffic prediction, including convolutional neural networks (CNNs) (Zhang, Zheng, and Qi 2017), recurrent neural networks (RNNs) (Lv et al. 2018), attention mechanisms (Yao et al. 2019), and graph neural networks (GNNs) (Li and Zhu 2021). CNNs have proven effective in modeling citywide traffic maps as images for spatio-temporal pattern encoding (Diao et al. 2019; Zhang, Zheng, and Qi 2017), while RNNs excel at capturing temporal dependencies in time-evolving traffic data (Lv et al. 2018). To model spatial traffic similarities adaptively, research has explored spatial dependency graphs with learn-

R#1vnE-W7
R#nHf6-Q2

able region adjacency matrices (Shleifer, McCreery, and Chitters 2019; Bai, Yao et al. 2020; Rao et al. 2022; Lan et al. 2022). For instance, DSTAGNN (Lan et al. 2022) utilizes a multi-head attention mechanism to exploit spatial correlations with multi-scale neighborhoods. FOGS (Rao et al. 2022) learns a spatial-temporal correlation graph using first-order gradients during training. While GNN-enhanced traffic prediction models hold promise, their computational complexity hinders scalability and real-world deployment. This study addresses this challenge by leveraging spatio-temporal knowledge distillation to reduce inference time, enabling our model to effectively scale to larger datasets.

Recent traffic prediction methods have employed Multilayer Perceptron (MLP)-based networks, including STID (Shao et al. 2022) and ST-MLP (Wang et al. 2023). STID leverages MLPs to capture spatio-temporal dynamics using spatial and temporal identity mappings. ST-MLP (Wang et al. 2023) focuses on using MLPs exclusively, relying on predefined spatial graph structures to improve efficiency. While Transformer-based approaches, such as STAEformer (Liu et al. 2023), show promising performance, their high computational cost and resource requirements hinder their efficiency and scalability.

Knowledge Distillation for Graphs. Knowledge distillation on graphs provides a promising approach for transferring knowledge from complex teacher GNNs to smaller student models, effectively reducing computational costs while preserving accuracy (Guo et al. 2022; Wu, Lin et al. 2022; Feng et al. 2022; Qin et al. 2021). This technique has been applied to various graph applications, including node/graph classification (Zhang et al. 2022; He et al. 2022), social media analysis (Qian et al. 2021), and recommender systems (Tao et al. 2022). Adversarial training, involving a discriminator and generator, has been employed to enhance knowledge distillation (He et al. 2022). Qian et al. (2021) applied knowledge distillation to a heterogeneous graph for analyzing drug trafficking from social media data. Tao et al. (2022) proposed a distillation-enhanced relational encoder to improve recommendation accuracy by capturing user-item interactions and social connections. **STKD** (Wang et al. 2024) is a method that adopts distillation via 1D CNN and LSTM to detect traffic anomalies. However, 1D CNNs are limited by fixed-size windows, hampering their ability to capture long-range dependencies crucial for modeling complex temporal patterns in traffic data. Additionally, both 1D CNNs and LSTMs produce fixed-length representations, potentially overlooking fine-grained temporal details necessary for capturing subtle variations and nuanced temporal correlations in dynamic traffic scenarios. Inspired by this, we effectively transfer complex spatial and temporal patterns from GNNs to a lightweight student model that is built upon MLP, addressing the challenge of achieving efficient and accurate traffic predictions.

Limitations

A potential lack of detailed discussion on the generalizability of our method LightST method across diverse traffic datasets. Additionally, our paper may not thoroughly address the robustness of LightST in real-world scenarios with

incomplete data. Furthermore, the paper might not extensively cover potential challenges in deploying LightST in large-scale practical applications, considering factors such as computational resources, and adaptability to dynamic traffic conditions.

