



A dynamic teacher-student gating framework for traffic prediction

Guangyu Wang^{a*}, Yujie Chen^{b*}, Ming Gao^{c*}, Jingmin An^{c*}, Zhiqiao Wu^{c*}, Jiafu Tang^{c*}
and Bowei Chen^{d*}

^aData Science and Artificial Intelligence, Dongbei University of Finance and Economics, Dalian, Liaoning, China; ^bSchool of Statistics, Dongbei University of Finance and Economics, Dalian, Liaoning, China; ^cSchool of Management Science and Engineering, Key Laboratory of Big Data Management Optimization and Decision of Liaoning Province, Dongbei University of Finance and Economics, Dalian, Liaoning, China; ^dAdam Smith Business School, University of Glasgow, Glasgow, UK

ABSTRACT

Accurate traffic speed prediction is essential for efficient urban mobility, yet increasingly complex spatial models often yield diminishing returns. We propose the Dynamic Teacher–Student Gating Framework (DTSGF), which combines diverse spatial modelling strategies through knowledge distillation without increasing student complexity. Its predict–evaluate–teach paradigm uses fine-grained teacher routing to select the most suitable teacher’s hidden state at each time step, generating soft labels for a lightweight student. The resulting three-layer Transformer, DTSGFormer, outperforms baseline models on two real-world datasets while using only 5% of their parameters.

ARTICLE HISTORY

Received 15 April 2025
Accepted 13 October 2026

KEYWORDS

Traffic prediction; knowledge distillation; spatiotemporal prediction

1. Introduction

Contemporary enterprise information systems (EIS) must convert heterogeneous operational streams into short-horizon forecasts that steer routing, scheduling, and inventory decisions at the pace of live operations (L. D. Xu 2011). Two gaps persist in the EIS and supply chain forecasting literature: spatially distributed decision workflows still rely on rigid predictive backbones unable to track temporally shifting regimes (Seyedan and Mafakheri 2020), and deep forecasters capable of capturing such regimes typically carry parameter budgets incompatible with the edge gateways and embedded controllers that form the last-mile of enterprise sensing (Murshed et al. 2021). Traffic speed prediction, whose sensor graph, diurnal variability, and latency budget jointly mirror these constraints, therefore serves as the proving ground of this study.

Effective traffic management is essential for urban functionality and sustainable development, reducing congestion, improving transportation efficiency, and supporting smart city initiatives (Lin et al. 2025; Y. Liu, Zhang, and Wang 2020; Sun et al. 2024). With advances in intelligent transportation systems, traffic speed prediction has become central to resource optimisation and improved travel efficiency through real-time data analysis. However, this prediction process presents unique challenges due to the complex spatiotemporal nature of traffic data (S. Liu et al. 2024; Z. Shao et al. 2024; Yan et al. 2025). For example, speed patterns recorded by the same sensor can vary significantly across different time steps. As illustrated in Figure 1(a), Sensor 1 in the PEMS04 dataset shows marked differences between weekdays (e.g. Monday and Tuesday) and weekends (e.g. Saturday), especially during morning peak hours.¹ Additionally, while sensors located at different points may show similar patterns during certain periods, they often differ notably in specific time frames. For instance, Sensors 215 and 217 in Figure 1(b) generally exhibit similar trends but diverge during morning and evening peak hours. As the number of sensors grows, spatiotemporal heterogeneity becomes increasingly pronounced (Gao et al. 2024), which diminishes the effectiveness of traditional prediction methods, such as Support Vector Regression (Awad et al. 2015), Random Forest (Rigatti 2017), and Gradient Boosting Decision Trees (Ke et al. 2017), in handling traffic speed prediction tasks.

Within this proving ground, the field has converged on increasingly innovative spatial modelling approaches to contend with the underlying heterogeneity. Grid-based models, which utilise standard

CONTACT Ming Gao  gm@dufe.edu.cn  School of Management Science and Engineering, Key Laboratory of Big Data Management Optimization and Decision of Liaoning Province, Dongbei University of Finance and Economics, Dalian, Liaoning; Bowei Chen  bowei.chen@glasgow.ac.uk  Adam Smith Business School, University of Glasgow, Glasgow, UK

*All authors contributed equally.

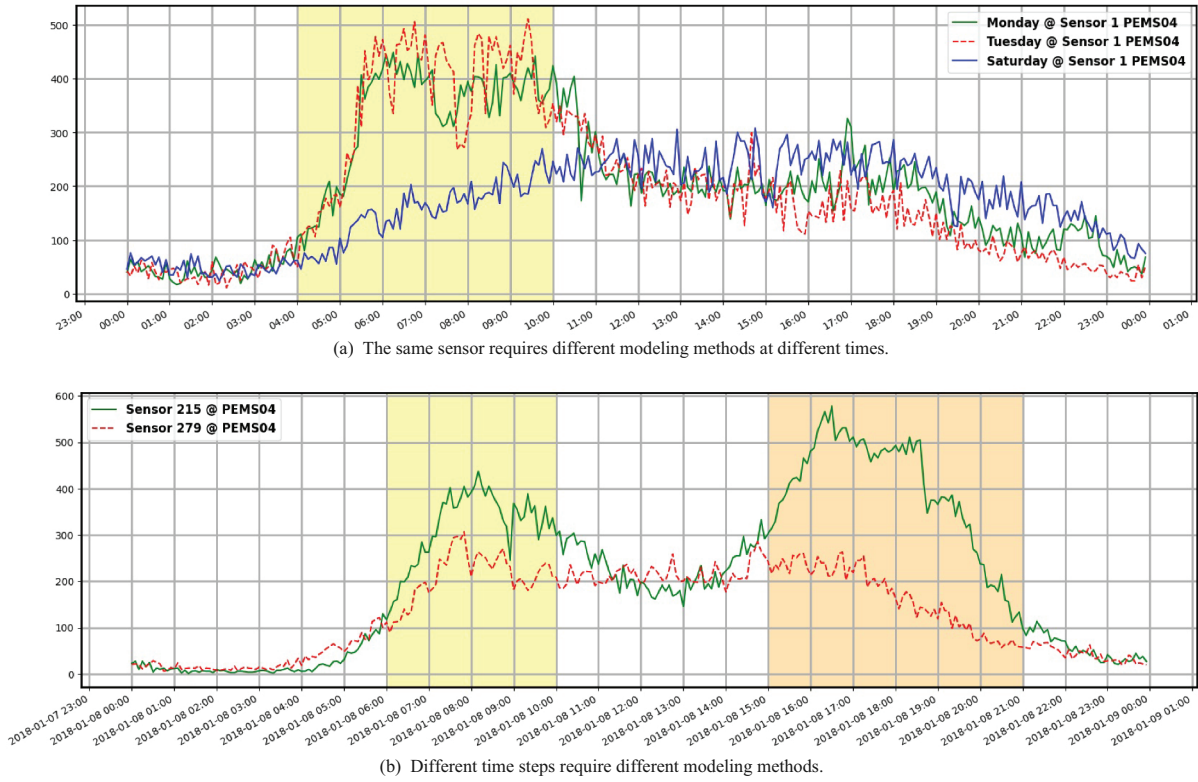


Figure 1. Time series plots of: (a) traffic speed patterns for sensor 1 from the PEMS04 dataset, highlighting speed variations between weekdays and weekends; (b) speed patterns for sensors 215 and 217, showcasing similarities and differences across various time intervals.

convolutional operations, have demonstrated strong performance in analysing standardised traffic speed data (Y. Shao et al. 2021; Yang et al. 2019; W. Zhang et al. 2019). However, real-world traffic data often relies on non-uniform grids, exposing the limitations of traditional convolutional methods in capturing the complex and dynamic structures of urban traffic (Chu et al. 2024). Graph Convolutional Networks (GCNs) address these challenges by modelling road segments as graph nodes, significantly enhancing traffic speed forecasting accuracy. Several studies have advanced GCN-based traffic speed modelling, including STGCN, ASTGCN (Guo et al. 2019), GWNNet (Wu et al. 2019), and MegaCRN (R. Jiang et al. 2023). While these methods improve performance, they often rely on increasingly complex structures that deliver diminishing returns. Frameworks that integrate multiple spatial modelling techniques have been explored (Lee and Ko 2024; Luo et al. 2024). However, these approaches generally deliver only incremental improvements over single-method models.

To move past these saturating returns of ever deeper spatial architectures, we draw on knowledge distillation, recently reframed as knowledge fusion, whose robust transfer capabilities can consolidate diverse spatial models into a single compact predictor. Specifically, we design multiple models, each employing distinct spatial modelling techniques, and integrate their knowledge into a compact and simplified student model through a multi-teacher knowledge distillation strategy. By directly supervising the student model with ground truth values, the risk of catastrophic errors from incorrect teacher selection is effectively mitigated. This approach enables the student model to fully leverage the strengths of various spatial modelling methods while avoiding the diminishing returns of excessive complexity. Unlike traditional ensemble, switching, or boosting-based multi-teacher distillation frameworks, we propose a novel predict-evaluate-teach paradigm specifically tailored for time series forecasting tasks. This paradigm incorporates a fine-grained routing mechanism that, at each time step, acts as a context-conditioned model selection policy, hard-selecting the single teacher whose latent representation most faithfully captures the prevailing spatiotemporal regime and enhancing performance by leveraging the best interpretation of spatiotemporal information at every instance. We term this dynamic teacher-student gated framework DTSFG and introduce

its student model, DTSGFormer, featuring a three-layer Transformer architecture optimised for predictive capabilities. Experimental results on two real-world datasets, METR-LA and PEMS-BAY, validate the effectiveness of this approach, setting a new benchmark in time series forecasting. Remarkably, DTSGFormer achieves superior performance while utilising just 5% of the parameters required by baseline models, demonstrating exceptional efficiency and scalability.

From a theoretical perspective, the predict-evaluate-teach paradigm advances the state-of-the-art in adaptive modelling by formalising dynamic teacher routing as an input-aware teacher selection process. Throughout the forecasting horizon, the gating module evaluates every candidate teacher against the observed spatiotemporal input and assigns prediction authority to the one whose hidden-state representation best matches the evolving traffic dynamics. Unlike mixture-of-experts (MoE) aggregation, in which a soft-weighted convex combination of all expert outputs risks diluting the sharpest predictive signal from the most contextually appropriate model, our hard-selection strategy preserves model-specific fidelity, while concurrent ground-truth supervision simultaneously safeguards against drift from erroneous routing decisions (Y. Liu, Zhang, and Wang 2020). This theoretical grounding is directly relevant to EIS, where heterogeneous sensor streams and distributed knowledge sources must be dynamically orchestrated to support real-time, context-aware enterprise decision-making (G. Jin et al. 2024; Lin et al. 2025; Sun et al. 2024).

Building on this grounding, our study makes significant methodological contributions to advancing time series forecasting. First, we introduce a novel multi-teacher knowledge distillation framework that departs from traditional static or ensemble-based strategies, leveraging a dynamic, fine-grained teacher selection mechanism to optimise knowledge transfer at each time step. At the core of this framework is the predict-evaluate-teach paradigm, which represents a new methodological approach to adaptively integrating diverse spatial modelling techniques while ensuring context-aware decision-making. Additionally, our framework redefines the relationship between model complexity and performance by demonstrating that parameter-efficient architectures can achieve superior results without relying on increasingly intricate designs. By supporting diverse architectural structures and promoting scalability, our framework establishes a methodological foundation that is both flexible and robust, as validated by its performance on real-world datasets. These methodological innovations offer a transformative perspective on achieving efficiency, adaptability, and accuracy in spatiotemporal modelling.

The rest of this paper is organised as follows. Section 2 reviews the related work and outlines the limitations of existing approaches. Section 3 formulates the problem and introduces the proposed DTSGF framework and its DTSGFormer model. Section 4 describes the experimental setup, datasets, and evaluation metrics, and presents the results. Section 5 discusses deployment feasibility, dataset scope, and broader EIS transferability. Finally, Section 7 concludes the paper and outlines future work.