Conclusion

We propose LightST, a scalable and high-performance traffic flow prediction framework that offers both model efficiency and generalization capability, which are often lacking in existing solutions. We draw inspiration from the knowledge distillation paradigm to achieve efficiency and retain the awareness of high-order spatial-temporal traffic dependencies across locations and time. To this end, we perform both explicit prediction-level and implicit embedding-level distillation to transfer spatio-temporal knowledge from a cumbersome GNN teacher to a simple yet effective MLP student. Additionally, our model adopts a new adaptive model alignment schema to further enhance the student model by alleviating over-smoothing effects. In future work, we plan to enhance our spatio-temporal knowledge distillation with causal inference, to identify confounding variables that may affect the spatio-temporal data and adjust for their effects in the knowledge distillation process.

References

- Bai, L.; Yao, L.; et al. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *International Conference on Neural Information Processing Systems (NeurIPS)*.
- Chen, D.; Lin, Y.; Li, W.; Li, P.; Zhou, J.; and Sun, X. 2020a. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 3438–3445.
- Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020b. Simple and deep graph convolutional networks. In *International Conference on Machine Learning (ICML)*, 1725–1735. PMLR.
- Chen, Y.; Segovia, I.; and Gel, Y. R. 2021. Z-GCNETs: time zigzags at graph convolutional networks for time series forecasting. In *International Conference on Machine Learning (ICML)*, 1684–1694. PMLR.
- Chen, Y.; Segovia-Dominguez, I.; et al. 2021. TAMP-S2GCNets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting. In *International Conference on Learning Representations (ICLR)*.
- Diao, Z.; Wang, X.; Zhang, D.; Liu, Y.; Xie, K.; and He, S. 2019. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 890–897.
- Fang, Z.; Long, Q.; Song, G.; and Xie, K. 2021. Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 364–373. ACM.
- Feng, K.; Li, C.; Yuan, Y.; and Wang, G. 2022. FreeKD: Free-direction Knowledge Distillation for Graph Neural Networks. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 357–366.
- Guo, Z.; Shiao, W.; Zhang, S.; Liu, Y.; Chawla, N.; Shah, N.; and Zhao, T. 2022. Linkless Link Prediction via Relational Distillation. In *International Conference on Machine Learning (ICML)*.
- Han, X.; and Gong, S. 2022. LST-GCN: Long Short-Term Memory Embedded Graph Convolution Network for Traffic Flow Forecasting. *Electronics*, 11(14): 2230.
- He, H.; Wang, J.; Zhang, Z.; and Wu, F. 2022. Compressing deep graph neural networks via adversarial knowledge distillation. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 534–544.
- Hu, Z.; and Hong, L. J. 2013. Kullback-Leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 1(2): 9.
- Izadi, M.; Safayani, M.; and Mirzaei, A. 2024. Knowledge Distillation on Spatial-Temporal Graph Convolutional Network for Traffic Prediction. *arXiv preprint arXiv:2401.11798*.
- Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; and Li, P. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International Conference on Machine Learning (ICML)*, 11906–11917. PMLR.
- Li, M.; and Zhu, Z. 2021. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 4189–4196.
- Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR)*. OpenReview.net.
- Liang, Y.; Ouyang, K.; Jing, L.; Ruan, S.; Liu, Y.; Zhang, J.; Rosenblum, D. S.; and Zheng, Y. 2019. Urbanfm: Inferring fine-grained urban flows. In *International Conference on Knowledge Discovery & Data Mining (KDD)*, 3132–3142.
- Liu, H.; Dong, Z.; Jiang, R.; Deng, J.; Deng, J.; Chen, Q.; and Song, X. 2023. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, 4125–4129.
- Lv, Z.; Xu, J.; Zheng, K.; Yin, H.; Zhao, P.; and Zhou, X. 2018. Lc-rnn: A deep learning model for traffic speed prediction. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2018, 27.
- Pan, Z.; Liang, Y.; Wang, W.; Yu, Y.; Zheng, Y.; and Zhang, J. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *International Conference on Knowledge Discovery & Data Mining (KDD)*, 1720–1730.
- Qian, Y.; Zhang, Y.; Ye, Y.; Zhang, C.; et al. 2021. Distilling meta knowledge on heterogeneous graph for illicit drug trafficker detection on social media. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 26911–26923.
- Qin, C.; Zhao, H.; Wang, L.; Wang, H.; Zhang, Y.; and Fu, Y. 2021. Slow learning and fast inference: Efficient graph similarity computation via knowledge distillation. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 14110–14121.
- Rao, X.; Wang, H.; Zhang, L.; Li, J.; Shang, S.; and Han, P. 2022. Fogs: First-order gradient supervision with learning-based graph for traffic flow forecasting. In *International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org.
- Shao, Z.; Zhang, Z.; Wang, F.; Wei, W.; and Xu, Y. 2022. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 4454–4458.
- Shleifer, S.; McCreery, C.; and Chitters, V. 2019. Incrementally improving graph WaveNet performance on traffic prediction. *arXiv preprint arXiv:1912.07390*.
- Song, C.; Lin, Y.; Guo, S.; and Wan, H. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 914–921.

Tao, Y.; Li, Y.; Zhang, S.; et al. 2022. Revisiting Graph based Social Recommendation: A Distillation Enhanced Social Graph Network. In *ACM Web Conference (WWW)*, 2830–2838.

Wang, X.; Ma, Y.; Wang, Y.; Jin, W.; Wang, X.; Tang, J.; et al. 2020. Traffic flow prediction via spatial temporal graph neural network. In *The Web Conference (WWW)*, 1082–1092.

Wang, X.; Wang, Z.; Wang, E.; and Sun, Z. 2024. Spatial-temporal knowledge distillation for lightweight network traffic anomaly detection. *Computers & Security*, 137: 103636.

Wang, Z.; Nie, Y.; Sun, P.; Nguyen, N. H.; Mulvey, J.; and Poor, H. V. 2023. St-mlp: A cascaded spatio-temporal linear framework with channel-independence strategy for traffic forecasting. *arXiv preprint arXiv:2308.07496*.

Wu, L.; Lin, H.; et al. 2022. Knowledge Distillation Improves Graph Structure Augmentation for Graph Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *International Conference on Knowledge Discovery & Data Mining (KDD)*, 753–763.

Yao, H.; Tang, X.; Wei, H.; Zheng, G.; and Li, Z. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, 5668–5675.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3634–3640. ijcai.org.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Zhang, S.; Liu, Y.; Sun, Y.; and Shah, N. 2022. Graph-less neural networks: Teaching old mlps new tricks via distillation. In *International Conference on Learning Representations (ICLR)*.

Zhou, K.; Huang, X.; Li, Y.; Zha, D.; Chen, R.; and Hu, X. 2020. Towards deeper graph neural networks with differentiable group normalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 4917–4928.

Zhu, J.; Wang, Q.; Tao, C.; Deng, H.; et al. 2021. AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. *IEEE Access*, 9: 35973–35983.

Reproducibility Checklist

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced [Yes]
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results [Yes]

- Provides well marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper [Yes]
- Does this paper make theoretical contributions? [Yes]
- All assumptions and restrictions are stated clearly and formally. [Yes]
- All novel claims are stated formally (e.g., in theorem statements). [Yes]
- Proofs of all novel claims are included. [Yes]
- Proof sketches or intuitions are given for complex and/or novel results. [Yes]
- Appropriate citations to theoretical tools used are given. [Yes]
- All theoretical claims are demonstrated empirically to hold. [Yes]
- All experimental code used to eliminate or disprove claims is included. ([Yes]
- Does this paper rely on one or more datasets? [Yes]

If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets [Yes]
- All novel datasets introduced in this paper are included in a data appendix. [NA]
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. [NA]
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. [Yes]
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. [Yes]
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisfying. [Yes]
- Does this paper include computational experiments? [Yes]

If yes, please complete the list below.