2. Related work

This section covers two key streams of relevant literature. We begin with a review of traffic speed prediction methods, a task that presents substantial spatiotemporal heterogeneity and complex interactions among multiple variables. Several GCN-based methods are examined due to their strong capabilities in spatial modelling. C. Xu and Xu (2024) enhances the GCN network by incorporating Pearson correlation as prior knowledge, proving the huge role that prior knowledge plays in traffic speed prediction. An et al. (2021) reaffirms this by analysing the distribution of sensor data to obtain matrices for GCN's prior. However, dynamic modelling techniques have shown superior results compared to those based on prior knowledge methods. DMGAN uses multi-hop operations in each message-passing layer to dynamically model remote spatial dependencies (R. Li et al. 2023). Chu et al. (2024) approaches dynamic spatial modelling from a different angle, adaptively determining the number of GCN layers based on traffic complexity to achieve dynamic adjustment of the network structure. While these innovative modelling approaches improve performance, they inevitably require more complex structures and parameters. However, the marginal performance gains do not always justify the increased complexity, leading to a diminishing return. Luo et al. (2024) noticed this and attempted to model spatial features by integrating Adaptive GCN, Bi-LSTM, and TCN within a single framework. Applying multiple methods equally across all time steps can lead to a bloated model and decreased performance, despite avoiding diminishing returns from more complex structures. The purpose of the mixture-of-experts (MOE) design is to enhance the performance of traffic speed prediction

through the specialisation of expert functions. For instance, CP-MOE (W. Jiang et al. 2024a) has developed trend and periodicity experts, while TESTAM (Lee and Ko 2024) has created experts for capturing repetitive and non-repetitive patterns, addressing the shortcomings of traditional ensemble methods' static modelling. However, MOE's suboptimal routing for time series tasks limits its effectiveness, as it does not accurately adjust the choice of modelling methods, leading to only marginal improvements over single-method approaches.

Hinton, Vinyals, and Dean (2015) pioneered the concept of treating the soft logits produced by a complex model as knowledge, which could be distilled into a more compact model. The suboptimal routing of the MOE structure is avoided by using real labels as supervision for the student model. Building on this foundation, Romero et al. (2015) introduced feature distillation, extending knowledge distillation (KD) to include outputs from intermediate layers. Since then, various KD methods have been developed to extract knowledge from a single teacher across different applications, as seen in notable studies such as (Tsantekidis, Passalis, and Tefas 2021; J. Yu and Liu 2021; Gou et al. 2021; Cho and Hariharan 2019). However, teacher selection in these works often relies on subjective criteria. Given the superior performance of ensemble methods, several multi-teacher KD approaches have emerged. For instance, as illustrated in Figure 2 (Ensemble), averaging outputs from all teachers has proven effective (Allen-Zhu and Li 2020; Fukuda et al. 2017; Zhu et al. 2018). Additionally, Chebotar and Waters (2016) explored fixed weighting for different teachers, using grid search to optimise weight combinations.

Nevertheless, simply combining all teachers with the same or fixed weights is insufficient, as each teacher may perform very differently on specific samples. Therefore, Fukuda et al. (2017) proposed two strategies for multi-teacher KD, illustrated in Figure 2 (Augmented, Switched). One approach, called switched training, randomly selects a model from the teacher pool to guide the student's training. The other approach, called augmented training, updates the student's parameters using each teacher's output separately. Q. Xu et al. (2024) showed that switched training is generally less effective than augmented training. We attribute this to the random selection process in switched training, which often leads to suboptimal teacher choices. To address this issue, we propose a fine-grained routing mechanism that adaptively balances the load across multiple teachers, selecting the most suitable teacher model's hidden state at each time step to guide the student's learning process. We refer to this learning paradigm as prediction-evaluation-teaching. Specifically, all teacher models first produce predictions for the input data. Next, the routing mechanism evaluates these teachers based on their hidden states, and finally, the best teacher is chosen to train the student model.

Positioning relative to recent EIS and adjacent literature. Three recent research strands, spanning Enterprise Information Systems and closely adjacent venues, inform the present work. In transportation analytics, Chiang and Liu (2025) contributes from the EIS community, while Sun et al. (2024) and Lin et al. (2025) contribute dynamic graph and multi-view transformer models from the Information Fusion venue, whereas we target continuous speed forecasting that feeds upstream routing and capacity-allocation decisions. In adaptive planning, the EIS outlet contributes Yin et al. (2024) on cyber-physical orchestration of construction supply chains and Desgourdes and Ram (2024) on big-data-driven project decision-making, alongside the adjacent operations research work of Chen and Sun (2024) on kernel-based multi-retailer replenishment, whereas DTSGF advances this line at the level of model selection through an observation-conditioned teacher-routing mechanism. In information fusion, the EIS outlet contributes Gupta et al. (2025) on big-data strategy formation and Pang et al. (2025) on scenario-oriented interoperability, alongside Yan

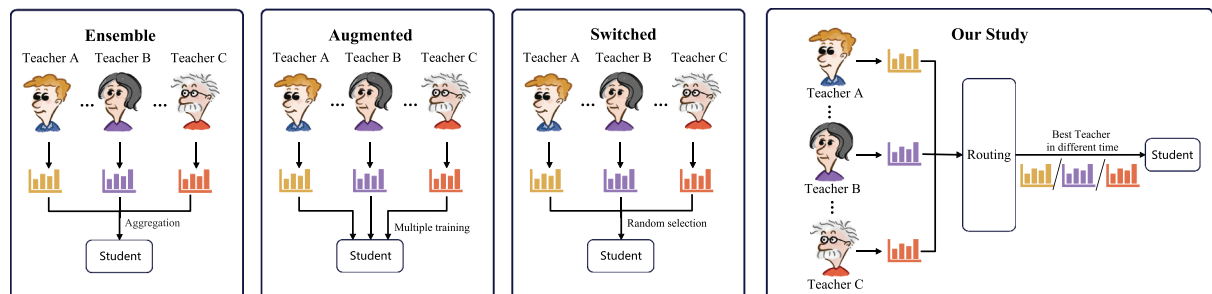


Figure 2. Schematic summary of multi-teacher distillation methods.

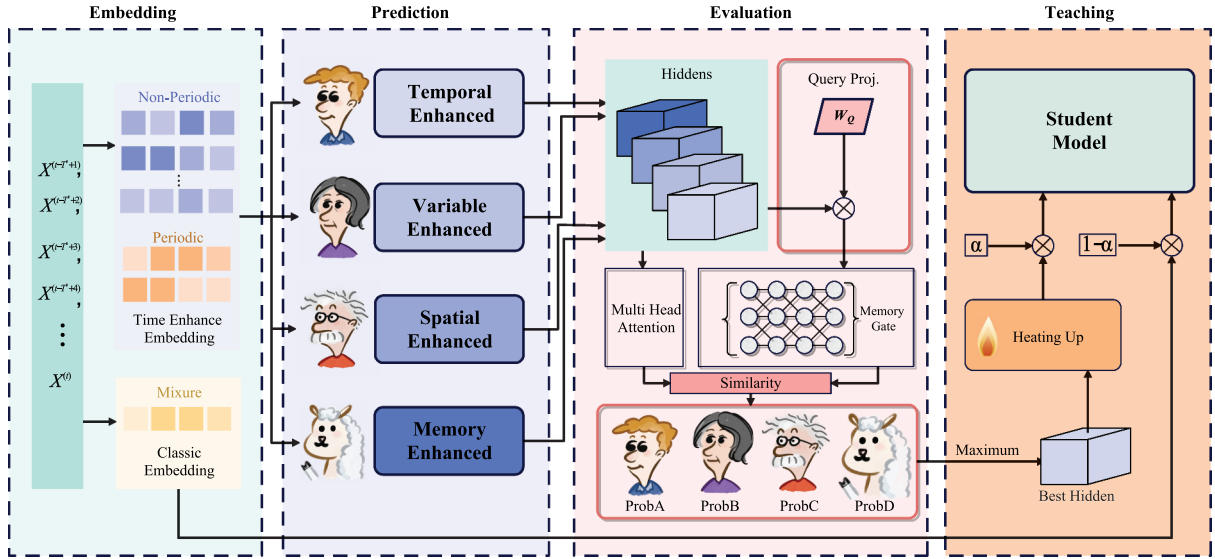


Figure 3. Schematic view of the proposed DTSGF, where the student model is DTSGFormer.

et al. (2025) from the Information Fusion venue on multimodal traffic fusion via retentive networks; in contrast, our predict-evaluate-teach paradigm selects a single teacher at each time step, preserving the predictive fidelity of the most suitable spatial model with a parameter footprint amenable to edge-level deployment.

3. Dynamic teacher-student gating framework

Traffic speed prediction is a spatio-temporal multivariate time series forecasting task. Given historical observations $\mathbf{X}_{in}^{(t)} = [\mathbf{X}^{(t-T'+1)}, \mathbf{X}^{(t-T'+2)}, \dots, \mathbf{X}^{(t)}] \in \mathbb{R}^{T' \times N \times F}$, where $\mathbf{X}^{(t)} \in \mathbb{R}^{N \times F}$ represents the feature matrix at time t , capturing F input attributes (such as speed and volume) for each of the N spatial nodes. The goal is to predict future traffic conditions for the next t time steps $\hat{\mathbf{X}}^{(t+1:t+T)} = [\hat{\mathbf{X}}^{(t+1)}, \hat{\mathbf{X}}^{(t+2)}, \dots, \hat{\mathbf{X}}^{(t+T)}] \in \mathbb{R}^{T \times N \times F'}$, where $\hat{\mathbf{X}}^{(t+\tau)} \in \mathbb{R}^{N \times F'}$ denotes the predicted features at future time $t + \tau$, and F' may be equal to F or differ, depending on the prediction task. This problem can be formulated using a spatio-temporal graph $G_t = (V, E_t, \mathbf{A}_t)$, where V is the set of N spatial nodes, E_t represents temporal edges at time t , $\mathbf{A}_t \in \mathbb{R}^{N \times N}$ is the adjacency matrix defining spatial relationships among nodes at time t . The objective is to learn a mapping function f parameterized by θ , that is $\hat{\mathbf{X}}^{(t+1:t+T)} = f(\mathbf{X}_{in}^{(t)}; \theta)$, where $f: \mathbb{R}^{T' \times N \times F} \rightarrow \mathbb{R}^{T \times N \times F'}$ maps historical observations to future predictions.

The overall architecture of the proposed DTSGF framework is presented in Figure 3. The input data, after an embedding layer, is processed by multiple teacher models $\{f_i\}_{i=1}^K$, each employing different spatial modeling strategies to generate predictions $\hat{\mathbf{X}}_i^{(t+1:t+T)}$. These predictions are assessed by a fine-grained routing mechanism within the evaluation module, which dynamically selects the most effective teacher model f_{i^*} at each time step. The hidden states from the selected teacher model serve as soft labels for training the student model $f_{student}$, while the true values provide hard labels for additional supervision. The student model, referred to as DTSGFormer, is implemented using a three-layer Transformer architecture.

3.1. General formulation of the predict-evaluate-teach paradigm

We first state the predict-evaluate-teach paradigm. Consider a forecasting task over N entities, such as sensors, products, warehouses, or production lines, with input space $\mathcal{X} = \mathbb{R}^{T' \times N \times F}$, output space $\mathcal{Y} = \mathbb{R}^{T \times N \times F'}$, and data distribution $\mathcal{D}$ over pairs $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$. The paradigm comprises a heterogeneous model pool, a routing function, and a knowledge-transfer loss.

(a) *Heterogeneous model pool.* The pool $\mathcal{F} = \{f_k\}_{k=1}^K$ consists of K predictors with shared input and output spaces but different structural priors. Each predictor factorizes as

$$f_k = g_k \circ h_k, \quad h_k : \mathcal{X} \rightarrow \mathcal{H} = \mathbb{R}^{\mathbf{T} \times \mathbf{N} \times \mathbf{C}}, \quad g_k : \mathcal{H} \rightarrow \mathcal{Y}, \quad (1)$$

where $\mathbf{H}_k = h_k(\mathbf{x}; \theta_k)$ is its hidden state. Each predictor is trained independently on a task loss ℓ :

$$\theta_k^* = \arg \min_{\theta_k} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} \ell(f_k(\mathbf{x}; \theta_k), \mathbf{y}), \quad k = 1, \dots, K. \quad (2)$$

No restriction is imposed on h_k , so attention-based, graph-based, recurrent, and memory-based predictors can coexist in the pool.

(b) *Routing function.* Let $\Omega = \{1, \dots, \mathbf{T}\} \times \{1, \dots, \mathbf{N}\}$ denote the set of routing units. The router $r_\phi : \mathcal{X} \times \mathcal{H}^K \rightarrow (\Delta^{K-1})^{|\Omega|}$ assigns each unit $\omega \in \Omega$ a distribution over the pool,

$$p_\omega^{(k)} = \frac{\exp(s(\mathbf{q}_\omega, \mathbf{z}_{k,\omega}))}{\sum_{j=1}^K \exp(s(\mathbf{q}_\omega, \mathbf{z}_{j,\omega}))}, \quad \mathbf{q} = q_\phi(\mathbf{x}), \quad \mathbf{z}_k = \psi(\mathbf{H}_k), \quad (3)$$

where q_ϕ maps the input to a context query, ψ summarizes a hidden state, s is a similarity function, and Δ^{K-1} is the probability simplex. Prediction authority is assigned by hard selection, which defines the routed knowledge target $\mathbf{H}^*$:

$$k_\omega^* = \arg \max_k p_\omega^{(k)}, \quad \mathbf{H}_\omega^* = \mathbf{H}_{k_\omega^*, \omega}, \quad \omega \in \Omega. \quad (4)$$

The router is trained with pseudo-labels $\mathbf{l}_\omega^{\text{best}}$ and $\mathbf{l}_\omega^{\text{worst}}$ obtained from quantiles of the per-unit errors $e_{k,\omega} = \ell(f_k(\mathbf{x})_\omega, \mathbf{y}_\omega)$, which reward low-error selections and penalize high-error selections:

$$\mathcal{L}_{\text{route}}(\phi) = -\frac{1}{|\Omega|K} \sum_{\omega \in \Omega} \sum_{k=1}^K (\mathbf{l}_{k,\omega}^{\text{best}} + \lambda \mathbf{l}_{k,\omega}^{\text{worst}}) \log p_\omega^{(k)}, \quad (5)$$

where λ balances the two terms.

(c) *Knowledge-transfer loss.* With the pool parameters $\{\theta_k^*\}_{k=1}^K$ fixed, a compact student $f_s = g_s \circ h_s$ is trained by combining distillation toward $\mathbf{H}^*$ with ground-truth supervision:

$$\mathcal{L}_s(\theta_s) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} \left[(1 - \alpha) \frac{1}{\rho^2} \ell\left(\frac{h_s(\mathbf{x}; \theta_s)}{\rho}, \frac{\mathbf{H}^*}{\rho}\right) + \alpha \ell(f_s(\mathbf{x}; \theta_s), \mathbf{y}) \right], \quad (6)$$

where $\rho > 0$ is the distillation temperature, denoted T in Section 3.4, and $\alpha \in [0, 1]$ weights the hard-label term, which bounds the effect of incorrect routing decisions. Predict, evaluate, and teach thus correspond to fitting $\mathcal{F}$, r_ϕ , and f_s , respectively. Only f_s is deployed, and inference reduces to $\hat{\mathbf{y}} = f_s(\mathbf{x}; \theta_s)$.

Since all elements are defined only through $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{H}$, and ℓ , the formulation is independent of the application domain. Section 3.2–3.4 instantiate it for traffic speed forecasting with $K = 4$ Transformer teachers, the memory query as q_ϕ , self-attention as ψ , cosine similarity as s , and MAE as ℓ . For demand forecasting, the entities are products or stores and the pool may combine hierarchical, promotion-aware, and graph-based predictors; for manufacturing, the entities are machines or process variables and the pool may combine physics-informed and data-driven predictors.

3.2. Teacher prediction

As illustrated in Figure 3, our proposed DTSGF framework integrates four distinct teacher models to evaluate its effectiveness. While these teacher models share a unified overall architecture, each utilises a unique spatial modelling strategy. In this section, we first describe the common components shared by all teacher models and then detail their individual spatial modelling approaches. All teacher models employ the classical Transformer architecture as their backbone network. We define the standard Multi-Head Self-Attention (MSA) mechanism and describe the operations following the embedding process. Given an input tensor after embedding $\mathbf{X}_{\text{emb}} \in \mathbb{R}^{\mathbf{T} \times \mathbf{N} \times \mathbf{C}}$, we first compute the queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$ by linear transformations of $\mathbf{X}_{\text{emb}}$ using learnable weight matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{\mathbf{C} \times \mathbf{C}}$:

$$\mathbf{Q} = \mathbf{X}_{\text{emb}} \mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}_{\text{emb}} \mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}_{\text{emb}} \mathbf{W}_V, \quad (7)$$

resulting in $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{T \times N \times C}$. For each spatial location $n \in \{1, \dots, N\}$, we extract the sequences $\mathbf{Q}_n = \mathbf{Q}_{:,n,:}$, $\mathbf{K}_n = \mathbf{K}_{:,n,:}$, and $\mathbf{V}_n = \mathbf{V}_{:,n,:}$ where $\mathbf{Q}_n, \mathbf{K}_n, \mathbf{V}_n \in \mathbb{R}^{T \times C}$. We compute the attention scores for each spatial location n by applying the scaled dot-product attention over the time dimension:

$$\mathbf{A}_n = \text{softmax}\left(\frac{\mathbf{Q}_n \mathbf{K}_n^T}{\sqrt{d}}\right), \quad (8)$$

where $\mathbf{A}_n \in \mathbb{R}^{T \times T}$. The output for each spatial location n is then computed as:

$$\mathbf{X}_{\text{out},n} = \mathbf{A}_n \mathbf{V}_n, \quad (9)$$

resulting in $\mathbf{X}_{\text{out},n} \in \mathbb{R}^{T \times C}$. For MSA with h heads, we perform the above computations for each head $i \in \{1, \dots, h\}$, obtaining outputs $\mathbf{X}_{\text{out},n}^{(i)} \in \mathbb{R}^{T \times d_i}$ where $d_i = d/h$. The outputs from all heads are concatenated along the feature dimension, forming $\text{Concat}(\mathbf{X}_{\text{out},n}^{(1)}, \dots, \mathbf{X}_{\text{out},n}^{(h)}) \in \mathbb{R}^{T \times C}$, which is then projected back to the embedding dimension C using a learnable weight matrix $\mathbf{W}_O \in \mathbb{R}^{C \times C}$:

$$\mathbf{X}_{\text{out},n}^{\text{all}} = \text{Concat}(\mathbf{X}_{\text{out},n}^{(1)}, \dots, \mathbf{X}_{\text{out},n}^{(h)}) \mathbf{W}_O, \quad (10)$$

resulting in $\mathbf{X}_{\text{out},n}^{\text{all}} \in \mathbb{R}^{T \times C}$. Reconstructing the outputs for all spatial locations, we obtain the final output $\mathbf{X}_{\text{out}} \in \mathbb{R}^{T \times N \times C}$.

3.2.1. Shared structure of teacher models

The input for traffic speed prediction is $X_{\text{in}} \in \mathbb{R}^{T \times N \times F}$, where F represents the feature dimension (e.g. traffic speed, flow, etc.). Before feeding the data into the MSA module, we apply an embedding transformation to project the input features into a higher-dimensional space, enhancing the model's capacity to effectively transfer knowledge from the historical T' time steps (source domain) to the upcoming T time steps (target domain). Temporal features, such as the time of day, exhibit specific periodic patterns and serve as global positional indicators. Therefore, we encode temporal position with a Temporal Enhanced Embedding (TEM), as proposed by Kazemi et al. (2019), which has been demonstrated to improve the quality of temporal representations. As illustrated in Figure 4, the TEM framework integrates both periodic and non-periodic temporal characteristics to generate a comprehensive vector representation of time. Specifically, for a temporal feature $\tau \in \mathbb{N}$, we represent τ with an h -dimensional embedding vector $v(\tau)$, using learnable parameters w_i and ϕ_i for each embedding dimension i :

$$v(\tau)[i] = \begin{cases} w_i \tau + \phi_i, & \text{if } i = 0, \\ \mathcal{F}(w_i \tau + \phi_i), & \text{if } 1 \leq i \leq h-1, \end{cases} \quad (11)$$

where $\mathcal{F}$ is a periodic activation function, such as sine or cosine. After embedding, the input data is combined with the temporal embedding:

$$X_{\text{emb}} = \text{concat}(X_{\text{in}}, X_{\text{TEM}}), \quad (12)$$

where $X_{\text{TEM}} \in \mathbb{R}^{T \times N \times C}$ is the temporal embedding applied to the input. This combined input X_{emb} is then fed into the MSA mechanism. The embedding dimension C represents the transformed features derived from the original input features F :

$$X_{\text{in}} \in \mathbb{R}^{T \times N \times F} \rightarrow X_{\text{emb}} \in \mathbb{R}^{T \times N \times C}. \quad (13)$$

$$X_{\text{out}} = \text{MSA}(X_{\text{emb}}). \quad (14)$$

As shown in Figure 5, all teacher models undergo the MSA module, followed by their distinct spatial modelling layers, and then pass through a feed-forward network to obtain the final output. The output X_{out} is projected into a lower-dimensional space to produce the future predictions. When a teacher model is selected to train the student model, X_{out} , now serving as the hidden state, is used for further training of the student model.

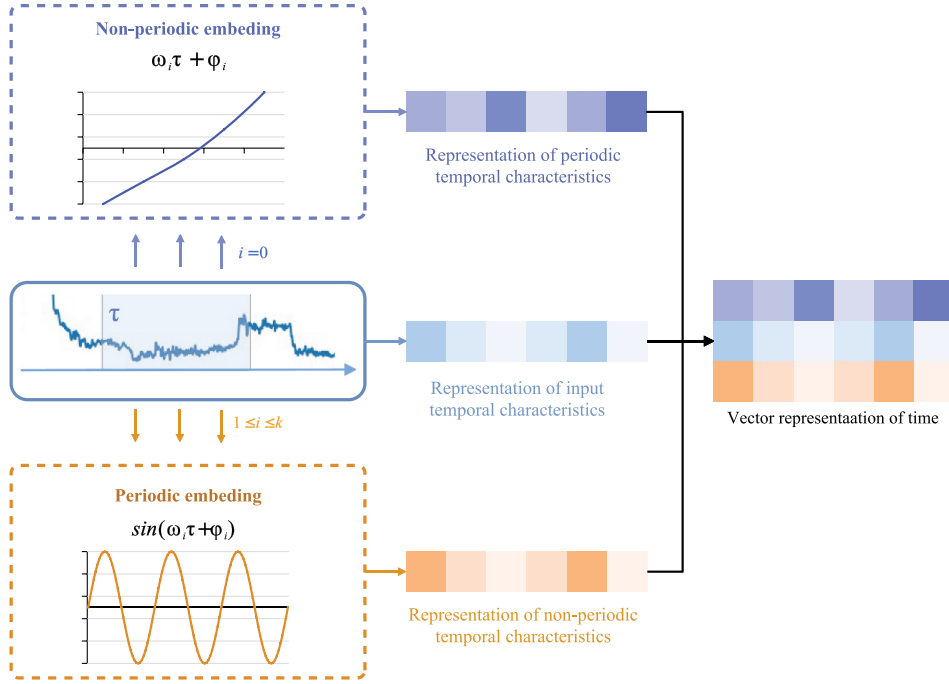


Figure 4. Temporal enhanced embedding (TEM) framework for modeling the periodicity and linearity of time series data.