- Any code required for pre-processing data is included in the appendix. [Yes]
- All source code required for conducting and analyzing the experiments is included in a code appendix. [Yes]
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. [Yes]
- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from [Yes]
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. [Yes]

- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. [Yes]
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. [Yes]
- This paper states the number of algorithm runs used to compute each reported result. [Yes]
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. [NA]
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). [Yes]
- This paper lists all final hyperparameters used for each model/algorithm in the paper's experiments. [Yes]
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. [Yes]

Appendix

A.1 Algorithm of LightST

Algorithm 1: The LightST Learning Algorithm

Input: Historical observation tensor $\mathbf{X} \in \mathbb{R}^{N \times T}$ and the spatial graph \mathcal{G} , τ_2 , τ_3 , λ_1, λ_2 , learning rate η , maximum training epochs S

Output: trained parameters in Θ

- 1 Initialize all parameters in Θ ;
- 2 Train teacher model by Equation 5, and collect the traffic embeddings at each time interval t , denoted as $E_t^{(T)}$
- 3 **for** $epoch = 1, 2, \dots, S$ **do**
- 4 Calculate the loss $\mathcal{L}^{(S)}$ by Equation 5;
- 5 Perform prediction-level distillation and compute the KL loss \mathcal{L}^{KL} by Equation 6;
- 6 Perform embedding-level distillation from spatial and temporal dimensions by Equation 7;
- 7 Calculate the overall loss \mathcal{L} by Equation 8;
- 8 **for** *Optimizing parameters on weight factor in prediction-level* **do**
- 9 Calculating weight factor according to Eq 10
- 10 **end**
- 11 **for** $\theta \in \Theta$ **do**
- 12 $\theta = \theta - \eta \cdot \frac{\partial \mathcal{L}^{(S)}}{\partial \theta}$
- 13 **end**
- 14 **end**
- 15 **Return** all parameters Θ

A.2 Description of Baselines

We compare 18 baselines including many state-of-art Graph Neural Networks (GNNs)-based traffic flow prediction methods, where are divided into six groups:

- GCN-based traffic prediction methods. **STGCN** (Yu, Yin, and Zhu 2018): It uses gated temporal convolution module with graph neural network to capture spatial-temporal dependency. **DCRNN** (Li et al. 2018): It adopts diffusion convolutional RNN with fusion process to model spatial-temporal dependencies. **GWN** (Shleifer, McCreery, and Chitters 2019): It is a method which combines diffusion graph convolutions with 1D dilated convolutions to capture spatial and temporal dynamics, which benefits for improving traffic prediction performance.
- GAT-based Methods. **ASTGCN** (Zhu et al. 2021): It is an attention-based GCN model which further combines spatial-temporal attention with STGCN for capturing dynamic spatial and temporal information. **LSTGCN** (Han and Gong 2022): It combines graph attention network with graph convolution network to capture spatial dynamics. And it adopts temporal convolution network to capture temporal dynamics.
- Differential GNNs. **STG-ODE** (Fang et al. 2021): It adopts differential equation to overcome the shortcoming

Table 5: Data Description and Statistics.

Datasets	#Sensors	Time Period	Time Steps	Interval
PeMSD4	307	2018/1/1-2018/2/28	16,992	5 minutes
PeMSD8	170	2016/7/1-2016/8/31	17,856	5 minutes
PeMSD3	358	2018/9/1-2018/11/30	26,208	5 minutes
PeMSD7	883	2017/5/1-2017/8/31	28,224	5 minutes
PeMS-Bay	325	2017/1/1-2017/5/31	52,116	5 minutes

of depth of neural networks, which aims to solve the limitation (1) shallow GNNs cannot capture long-range spatial dynamics (2) temporal dynamics are ignored that plays a key role in traffic prediction, and only spatial dynamics are captured.