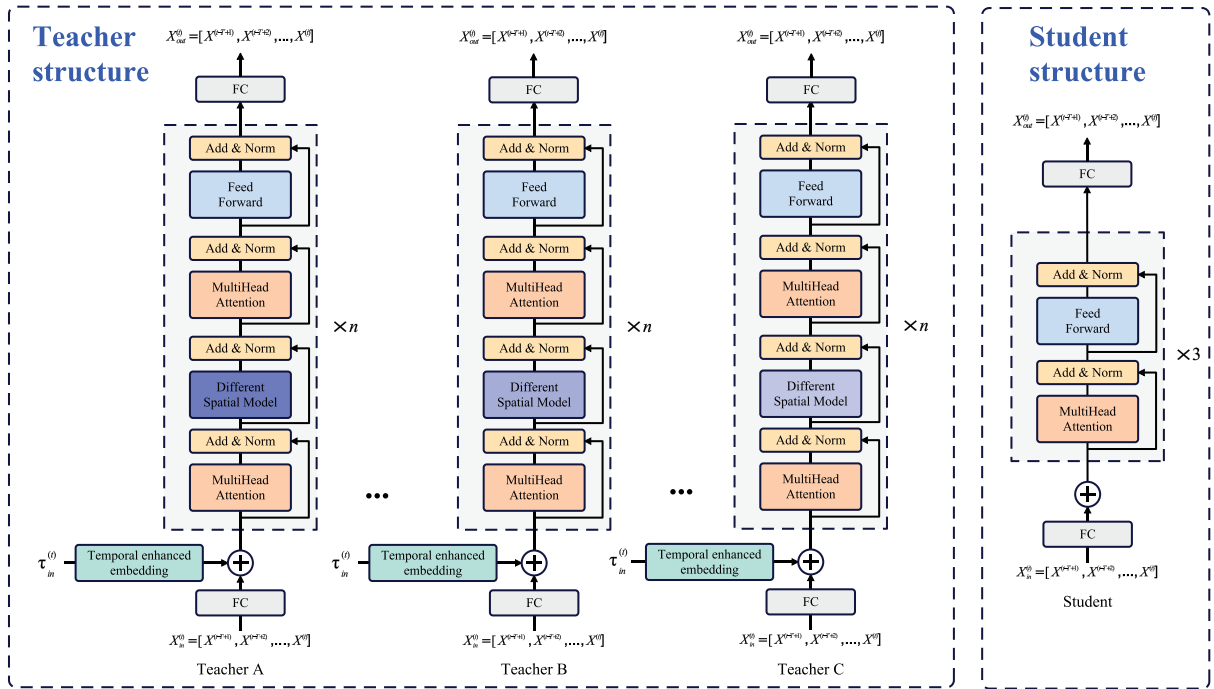


Figure 5. Structures of the teacher and student models. The teacher models (A, B, C) share a unified architecture but differ in their spatial modelling approaches. Each teacher employs temporal-enhanced embeddings, multi-head attention, and feed-forward layers. The student model features a streamlined structure designed to integrate the distilled knowledge from the teachers during training.

3.2.2. Distinct spatial modelling strategies of teacher models

To overcome the limitations of using a single spatial modelling approach and reduce inductive bias, we designed four distinct spatial modelling strategies, each applied to a different teacher model. While Figures 4–5 depict only three models for aesthetic reasons, Figure 6 illustrates all four models. In this figure,

the red lines highlight the spatial features emphasised by each modelling strategy, with H_l and H_{l-1} representing two adjacent neural network layers, and $\{t_1, \dots, t_i, \dots, t_p\}$ denoting the temporal dimension. All spatial modeling strategies are implemented by modifying MSA or combined with GCN. Next, we will provide a detailed explanation of how to achieve these four modeling strategies.

- (a) *Teacher with Temporal Enhanced* as shown in Figure 6(a), the teacher with temporal enhancement employs a classical temporal attention mechanism to predict future values based on historical time information. The objective of temporal attention aligns with the shared module in the teacher model, aiming to facilitate the transition of the model's domain from the historical T' time steps to the subsequent T time steps. Specifically, the model computes attention weights which are embedded via TEM over the temporal dimension of $X_{emb} \in \mathbb{R}^{T' \times N \times C}$, allowing information from different time points to be weighted and aggregated, resulting in the output $X_{out}^{tem} = \text{MSA}(X_{emb})$.
- (b) *Teacher with Variable Enhanced* aims to implement variable-centric modelling. The importance was first pointed out by Y. Zhang and Yan (2023) and further deepened by Y. Liu et al. (2024) by simply modifying the MSA mechanism to ultimately achieve the feature capture depicted in Figure 6(b). In this approach, each variable in layer H_l is connected to every variable in layer H_{l-1} , enabling comprehensive inter-variable interactions across layers. To accomplish this, time points from each time series are embedded into variable tokens, allowing the attention mechanism to exploit these tokens for capturing intricate correlations among multiple variables. Simultaneously, a feed-forward network is applied to each variable token to learn its nonlinear representations, enhancing the model's capacity to capture complex patterns.

This teacher replaces TEM with a variable-centric embedding (VEM), which embeds latent information across time steps into the nodes. This approach facilitates the modelling of variable-specific temporal dynamics. To ensure that the teacher with variable enhancement can be governed by the same routing mechanism as the other teachers, we adopt a projection method as proposed by Wang et al. (2024). Given the TEM-embedded input tensor $\mathbf{X}_{emb} \in \mathbb{R}^{T' \times N \times C}$, we first apply a linear transformation to adjust the feature dimensions and capture essential variable characteristics:

$$\mathbf{X}'_{emb} = \mathbf{X}_{emb} \mathbf{W}_\phi + \mathbf{b}_\phi, \quad \mathbf{W}_\phi \in \mathbb{R}^{C \times F}, \quad \mathbf{b}_\phi \in \mathbb{R}^F, \quad (15)$$

resulting in an intermediate tensor $\mathbf{X}'_{emb} \in \mathbb{R}^{T' \times N \times F}$. To facilitate variable-centric embedding, we permute the tensor dimensions to transpose the temporal and spatial axes:

$$\mathbf{X}_{emb} = \mathcal{P}(\mathbf{X}'_{emb}), \quad \mathbf{X}_{emb} \in \mathbb{R}^{N \times T' \times F}, \quad (16)$$

where $\mathcal{P}$ denotes the permutation operator such that $\mathbf{X}_{emb}(n, t, f) = \mathbf{X}'_{emb}(t, n, f)$. This rearrangement aligns the data for variable-centric processing. The VEM is then applied to each variable n to capture variable-specific temporal dependencies:

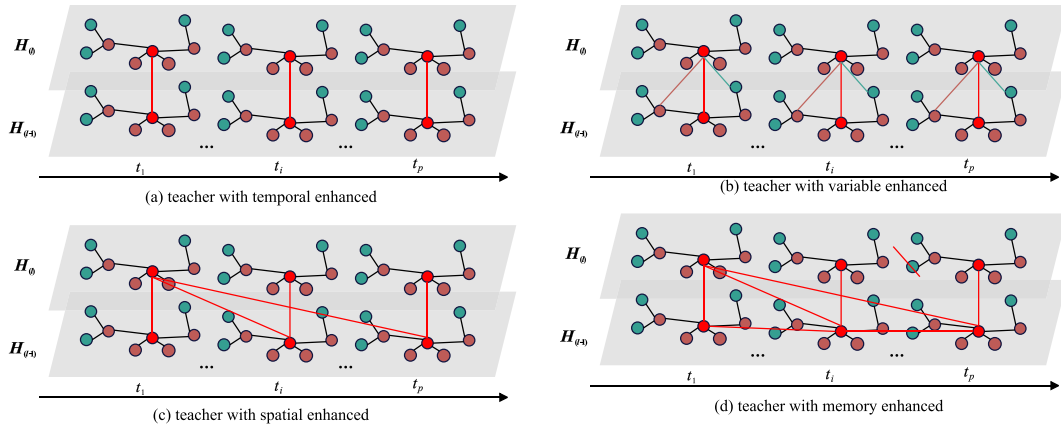


Figure 6. Different spatial modeling strategies used by the teacher models.

$$\hat{\mathbf{X}}_{\text{emb}}(n, t) = \sigma(\mathbf{X}_{\text{emb}}(n, t)\mathbf{W}_\psi + \mathbf{b}_\psi), \quad \mathbf{W}_\psi \in \mathbb{R}^{F \times C'}, \quad \mathbf{b}_\psi \in \mathbb{R}^{C'}, \quad (17)$$

where $\sigma(\cdot)$ is an activation function, and $\hat{\mathbf{X}}_{\text{emb}} \in \mathbb{R}^{N \times T' \times C'}$. We set $C' = C$ to maintain structural consistency across the teacher models. This embedding effectively captures the temporal patterns inherent to each variable. To integrate the variable-centric representations back into the original framework, we perform an inverse permutation to restore the original tensor layout:

$$\hat{\mathbf{X}}'_{\text{emb}} = \mathcal{P}^{-1}(\hat{\mathbf{X}}_{\text{emb}}), \quad \hat{\mathbf{X}}'_{\text{emb}} \in \mathbb{R}^{T' \times N \times C'}, \quad (18)$$

where $\hat{\mathbf{X}}'_{\text{emb}}(t, n, c') = \hat{\mathbf{X}}_{\text{emb}}(n, t, c')$. The MSA mechanism is then directly applied to the embedded input to capture long-range dependencies:

$$\mathbf{X}_{\text{out}}^{\text{var}} = \text{MSA}(\hat{\mathbf{X}}'_{\text{emb}}), \quad (19)$$

where $\mathbf{X}_{\text{out}}^{\text{var}} \in \mathbb{R}^{T' \times N \times C'}$. This operation allows the model to attend to variable-specific information across different time steps, enhancing its ability to model complex temporal dynamics. Finally, to ensure compatibility with the other teacher models and facilitate seamless integration within the overall framework, we project the output back to the original feature space using a linear transformation:

$$\mathbf{X}_{\text{out}}^{\text{var}} = \mathbf{X}_{\text{out}}^{\text{var}}\mathbf{W}_{\phi'} + \mathbf{b}_{\phi'}, \quad \mathbf{W}_{\phi'} \in \mathbb{R}^{C' \times C}, \quad \mathbf{b}_{\phi'} \in \mathbb{R}^C. \quad (20)$$

(c) Teacher with Spatial Enhanced aims to improve temporal prediction by leveraging the similarity between nodes. Unlike the Teacher with Temporal Enhancement, this teacher model first applies a MSA layer across the N nodes, followed by another MSA layer over the T' time steps to capture both spatial and temporal dependencies. This approach computes spatial attention from node similarity, enabling the model to consider interactions between any pair of nodes, including nodes without a direct physical connection. This setup facilitates similarity-based attention, leading to better generalization. The corresponding computations are as follows: Given the embedded input tensor $\mathbf{X}_{\text{emb}} \in \mathbb{R}^{T' \times N \times C}$, we first permute the dimensions to transpose the time and node axes:

$$\mathbf{X}_{\text{emb}}^{\text{perm}} = \text{Transpose}(\mathbf{X}_{\text{emb}}, (1, 0, 2)) \in \mathbb{R}^{N \times T' \times C}, \quad (21)$$

We then apply MSA over the spatial dimension (nodes):

$$\hat{\mathbf{X}}_{\text{out}}^{\text{spa}} = \text{MSA}_{\text{nodes}}(\mathbf{X}_{\text{emb}}^{\text{perm}}) \in \mathbb{R}^{N \times T' \times C}, \quad (22)$$

Next, we transpose the tensor back to its original arrangement:

$$\hat{\mathbf{X}}_{\text{out}}^{\text{perm}} = \text{Transpose}(\hat{\mathbf{X}}_{\text{out}}^{\text{spa}}, (1, 0, 2)) \in \mathbb{R}^{T' \times N \times C}. \quad (23)$$

Subsequently, we apply MSA over the temporal dimension (time steps):

$$\mathbf{X}_{\text{out}}^{\text{spa}} = \text{MSA}_{\text{time}}(\hat{\mathbf{X}}_{\text{out}}^{\text{perm}}) \in \mathbb{R}^{T' \times N \times C}. \quad (24)$$

Since spatial attention is computed based on similarity, feature interactions occur between any two similar nodes throughout the neural network layers, effectively capturing the relationships illustrated in Figure 6. By learning spatial attention from the inherent similarities between nodes, this method enhances the model's ability to generalise.

(d) Teacher with memory enhanced inspired by memory-augmented graph structure learning (Hong et al. 2024; Lee et al. 2022), the memory module M is a set of vectors with shape $\mathbb{R}^{N \times m}$, where M represents the memory size. A hypernetwork (Ha, Dai, and Le 2016) generates node embeddings conditioned on M . After generating new node embeddings, the Teacher with Memory Enhancement first performs spatial feature aggregation through a GCN and then passes the output to an MSA layer for final prediction. This teacher model can capture various input data patterns by matching feature aggregation patterns within the memory since the node embeddings are generated conditionally on M , which is optimized through the loss function. Theoretically, Teachers with Memory Enhancement can establish links between nodes and time steps, as shown in Figure 6. The process of generating node embeddings and performing GCN is as follows:

$$E = \mathbf{M}W_E; \quad \tilde{A} = \text{softmax}(\text{ReLU}(EE^T)); \quad X_{emb} = \tilde{A}X_T W_{GCN}, \quad (25)$$

where W_E and W_{GCN} are learnable weights. After the GCN operation is completed, the resulting node features H are input into the MSA layer, producing the prediction of the Teacher with Memory Enhancement: $X_{out}^{mem} = \text{MSA}(X_{emb})$

3.3. Teacher evaluation

Multiple teacher models with distinct spatial modelling strategies often exhibit varying hidden states and behaviours, leading to challenges such as conflicting modelling approaches, optimisation difficulties when handled simultaneously, increased computational complexity, and issues in workload allocation—particularly in determining optimal activation distribution over time steps. To address these challenges, we propose evaluating the hidden state of each teacher model at every time step to assign the optimal teacher to the student, optimising the process by selecting appropriate teacher models based on their hidden states. This dynamic teacher selection is akin to the expert selection in Mixture-of-Experts (MOE) models like ST-MOE (Zoph et al. 2022) and CP-MOE (W. Jiang et al. 2024b), which use teacher outputs to make confidence-based decisions or manually specify activation strategies.

However, such designs often result in more powerful teachers receiving stronger optimisation, while teachers intended for specific, less frequent scenarios may not get adequate training—contradicting our goal of designing diverse spatial modelling strategies. To mitigate this, we define the generation of the choice matrix C as a memory query, which is jointly supervised by the loss functions and the memory state. In Section 6, the Teacher with Memory Enhancement performs feature aggregation and pattern matching to model various input data patterns. Similarly, the memory query process matches features in memory to select the most appropriate teacher for different input patterns. For the output X_{out} , containing outputs of multiple teacher $X_{out}^{agg} = \{X_{out}^{tem}, X_{out}^{var}, X_{out}^{spa}, X_{out}^{mem}\}$, the subscripts represent outputs from the four teacher models: temporal, variant, spatial, and memory-enhanced.

The memory query process is redefined using the attention mechanism: We compute the memory-based representation O by attending to the memory M using the input features X_{in} as queries:

$$O = \text{softmax}\left(\frac{X_{in}M^T}{\sqrt{d}}\right)M. \quad (26)$$

For each teacher i , we compute a self-attention-based representation $\hat{O}^{(i)}$ using the teacher's output $X_{out}^{(i)}$:

$$\hat{O}^{(i)} = \text{softmax}\left(\frac{X_{out}^{(i)}(X_{out}^{(i)})^T}{\sqrt{d}}\right)X_{out}^{(i)}. \quad (27)$$

We then compute the teacher choice probability $p^{(i)}$ by measuring the similarity between O and $\hat{O}^{(i)}$:

$$s^{(i)} = \text{sim}(O, \hat{O}^{(i)}) = \frac{O \cdot \hat{O}^{(i)}}{\|O\| \|\hat{O}^{(i)}\|}, \quad p^{(i)} = \frac{\exp(s^{(i)})}{\sum_j \exp(s^{(j)})}. \quad (28)$$

The teacher with the highest $p^{(i)}$ is selected.

We optimise the selection of teachers using the *Best Routing Loss* and the *Avoid Worst Routing Loss*. At each time step t , we generate a choice matrix $\mathbf{C} = R \in \mathbb{R}^{T \times N}$ representing the selection results. By comparing each teacher's predictions with the ground truth, we identify the best and worst teachers—the one with the smallest error is the best expert, while the one with the largest error is the worst. Our optimisation objective is to make the selection as close as possible to the best choice while avoiding the worst. By selecting the optimal teacher for each node at each time step, only the selected teacher computes the loss during training, addressing the challenge of optimising multiple teachers simultaneously.

We define the Best Choice Loss as a cross-entropy loss with pseudo-labels. The prediction $\hat{y}$ from the optimal teacher is obtained based on the teacher selection probability $\mathbf{p}$. The Best Choice Loss is defined as:

$$L_{\text{Best}}(\mathbf{p}) = -\frac{1}{E} \sum_e l_{\text{Best}} \log(p_e), \quad (29)$$

where E is the total number of teachers.

In best routing selection, the routing is considered correct (optimal) if the loss $L(y, \hat{y})$ is smaller than the $1 - q$ -th quantile. The pseudo-label l_{Best} is defined as:

$$l_{\text{Best}} = \begin{cases} 1, & \text{if } L(y, \hat{y}) < \text{Quantile}_{1-q} \text{ and } e = \arg \max(\mathbf{p}), \\ \frac{1}{E-1}, & \text{if } L(y, \hat{y}) \geq \text{Quantile}_{1-q} \text{ and } e \neq \arg \max(\mathbf{p}), \\ 0, & \text{otherwise.} \end{cases} \quad (30)$$

Similarly, we define the Avoid Worst Routing Loss to prevent selecting the worst teacher. If the loss $L(y, \hat{y})$ is greater than the q -th quantile, indicating a poor prediction, we consider the routing at time t for node n as incorrect (the worst route). The pseudo-label l_{Worst} is defined as:

$$l_{\text{Worst}} = \begin{cases} 0, & \text{if } L(y, \hat{y}) > \text{Quantile}_q \text{ and } e = \arg \max(\mathbf{p}), \\ \frac{1}{E-1}, & \text{if } L(y, \hat{y}) \leq \text{Quantile}_q \text{ and } e \neq \arg \max(\mathbf{p}), \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

The Avoid Worst Routing Loss is then defined as:

$$L_{\text{Worst}}(\mathbf{p}) = -\frac{1}{E} \sum_e l_{\text{Worst}} \log(p_e). \quad (32)$$

Our final loss function combines both the Best Choice Loss and the Avoid Worst Routing Loss:

$$L = L_{\text{Best}} + \lambda L_{\text{Worst}}, \quad (33)$$

where λ is a hyperparameter balancing the two loss terms. By incorporating both losses, we encourage the model to select the best teacher while avoiding the worst, thus improving overall performance.

3.4. Teaching the student

Although using multiple spatial modelling strategies and their dynamic routing can address various real-world scenarios, our goal is to train a model that simultaneously possesses multiple spatial modelling capabilities without needing a complex dynamic routing process. Therefore, we designed a lightweight student model with MSA and feedforward layers without additional temporal enhancement embeddings or spatial modelling methods. This design makes the model extremely lightweight, with parameter counts approximately 5% of those in GMAN (Zheng et al. 2020), resulting in improved computational efficiency. In order to equip the student model with the spatial modelling capabilities of the teacher model and unique insights into time series, we use the optimal output of the teacher model with the largest routing probability selected from Eq. (28) as the soft label of the student model, while the true values serve as hard labels for supervision. The structure of the student model is illustrated in Figure 5, where each layer consists solely of MSA and feedforward layers. The number of stacked layers will be discussed in the experimental section. Next, we describe how the student model generates prediction results, followed by the training process through distillation.

For the input data, we apply the MSA mechanism directly after embedding, i.e. $X_{\text{Stu}} = \text{MSA}(X_{\text{in}}, X_{\text{in}}, X_{\text{in}})$, with the resulting output serving as knowledge for the student model to learn alongside the teacher model. Finally, the output is obtained through a linear mapping: $X_{\text{out}} = \text{Proj}(X_{\text{Stu}})$. The teacher model is initially trained to converge, ensuring it provides unique insights into the input data. In contrast, the teacher model does not participate in backpropagation during the training of the student model. We compute two losses: the distillation loss L_{dis} and the hard label loss L_{hard} . The distillation loss is defined as:

$$L_{\text{dis}} = \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{T^2} \text{MAE} \left(\frac{X_{\text{Stu}}[i]}{T}, \frac{X_{\text{Best}}[i]}{T} \right) \right), \quad (34)$$

where T is the temperature parameter, N is the number of samples, and X_{Best} is the output from the teacher model. The hard label loss is defined as:

$$L_{\text{hard}} = \frac{1}{N} \sum_{i=1}^N L_{\text{MAE}}(y[i], \hat{y}[i]), \quad (35)$$

The final total loss is given by:

$$L = (1 - \alpha)L_{\text{dis}} + \alpha L_{\text{hard}}, \quad (36)$$

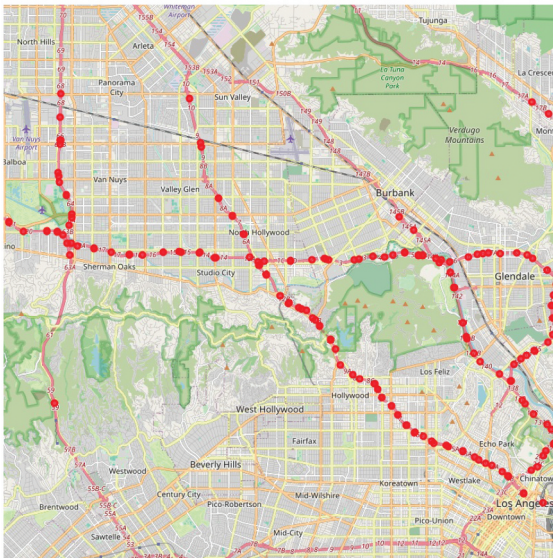
where α is a weight coefficient used to balance the influences of the distillation loss and the hard label loss. In the experimental section, we explore the effects of different choices of α and T on model performance.

4. Experiments

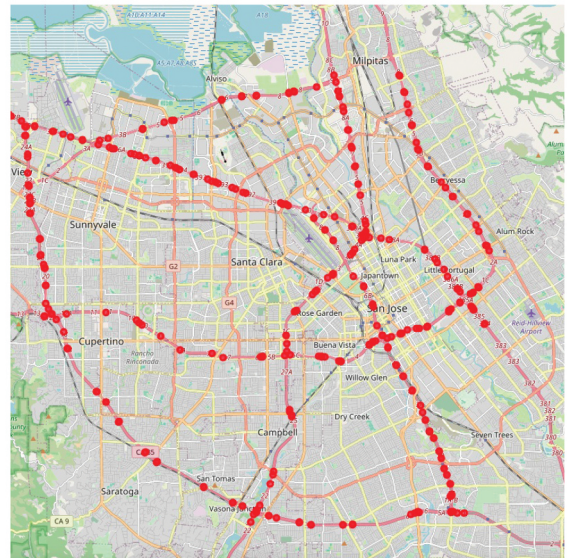
In this section, we introduce the used datasets and our experimental setup, and present the analysis of results to validate the DTSGFormer framework. The section covers data preprocessing, parameter configurations, and performance assessments across various prediction horizons, demonstrating the model's efficiency, adaptability, and overall effectiveness in traffic forecasting.

4.1. Datasets

Experiments are conducted using two extensive real-world traffic speed datasets introduced by Y. Li et al. (2018). The first, METR-LA, consists of traffic information gathered from loop detectors installed on highways in Los Angeles County (Jagadish et al. 2014). It involves 207 selected sensors, with data collected over a 4-month period between 1st March and 30 June 2012, resulting in a total of 6,519,002 observed traffic data points. The second, PEMS-BAY, is provided by the California Transportation Agencies (CalTrans) through the Performance Measurement System (PeMS). It includes traffic data from 325 sensors distributed across the Bay Area, collected over a 6-month period between 1st January and 31 May 2017, amounting to 16,937,179 observed traffic data points. The spatial distribution of sensors for both datasets is presented in Figure 7. Both benchmarks are large-scale, well-curated, and widely used for controlled comparison. Section 5 further evaluates DTSGFormer on the SupplyGraph supply-chain benchmark (Wasi, Islam, and Akib 2024, 2025).



(a) METR-LA



(b) PEMS-Bay

Figure 7. Distribution of sensors of (a) the METR-LA dataset and (b) the PEMS-BAY dataset.

4.2. Experimental setup

To evaluate the effectiveness of DTSGFormer, we compare its performance against both traditional and state-of-the-art methods that leverage GNNs, RNNs, and attention mechanisms for traffic prediction tasks. For the reader's reference, Table 1 provides a summary of the examined methods.