- GNNs enhanced with Zigzag persistence. **Z-GCNETs** (Chen, Segovia, and Gel 2021): It adopts zigzag persistence with temporal-aware graph convolution network for traffic flow prediction. **TAMP-S2GCNets** (Chen, Segovia-Dominguez et al. 2021): It deploys several persistence to capture temporal properties and then feed it into graph convolutional networks to capture dynamic spatial dependencies.
- Hybrid spatial-temporal GNNs. **FOGS** (Rao et al. 2022): It is a method which constructs the association graph via the spatial-temporal dynamics among nodes. **AGCRN** (Bai, Yao et al. 2020): It adopts learned node embeddings in graph convolutions to capture spatial-temporal correlations. **STSGCN** (Song et al. 2020): it captures spatial-temporal correlations via stacking multiple localized GCN layers with adjacent matrix on the time dimension. **STFGNN** (Li and Zhu 2021): It utilizes a spatial-temporal fusion graph neural network to capture spatial-temporal correlations.
- Distillation methods. **STKD** (Wang et al. 2024): STKD introduces a Spatial-Temporal Knowledge Distillation (STKD) framework for lightweight network traffic anomaly detection. This framework integrates multi-scale 1D CNNs and LSTMs, incorporating identity mapping to enhance performance. KD-pruning (Izadi, Safayani, and Mirzaei 2024): KD-pruning calculates pruning scores using a cost function and fine-tunes the student network, which incorporates a GNN architecture for enhanced performance.

A.3 Theoretical Analysis

The specific technical details of oversmoothing alleviation and caption of spatial-temporal dynamics, which is beneficial for predicting future traffic states are showed as follows:

Analysis of Adaptive Smoothing We present details of deduction related to adaptive smoothing. Firstly, we present the effect of high-order over-smoothing of GNN-based teacher on gradients, which generates Eq 9. The purpose of Eq 10 is to mitigate the issue of smoothing adaptively. Next, we show the details of derivations that obtain Eq 10 as fol-

lows:

$$\begin{aligned}
\frac{\partial \mathcal{L}^{(KL)}}{\partial \tilde{\mathbf{E}}_n^{(S)}} &= \sum_{n=1}^N \sum_{t'=T+1}^{T+H} \frac{1}{\text{softmax}(\tilde{\mathbf{E}}_{n,t'}^{(S)})} \frac{\partial \text{softmax}(\tilde{\mathbf{E}}_{n,t'}^{(S)})}{\partial \tilde{\mathbf{E}}_{n,t'}^{(S)}} \frac{\partial \tilde{\mathbf{E}}_{n,t'}^{(T)} \tilde{\mathbf{E}}_{n,t'}^{(S)}}{\partial \tilde{\mathbf{E}}_{n,t'}^{(S)}} \\
&= \sum_{n=1}^N \sum_{t'=T+1}^{T+H} \frac{1}{\text{softmax}(\tilde{\mathbf{E}}_{n,t'}^{(S)})} \frac{\partial \frac{e^{\tilde{\mathbf{E}}_{j \neq n,t'}^{(S)}}}{\sum_{o=1}^N e^{\tilde{\mathbf{E}}_{o,t'}^{(S)}}}}{\partial \tilde{\mathbf{E}}_{n,t'}^{(S)}} \frac{\partial \tilde{\mathbf{E}}_{n,t'}^{(T)} \tilde{\mathbf{E}}_{n,t'}^{(S)}}{\partial \tilde{\mathbf{E}}_{n,t'}^{(S)}} \\
&= \sum_{n=1}^N \sum_{t'=T+1}^{T+H} \omega \frac{\partial \tilde{\mathbf{E}}_{n,t'}^{(T)} \tilde{\mathbf{E}}_{n,t'}^{(S)}}{\tilde{\mathbf{E}}_{n,t'}^{(S)}} \\
\omega &= \frac{1}{\text{softmax}(\tilde{\mathbf{E}}_{n,t'}^{(S)})} \left(-\frac{e^{\tilde{\mathbf{E}}_{j \neq n,t'}^{(S)}}}{\sum_{o=1}^N e^{\tilde{\mathbf{E}}_{o,t'}^{(S)}}} \right) \left(-\frac{e^{\tilde{\mathbf{E}}_{n,t'}^{(S)}}}{\sum_{o=1}^N e^{\tilde{\mathbf{E}}_{o,t'}^{(S)}}} \right)
\end{aligned} \tag{11}$$

where we assume $n \neq j$. And the weight of $\frac{\partial \mathcal{L}^{(KL)}}{\partial \tilde{\mathbf{E}}_n^{(S)}}$ is learnable, compared to that of $\tilde{\mathbf{E}}_n^{(T)}$ in GNN in Eq 9 is unlearnable, which benefits for adaptive smoothing.