Following Y. Li et al. (2018), z-score normalisation is applied before training DTSGFormer. For the METR-LA and PEMS-BAY datasets, we use 70% of the data for training, 10% for validation, and 20% for testing. All experiments have been conducted on a system with eight NVIDIA A6000 GPUs running Ubuntu 22.04.5 LTS. The Adam optimiser is used with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. The hidden size $d = e = 32$, memory size $m = 20$, and the number of layers $l = 3$. We initialize parameters and embeddings using Xavier initialization, with the number of attention heads $K = 4$ and the hidden size of the feed-forward network $h_{ff} = 128$. The learning rate is adjusted during training using a cosine annealing with warm restarts scheduler (Z. Liu 2022) as per the following equation:

$$\text{lr}(T_{\text{cur}}) = \frac{1}{2}(\text{lr}_{\text{max}} - \text{lr}_{\text{min}}) \left[1 + \cos \left\{ \frac{T_{\text{cur}}}{T_{\text{freq}}} \pi \right\} \right] + \text{lr}_{\text{min}}, \quad (37)$$

where T_{cur} is the number of steps since the last restart. For all datasets, we set $T_{\text{warm}} = T_{\text{freq}} = 4000$, $\text{lr}_{\text{min}} = 10^{-7}$, and $\text{lr}_{\text{max}} = 3 \times 10^{-3}$.

We set the patience parameter to 10 to enable early stopping and prevent overfitting. The training loss function is the Mean Absolute Error (MAE). To assess the model's performance across varying prediction horizons, we compare MAE, RMSE, and MAPE for 15-minute (horizon 3), 30-minute (horizon 6), and 1-hour (horizon 12) forecasts, corresponding to short-term, mid-term, and long-term predictions, respectively. For all baseline models, we reproduce their results using the official open-source codes, validate our reproduced outcomes against those reported in their respective papers, and then incorporate them into our analysis. This approach aligns with Lee and Ko (2024) and R. Jiang et al. (2023).

Table 1. Summary of methods used for spatiotemporal data forecasting.

Method	Spatial dependency	Temporal dependency	Description
HA (Linares-Mustarós et al. 2019)	\	\	Predicts future speeds by using a weighted average of historical speed data.
STGCN (B. Yu, Yin, and Zhu 2018b)	GCN	CNN	A lightweight approach that predicts future traffic conditions by utilizing GCN
DCRNN (Y. Li et al. 2018)	GCN	GRU	A well-established model that integrates graph convolution into recurrent units
Graph (Wu et al. 2019)	GCN	CNN	Leverages graph embeddings for traffic prediction
STTN (M. Xu et al. 2021)	GCN	Attention	Utilizes spatial and temporal transformers to capture dynamic spatial dependencies
GMAN (Zheng et al. 2020)	Attention	Attention	A spatiotemporal attention model designed for traffic forecasting
MTGNN (M. Jin et al. 2022)	GCN	CNN	Designed for multivariate time series combines mix-hop propagation
AGCRN (Bai et al. 2020)	GCN	GRU	A model that combines GNN with RNN
CCRNN (Ye et al. 2021)	GCN	GRU	Learns multiple parameterized matrices for different layers of graph convolution.
GTS (Z. Shao et al. 2022)	GCN	Attention	Models the probability of each link based on the long historical data of each variable.
PM-MemNet (Lee et al. 2021)	Memory GCN	Attention	Employs memory networks for traffic pattern matching.
DMGAN (R. Li et al. 2023)	GCN	GRU	A Multi-Hop Graph Attention Network that utilizes dynamic graph modeling
RGDAN (Fan et al. 2024)	Random GAT	Attention	An advanced approach that utilizes random graph attention
AdpSTGCN (Chen and Sun 2024)	GCN	Attention	Employs multi-head attention to generate multi-view feature graphs
DTSGFormer (Our Study)	Variable/Spatial-enhanced Attention	Temporal/Memory-enhanced Attention	Simultaneously integrates multiple spatial modeling techniques via Dynamic Teacher-Student Gating.

Table 2. Summary of model performance on the METR-LA and PEMS-BAY datasets.

METR-LA	15 min			30 min			60 min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA (2018)	4.16	7.80	13.00%	4.16	7.80	13.00%	4.16	7.80	13.00%
STGCN (2018)	2.88	5.74	7.62%	3.47	7.24	9.57%	4.59	9.40	12.70%
DCRNN (2018)	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.59	10.50%
Graph-WaveNet (2019)	2.69	5.15	6.90%	3.07	6.22	8.37%	3.53	7.37	10.01%
STTN (2020)	2.79	5.48	7.19%	3.16	6.50	8.53%	3.60	7.60	10.16%
GMAN (2020)	2.80	5.55	7.41%	3.12	6.49	8.73%	3.44	7.35	10.07%
MTGNN (2020)	2.69	5.18	6.86%	3.05	6.17	8.19%	3.49	7.23	9.87%
AGCRN (2020)	2.86	5.55	7.55%	3.25	6.57	8.99%	3.68	7.56	10.46%
CCRNN (2021)	2.85	5.54	7.50%	3.24	6.54	8.90%	3.73	7.65	10.59%
GTS (2021)	2.65	5.20	6.80%	3.05	6.22	8.28%	3.47	7.29	9.83%
PM-MemNet (2022)	2.65	5.29	7.01%	3.03	6.29	8.42%	3.46	7.29	9.97%
DMGAN (2023)	2.61	5.05	6.66%	2.99	6.11	8.04%	3.42	7.22	9.65%
RGDAN (2024)	2.69	5.20	7.14%	2.99	5.98	8.07%	3.26	7.02	9.73%
AdpSTGCN(2024)	2.59	5.11	6.68%	2.96	6.08	8.02%	3.40	7.21	9.45%
DTSGFormer(Our Study)	2.63	4.77	6.66%	2.96	5.69	7.93%	3.56	6.86	10.31%
PEMS-BAY	15 min			30 min			60 min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA (2018)	2.88	5.59	6.80%	2.88	5.59	6.80%	2.88	5.59	6.80%
STGCN (2018)	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
DCRNN (2018)	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
Graph-WaveNet (2019)	1.30	2.74	2.73%	1.63	3.70	3.67%	1.95	4.52	4.63%
STTN (2020)	1.36	2.87	2.89%	1.67	3.79	3.78%	1.95	4.50	4.58%
GMAN (2020)	1.35	2.90	2.87%	1.65	3.82	3.74%	1.92	4.49	4.52%
MTGNN (2020)	1.32	2.79	2.77%	1.65	3.74	3.69%	1.94	4.49	4.53%
AGCRN (2020)	1.36	2.88	2.93%	1.69	3.87	3.86%	1.98	4.59	4.63%
CCRNN (2021)	1.38	2.90	2.90%	1.74	3.87	3.90%	2.07	4.65	4.87%
GTS (2021)	1.34	2.84	2.83%	1.67	3.83	3.79%	1.98	4.56	4.59%
PM-MemNet (2022)	1.34	2.82	2.81%	1.65	3.76	3.71%	1.95	4.49	4.54%
DMGAN (2023)	1.27	2.69	2.66%	1.58	3.65	3.57%	1.87	4.43	4.44%
RGDAN (2024)	1.31	2.79	2.77%	1.56	3.55	3.47%	1.82	4.20	4.28%
AdpSTGCN (2024)	1.29	2.75	2.69%	1.64	3.69	3.67%	1.92	4.49	4.62%
DTSGFormer(Our Study)	1.20	2.37	2.41%	1.48	3.19	3.26%	1.80	4.03	4.13%

4.3. Overall results

To facilitate comparison with baseline results, we evaluate the three-layer, Transformer-based DTSGFormer model across 3-step, 6-step, and 12-step prediction horizons on two datasets. As shown in Table 2, DTSGFormer achieves the best performance at most time steps. However, its performance declines slightly for 12-step predictions, likely due to its limited number of parameters, only a fraction of those in GMAN, which affects its overall effectiveness. The success of DTSGFormer can be attributed to the diverse spatial modelling strategies employed by the teacher models. These strategies adapt well to various real-world conditions and foster robust learning. Furthermore, the prediction-assessment-teaching framework of the teacher-student structure exploits inherent imbalances across time steps and variables, ensuring that the teacher routes information precisely to guide the student model effectively.

Models that learn static graphs (e.g. Graph-WaveNet, MTGNN, and CCRNN) and those that learn dynamic graphs (e.g. STTN and GMAN) also demonstrate competitive performance, showcasing the benefits of these approaches. In terms of temporal modelling, however, RNN-based models (e.g. DCRNN) struggle with long-term predictions due to error accumulation. In contrast, PM-MemNet maintains its advantage even in long-term forecasts by injecting memory-enhanced representation vectors into the decoder. Unlike these approaches, DTSGFormer does not employ a dedicated spatial modelling mechanism. Instead, it indirectly learns spatial dependencies through the hidden states of multiple teacher models. Despite this indirect approach, it still outperforms models like AGCRN, which uses adaptive graphs, and RGDAN, which employs random graph attention. This result underscores the strength of the proposed DTSGF mechanism.

DTSGFormer exhibits significant advantages in 15-minute and 30-minute predictions, particularly in terms of MAE and RMSE, underscoring its effectiveness for short- and medium-term traffic speed forecasting. This indicates that DTSGFormer effectively leverages dynamic spatial relationships and temporal information to capture complex spatiotemporal dependencies. On the PEMS-BAY dataset, DTSGFormer also performs exceptionally well, surpassing baseline models across all evaluation metrics. Notably, it demonstrates consistent performance across different time horizons, highlighting its adaptability to varying temporal scales.

4.4. Ablation studies

We conduct ablation studies to focus on three key questions: (i) What advantages does DTSGF offer over other multi-teacher distillation methods? (ii) Can we effectively apply the DTSGF framework to different student model architectures? and (iii) How does the number of stacked Transformer layers in DTSGFormer influence its performance, and can we derive practical guidelines for selecting the optimal layer configuration?

To evaluate the advantages of DTSGF, we consider two variants: w/o distillation and ensemble. In the w/o distillation setting, we train the student model directly without any teacher guidance; as expected, all distillation methods outperform this baseline, demonstrating the inherent value of incorporating teacher models. Table 3 shows that both MLP and Transformer student models produce higher MAE, RMSE, and MAPE values without distillation than they do under any distillation-based approach. In the ensemble variant, we replace the DTSGF framework with a simple weighted averaging of multiple teachers, a technique previously validated in 2017 Allen-Zhu and Li 2020; Zhu et al. 2018). Although the ensemble method surpasses the w/o distillation baseline, it still falls short of the DTSGF framework across all time horizons and datasets. These results underscore the effectiveness of DTSGF's dynamic routing and prediction-evaluation-teaching paradigm, which provides more accurate guidance for the student model. Notably, the terms MLP and Transformer in Table 3 refer to student models employing these architectures as their base. In both cases, DTSGF consistently outperforms the ensemble and w/o distillation variants, achieving significantly lower MAE, RMSE, and MAPE values – especially for short-term (15 min) and mid-term (30 min) predictions. Since DTSGF can be viewed as an enhanced version of both augmented and switched multi-teacher distillation methods, we did not separately compare these approaches. Nevertheless, the results clearly show that DTSGF excels in leveraging spatiotemporal dynamics and outperforms ensemble-based distillation, demonstrating its robustness across different student architectures and diverse time horizons.

Regarding the second question, as shown in 3, we extend the DTSGF framework to include MLP models as student models. Specifically, the MLP student model comprises 18 fully connected layers, each with a hidden size of $h = 128$, followed by a final linear transformation layer for output. When compared to the ensemble method and the approach without distillation, the MLP student model within the DTSGF framework demonstrates significant improvements. Notably, training the MLP model directly without any form of distillation yields poor results, particularly for the 15-minute prediction on the METR-LA dataset. This underscores the critical role of the teacher models in DTSGF, which provide additional knowledge and also act as supervision that prevents the student model from overfitting. Through the DTSGF framework, teacher models guide the student model to generalize better, especially in scenarios involving complex temporal dependencies. In contrast, the ensemble method offers minimal performance improvements or, in some cases, no improvement for Transformer-based student models. This is likely because, in the ensemble approach, the teacher models share the same architecture as the student model (i.e. both are based on Transformers), making it difficult for the student to extract diverse and useful information by simply averaging the outputs of multiple identical teacher models. Conversely, the DTSGF framework dynamically

Table 3. Comparison of the effects of different distillation methods and different student model structures.

	15 min			30 min			60 min		
METR-LA	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
MLP w/o Distillation	9.00	12.30	28.41%	3.88	7.52	11.51%	4.04	7.67	11.93%
MLP-ensemble	2.84	5.21	7.65%	3.36	6.04	9.05%	3.98	7.32	11.29%
DTSGF-MLP	2.76	5.08	6.97%	3.07	5.89	8.34%	3.67	7.01	10.24%
Transformer w/o Distillation	2.78	5.17	7.00%	3.09	6.00	8.37%	3.72	6.96	10.92%
Transformer-ensemble	2.78	5.18	7.00%	3.07	5.96	8.29%	3.55	7.06	10.82
DTSGFormer	2.63	4.77	6.66%	2.96	5.69	7.93%	3.56	6.86	10.31%
PEMS-BAY	15 min			30 min			60 min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
MLP w/o Distillation	1.30	2.50	2.75%	5.19	8.62	14.56%	2.32	4.78	5.64%
MLP-ensemble	1.25	2.45	2.60%	1.56	3.25	3.35%	2.10	4.38	4.95%
DTSGF-MLP	1.24	2.46	2.50%	1.55	3.24	3.33%	2.02	4.17	4.67%
Transformer w/o Distillation	1.43	2.80	3.22%	1.65	3.33	3.70%	1.97	4.15	4.62%
Transformer-ensemble	1.26	2.47	2.54%	1.56	3.22	3.34%	1.85	4.17	4.24%
DTSGFormer	1.20	2.37	2.41%	1.48	3.19	3.26%	1.80	4.03	4.13%

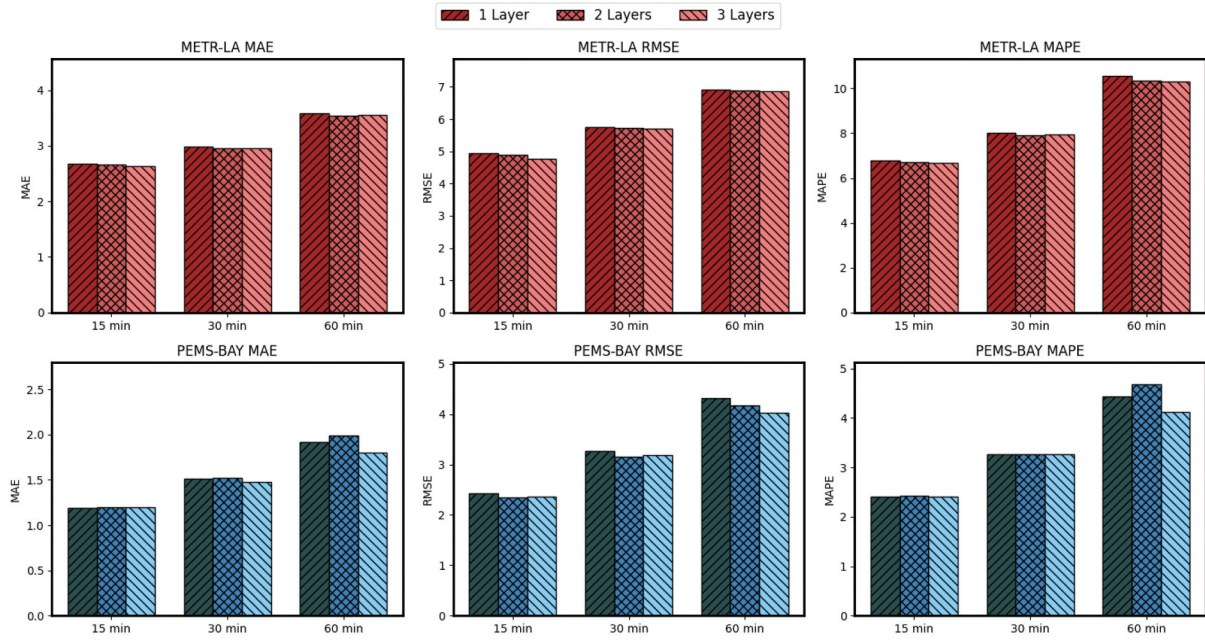


Figure 8. Impact of stacking 1, 2, and 3 Transformer layers within the DTSGFormer framework on the METR-LA and PEMS-BAY datasets. Performance is measured using MAE, RMSE, and MAPE over 15 min, 30 min, and 60 min prediction horizons.

routes the most suitable teacher to the student model at each time step, ensuring that the student benefits from more meaningful and varied guidance.

Finally, we investigate the impact of stacking different numbers of Transformer layers (1, 2, and 3) within the DTSGF framework and report the results on two datasets. As illustrated in Figure 8, varying markers represent the different layer configurations. The results clearly indicate that increasing the number of stacked layers generally enhances performance across various prediction horizons. This improvement is especially pronounced in long-term predictions (60 minutes), where additional Transformer layers significantly reduce error metrics. Specifically, on both the METR-LA and PEMS-BAY datasets, models with three stacked layers consistently outperform those with one or two layers, demonstrating that deeper architectures can capture more complex spatiotemporal dependencies and generalise better over longer horizons. However, it is important to note that while performance continues to improve with more layers, the rate of improvement decreases as the number of layers increases. The gains from increasing from one to two layers are more substantial than those from two to three layers, suggesting that there may be diminishing returns with further stacking.

4.5. Efficiency analysis

For efficiency analysis, we include several state-of-the-art models for comparison, including STGCN (B. Yu, Yin, and Zhu 2018b), one of the earliest spatio-temporal graph convolutional network models; DCRNN (Y. Li et al. 2018), which combines diffusion convolution with recurrent neural networks for spatio-temporal sequence modelling; Graph-WaveNet (Wu et al. 2019), utilising a learnable adjacency matrix in graph convolution to capture dynamic spatio-temporal dependencies; GMAN (Zheng et al. 2020), which employs a graph attention mechanism to model complex traffic patterns; RGDAN (Fan et al. 2024), designed to dynamically model complex traffic graph structures; and DMGAN (R. Li et al. 2023), a dynamic multi-hop graph attention network for time-varying spatio-temporal dependencies. We also considered other models for comparison but excluded them after careful evaluation. For instance, MTGNN (M. Jin et al. 2022) was excluded as it is an enhanced version of Graph-WaveNet with similar computational costs. Similarly, AGCRN (Bai et al. 2020) and CCRNN (Ye et al. 2021) were excluded because they are variants of DCRNN with only minor differences in computational costs. All models have been tested under identical experimental settings, with training and inference times

recorded using the same hardware. This uniform setup ensures a fair comparison of computational costs, allowing for a more accurate assessment of the resource consumption of different models.

Table 4 summarises the experimental results, including training time, inference time, and model parameter sizes for various models on traffic prediction tasks. Our DTSGFormer model, evaluated with different numbers of Transformer layers (1, 2, and 3), demonstrates competitive performance while utilising fewer parameters and requiring shorter training times compared to baseline models. Specifically, the 1-layer model requires approximately 17.6 seconds per epoch, whereas the 3-layer model takes around 22.7 seconds, highlighting significant efficiency improvements over other baseline approaches. Additionally, Figure 9 illustrates these findings, showing that our model achieves superior performance with fewer parameters and consistently outperforms traditional spatial modelling methods on real-world benchmark datasets.

4.6. Sensitivity analysis

To investigate the impact of hyperparameters on model performance, we conduct a comprehensive parameter study focusing on two key hyperparameters: the temperature T in the distillation process and the balance factor α between soft and hard labels for the student model. Specifically, temperature is varied within the range $T = \{1, 2, 3, \dots, 9\}$. As T increases, the teacher model generates softer probability distributions, which facilitate better generalisation for the student model. However, excessively high temperatures overly smooth the distribution, diluting critical information, while temperatures that are too low result in sharp logits, making it difficult for the student to learn nuanced patterns from the teacher's guidance. The balance factor $\alpha = \{0.1, 0.2, 0.3, \dots, 0.9\}$ determines the relative contribution of the teacher-generated soft labels and the ground-truth hard labels in the student's learning process. A higher α places more emphasis on the soft labels, enhancing knowledge transfer from the teacher, but may also cause the model to overfit to the teacher's outputs, potentially neglecting important task-specific information from the hard labels.

Table 4. Comparison of computational costs improvements across different baseline models for traffic prediction tasks.

Model	Training time/epoch	Inference time
STGCN	14.8 secs	16.70 secs
DCRNN	122.22 secs	13.44 secs
Graph-WaveNet	48.07 secs	3.69 secs
GMAN	312.1 secs	33.7 secs
RGDAN	68 secs	6.3 secs
DMGAN	765.2 secs	76.8 secs
Our Study[1 layer]	17.6 secs	0.21 secs
Our Study[2 layers]	20.7 secs	0.49 secs
Our Study[3 layers]	22.7 secs	0.61 secs

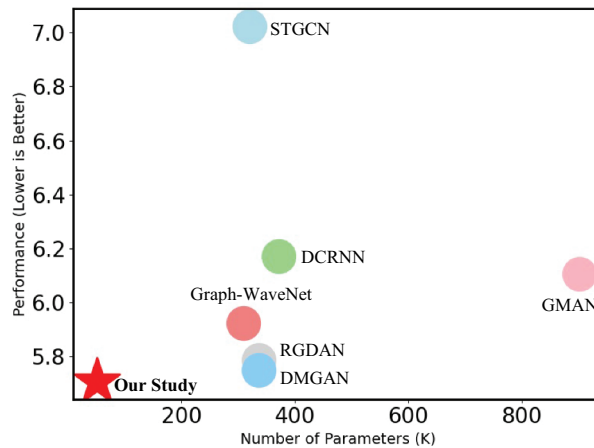


Figure 9. Computational cost improvements across baseline models for traffic prediction tasks. The closer to the lower left corner, the higher the performance and efficiency.

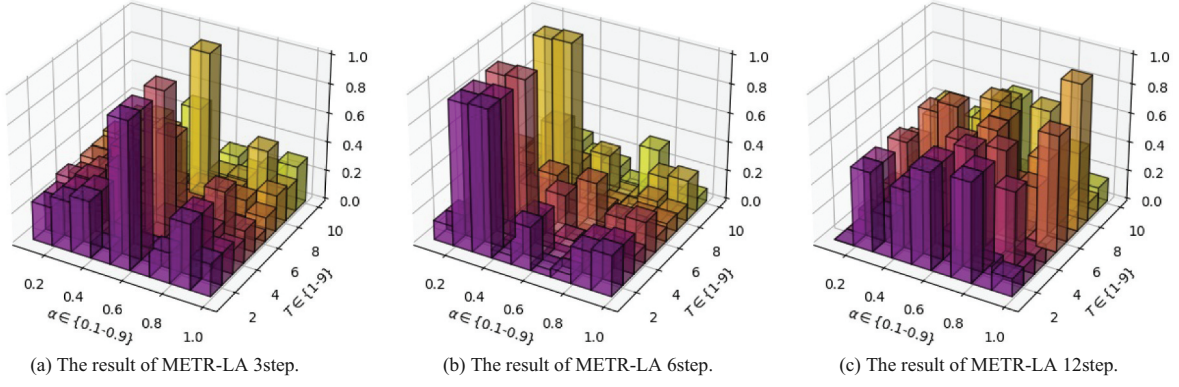


Figure 10. Parameter search on temperature (T) and balance factor α during distillation on the METR-LA dataset. The x-axis shows α values, and the y-axis shows T .

Conversely, a lower α prioritizes the hard labels, which can improve task-specific performance but may limit the benefits of knowledge distillation. Each experiment is repeated five times, and we report the MAE, RMSE, and MAPE for each test set. During each experiment, only one hyperparameter is varied while keeping all others constant to isolate its specific effect. These experiments are performed on the METR-LA dataset, with results presented for 3-step, 6-step, and 12-step prediction horizons. This systematic approach allows us to identify the optimal settings for T and α , thereby enhancing the student model's performance through effective hyperparameter tuning.