Detailed Complexity Analysis We provide detailed complexity analysis to answer the following two questions: (1) How GNN-based teacher model compared to MLP-based student model in terms of efficiency aspect? (2) What is the time complexity our KD and spatial-temporal operations? GNN-based methods must perform global graph information propagation process to obtain the spatial outputs. The process costs $\mathcal{O}(|\mathcal{E}| \times L \times T \times d)$ in our GNN-based teacher model during the training process. And it costs $\mathcal{O}(|B| \times T \times d)$ for each epoch during inference phase. In contrast, the structure of MLP-based student model is graph-less. It costs $\mathcal{O}(|B| \times L' \times T \times d)$ during training process and costs $\mathcal{O}(|B| \times T \times d)$. Thus, GNN-based model as encoders cause GNN-based methods lower efficiency in comparison with MLP-based graph-less structure methods.

A.4 Experiment Settings and Results

Evaluation Metrics and Evaluation Platform Following existing studies of traffic flow prediction (Li et al. 2018; Yu, Yin, and Zhu 2018; Shleifer, McCreery, and Chitters 2019; Zhu et al. 2021; Han and Gong 2022; Song et al. 2020; Bai, Yao et al. 2020; Li and Zhu 2021; Fang et al. 2021; Chen, Segovia, and Gel 2021; Chen, Segovia-Dominguez et al. 2021; Rao et al. 2022; Lan et al. 2022), we adopt three widely utilized metrics namely Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) as evaluation metrics for traffic prediction of four datasets, namely PeMSD4, PeMSD8, PeMSD3 and PeMSD7 in Table 1. Mean Average Distance (MAD) is used as the evaluation metric for detecting the smoothness of graph embeddings (Chen et al. 2020a).

All methods are implemented in Python 3.9 and PyTorch 1.12.0 with cuda 11.3. The experiments are conducted on

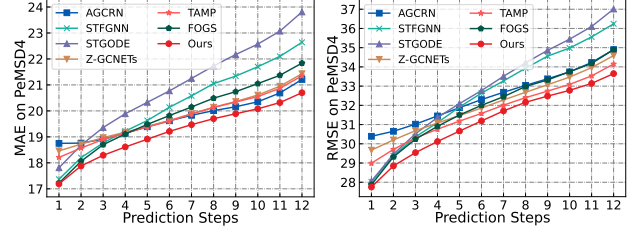


Figure 5: Multi-step prediction results on PeMSD4.

a server with 10-cores of Intel(R) Core(TM) i9-9820X CPU @ 3.30GHz 64.0GB RAM and 4 Nvidia GeForce RTX 3090 GPU.

Evaluation Protocol. To ensure a fair comparison, we follow exactly the same settings as (Lan et al. 2022; Rao et al. 2022; Fang et al. 2021; Li and Zhu 2021), including evaluation metrics and data split into training, validation and test sets. Specifically, we use the historical traffic flows from the previous 12 time steps (1 hour) to predict the traffic flows for the next 12 time steps (1 hour). In line with established practices, we adopt three widely used metrics, namely Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) to evaluate the accuracy of different traffic prediction models.

Multi-Step Prediction Results . In addition to the overall accuracy presented in Table 1, which is averaged over all time slots, we further evaluate the accuracy of our traffic forecasting model for individual time slots, as illustrated in Figure 5. This analysis provides a detailed understanding of the performance of our method for both short-term and long-range traffic forecasting, which is crucial for its practical applications in real-world traffic management systems. From the results, we can observe that our LightST consistently outperforms the compared baselines across all time slots, which further confirms the effectiveness of our new knowledge distillation paradigm for model compression.

Hyperparameter Study of LightST on PeMSD4 and PeMSD7 The objective of this section is to assess how important hyperparameters affect the functionality of our framework LightST. In Figure 7, we report our findings for the PeMSD3, PeMSD8, PeMSD4 and PeMSD7 datasets in terms of MAE, RMSE and MAPE. We highlight our findings as follows: Figure 7 Performance is depicted in (a), (b), (c), (d), (e), (f), (g) and (h) according to the number of MLP layers (from left to right: 1, 2, 3, and 4), as well as the batch size (from left to right: 2^3 , 2^4 , 2^5 , 2^6 , and 2^7) in (i), (j), (k) and (l). On PeMSD3, PeMSD8, PeMSD4 and PeMSD7 datasets, our framework LightST performs best when there are three layers and a batch size of 32 (64). Even when LightST performs at its worst, it still exceeds the majority of the baselines. These findings imply that the batch size and MLP depth have little effect on the performance of our LightST. 2) To regulate how severely our prediction-level and embedding-level restrict the joint model training, λ_1 , λ_2 are used as loss weights. Figure 7 (m), (n), (o), and (p) illustrate how λ_1 and λ_2 together influence the potency of the optimization of knowledge distillation. We discover that a higher weight of distillation results in performance mainte-

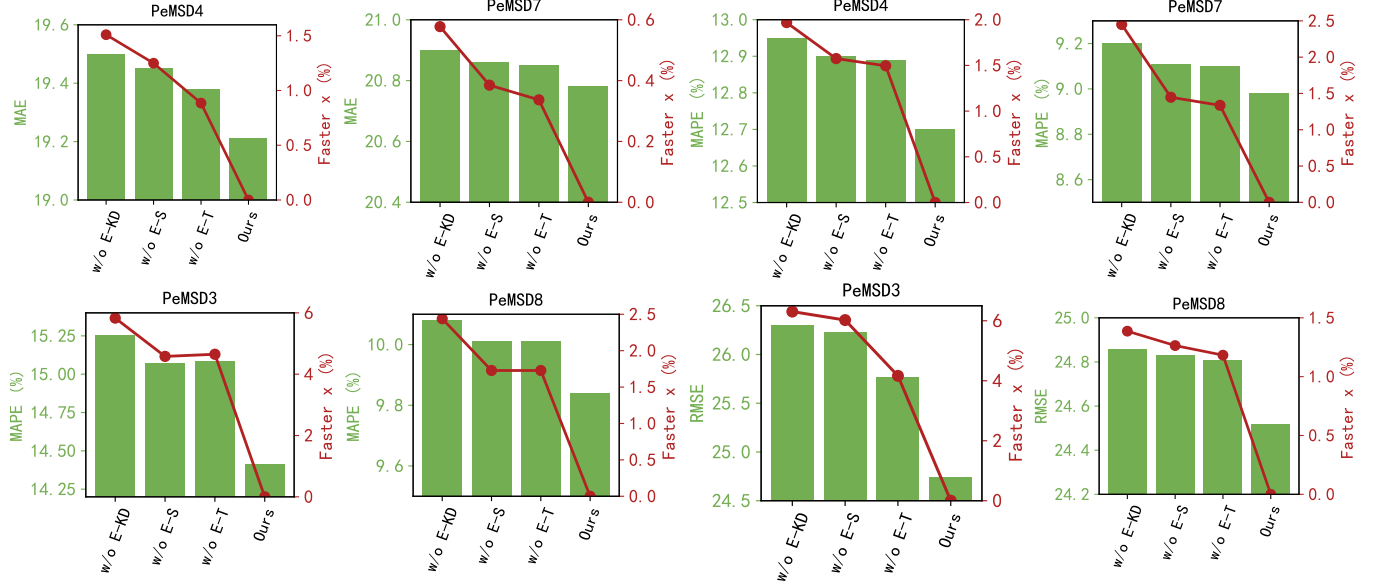


Figure 6: Ablation study of sub-modules in our spatio-temporal knowledge distillation paradigm. Our ablation study reveals the critical role of embedding-level knowledge distillation in transferring spatio-temporal signals, significantly improving prediction accuracy compared to models without this component. The study also demonstrates the effectiveness of adaptive embedding alignment across both spatial and temporal domains, capturing complex traffic dependencies. Notably, KL divergence is the most computationally demanding component of our model.

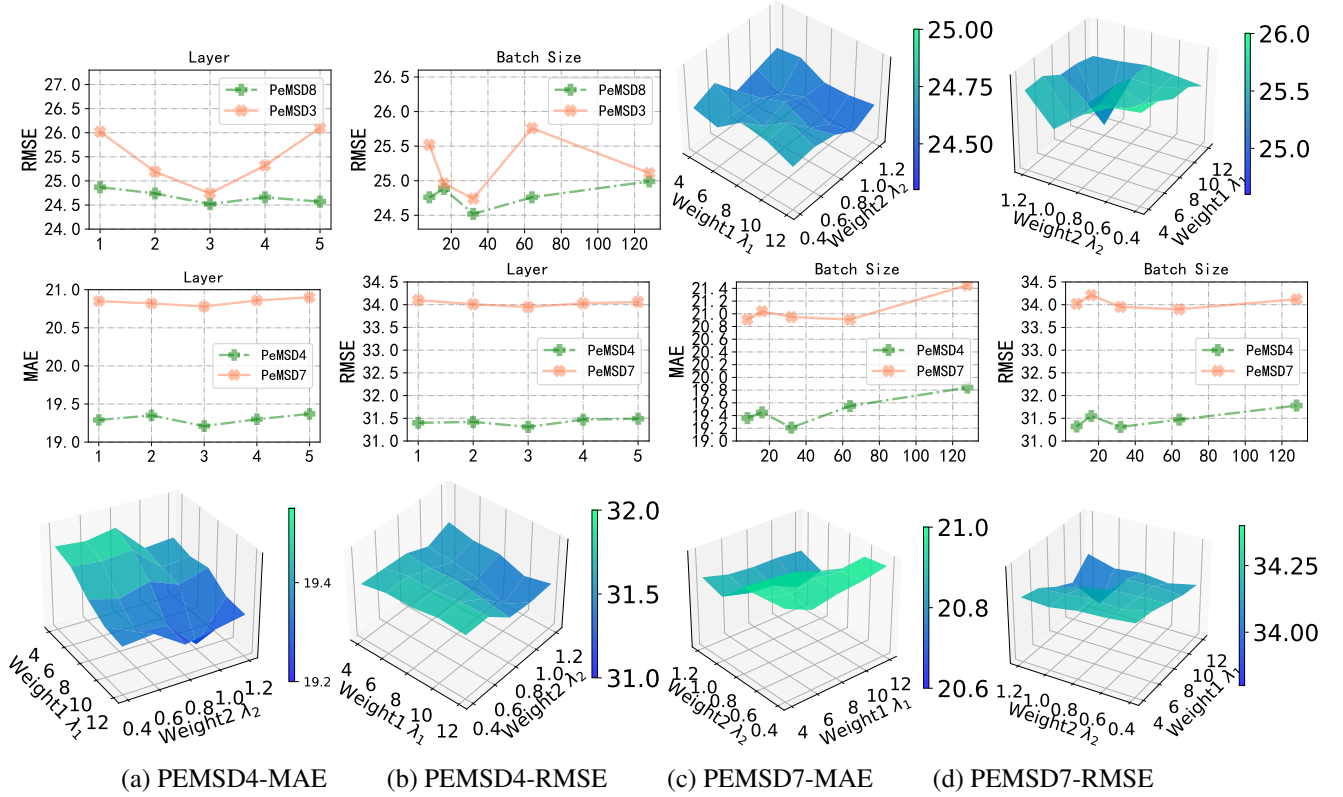


Figure 7: Hyperparameter study on PeMSD8 and PeMSD3 in terms of MAE and RMSE.

nance, allowing MLP to pick up enough information.