From Figure 10, we observe that for different values of α , the error initially decreases and then increases as T rises, indicating the presence of an optimal temperature that minimizes validation error. A low temperature fails to soften the logits sufficiently, making it difficult for the student model to learn subtle hints from the teacher. Conversely, excessively high temperatures result in an overly smooth distribution, diluting the critical information useful for the student. Thus, moderate temperature values strike the best balance for effective knowledge transfer. In the 3-step prediction, the optimal α is approximately 0.4, while in 6-step and 12-step predictions, it rises to around 0.8. This suggests that as the prediction horizon increases, the student model's ability to handle the task weakens, necessitating more knowledge from the teacher model and thus a larger α . Similarly, higher temperatures allow the teacher model to soften its output sufficiently, facilitating better learning for the student. As a result, the optimal distillation temperature increases with the prediction step size. In summary, the optimal hyperparameter values depend on the characteristics of the dataset and the specific task, ultimately determining the model's performance.

5. Discussion

5.1. Deployment and integration implications

We close by situating DTSGF within the operational requirements of enterprise information systems. Since the teacher pool and the routing module are exercised only during training, the deployed artefact reduces to a compact three-layer Transformer. Table 5 relates this artefact to specific device classes. DTSGFormer contains 117,868 parameters with a model size of 0.45 MB, fewer than STGCN, Graph-WaveNet, and GMAN. Its memory share is 0.0072% on an NVIDIA Jetson Orin Nano, 0.0144% on an NVIDIA Jetson Nano, and 0.4598% on an ATC-class roadside signal controller with 128 MB DRAM. This footprint makes provisioning at regional operation centres and at roadside units closer to the sensing layer feasible. Because DTSGFormer consumes and emits standard multivariate tensors, it can be embedded as a forecasting service behind message queues or REST endpoints alongside SCADA-style data hubs, transportation management systems, and ERP-linked dashboards, without altering upstream pipelines or downstream decision modules. At inference a single forward pass of the student suffices, keeping end-to-end latency within the update cadence typical of traffic-signal control, fleet dispatching, and incident alerting, while periodic retraining can be scheduled during off-peak windows. Beyond traffic speed forecasting, Section 3.1 defines the model pool,

Table 5. Deployment footprint of DTSGFormer on METR-LA. (a) Parameter count and model size of DTSGFormer and baseline models. (b) Memory share of DTSGFormer on representative device classes.

(a)					
Model		Parameters	Model size (MB)		
STGCN		246,028	0.94		
Graph-WaveNet		309,400	1.18		
GMAN		209,923	0.80		
DTSGFormer (Our Study)		117,868	0.45		
(b)					
Device class	Example device	Compute	Memory	Power	Memory share
Data-center GPU	NVIDIA RTX A6000	10,752 CUDA cores	48 GB GDDR6	300 W	0.0012%

(Continued)

the routing function, and the knowledge-transfer loss in domain-agnostic form. Under this abstraction DTSGF transfers to demand forecasting across multi-product catalogues, capacity and resource allocation under non-stationary workloads, and manufacturing process optimisation where operating regimes shift between product runs, yielding an edge-deployable surrogate for downstream enterprise decision systems. In practical traffic-management stacks, this placement maps to three concrete integration touchpoints. A stateless forecasting service emits 15-minute, 30-minute, and 60-minute horizon forecasts into the publish-subscribe topics already used by traffic-signal control, ramp metering, and fleet dispatching, which preserves the upstream pipeline and requires only a thin client wrapper on the consumer side. An edge instance of the compact student Transformer runs on roadside units next to loop detectors, which keeps forecasts available when wide-area connectivity is degraded and lets regional operation centres fall back to a central instance otherwise. An operator-facing audit panel embedded in advanced traffic management dashboards then displays the actual-versus-predicted visualisation in [Figure 11](#), so that traffic engineers can visually verify model behaviour during sustained congestion, transient sensor dropouts, and peak-period transitions before signing off on automated decisions in post-incident reviews.

5.2. Interpretability through actual-versus-predicted forecasts

Aggregate error metrics alone are insufficient for enterprise stakeholders who must judge whether a forecast is actionable in a specific operating regime. To complement the quantitative evaluation, we visualise DTSGFormer predictions against ground-truth speed trajectories on six randomly sampled 24-hour windows from the METR-LA test set, as reported in [Figure 11](#). Each panel annotates the corresponding sensor index and observation window, so operators can directly inspect the agreement between forecasts and observations.

Three operationally meaningful behaviours emerge from the displayed cases. Under free-flow conditions at Sensors 135 and 104, the prediction tracks the slow drift around 60 to 70 mph and stays stable when the ground-truth trace exhibits transient dropouts or isolated spikes, so the forecast suppresses sensor artefacts and keeps them out of downstream dispatch logic. During sustained congestion events at Sensors 140 and 115, the predicted curve reproduces both the onset and the recovery of the speed trough, thereby transmitting the correct direction and magnitude of the regime change. For the morning transition captured at Sensor 26 and the noisier diurnal trace at Sensor 117, the model recovers the acceleration from off-peak to day-time speed without overshooting the peak, while returning a phase-aligned yet smoothed estimate for the noisier segments.

These case-level displays make it possible for enterprise users to see where DTSGFormer is confident and where it filters isolated anomalies, which clarifies how the forecast should be weighted inside routing, alerting, and capacity-allocation workflows. The same plotting protocol can be reused as a runtime audit trail whenever DTSGFormer is embedded in a live operations dashboard, offering a transparent evidence layer that complements the quantitative accuracy reported in the earlier tables.

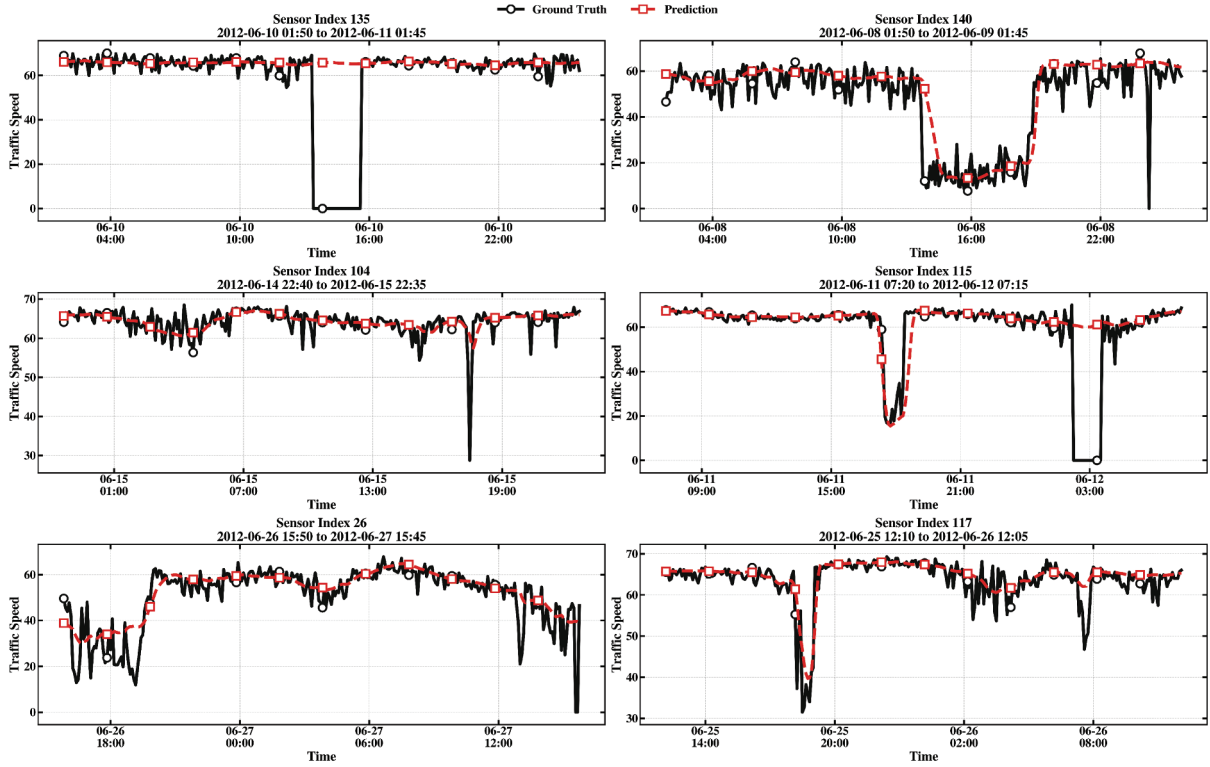


Figure 11. Actual versus predicted traffic speed on six randomly selected 24-hour windows from the METR-LA test set. Each panel reports the ground truth as a solid black line and the DTSGFormer prediction as a dashed red line, with the sensor index and observation window indicated in the title.

5.3. Dataset scope and cross-domain generalizability

To examine DTSGFormer beyond road traffic, we discuss the public SupplyGraph benchmark (Wasi, Islam, and Akib 2024, 2025) as a representative supply-chain extension. Table 6 reports horizon-specific comparative results on this benchmark and shows the DTSGFormer performance alongside established baselines.

After removing duplicated node identifiers, the benchmark contains 40 unique nodes, 221 daily observations, and 403 undirected links, and it is commonly evaluated with a 12-step input and 12-step output forecasting protocol. Under the standard temporal split, this produces 132 training windows, 22 validation windows, and 21 test windows, which is a much smaller temporal sample than METR-LA and PEMS-BAY. This shift matters because the graph remains relatively dense while the time axis is short, and because zero operational quantities can be semantically meaningful observations. The results in Table 6 show that DTSGFormer achieves the best MAE, RMSE, and MAPE at horizons 3, 6, and 12 on this supply-chain benchmark, while among the baselines DCRNN is strongest at horizon 3, GWNet is strongest at horizons 6 and 12, and STGCN remains competitive. These results provide initial evidence that DTSGFormer extends to supply planning, logistics coordination, and graph-based demand forecasting, while broader enterprise datasets are needed to establish cross-domain transfer in general.

Table 6. Horizon-specific performance of DTSGFormer and baseline models on the SupplyGraph benchmark.

SupplyGraph	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HI	8.39	19.72	3.20%	8.22	19.25	3.41%	9.04	21.61	3.19%
STGCN	3.26	8.23	1.39%	3.40	8.46	1.23%	3.93	10.14	0.95%
DCRNN	2.93	7.98	1.47%	3.07	8.15	1.29%	3.56	9.85	1.00%
GWNet	3.13	8.50	1.28%	3.00	7.89	1.51%	3.45	9.57	0.80%
MTGNN	5.08	11.93	1.82%	4.83	11.93	1.77%	4.92	11.66	1.55%
AGCRN	5.12	12.27	2.19%	5.13	12.30	2.62%	5.56	13.77	1.46%
GTS	4.94	11.68	1.89%	5.44	12.61	3.95%	6.29	14.12	2.59%
DTSGFormer (Our Study)	2.70	5.78	1.01%	2.89	5.85	0.93%	3.06	8.67	0.74%

5.4. Business case study: forecast-driven versus reactive supply-chain planning

We simulate a daily operations loop on the SupplyGraph network introduced above and compare two plans at the strategy layer. The reactive plan reserves capacity from the most recent realised demand, performs cross-node transfers only after local shortfalls materialise, and triggers premium emergency fulfilment as a last resort. The forecast-driven plan is powered by DTSGFormer, which supplies the node-level multi-step predictions consumed at every planning step: these predictions drive capacity pre-reservation, trigger cross-node transfers along the supply graph before demand arrives, and leave emergency fulfilment only for residual shocks that the network cannot absorb. The observed gains therefore measure how DTSGFormer's accuracy translates into downstream operational value for enterprise decision makers.

The simulation retains the SupplyGraph test split, with 40 nodes, 12-step inputs, and 12-step outputs, and is evaluated at three planning horizons that correspond to short-term dispatch, mid-term coordination, and longer-term capacity decisions. Table 7 records the cost calibration, in which the relative magnitudes reflect the managerial intuition that idle capacity is mildly costly, network-level transfers are moderately costly, and emergency fulfilment is the most expensive recourse.

Table 8 reports the operational outcomes. The forecast-driven plan lowers unit cost by 55.37% at horizon 3, by 57.54% at horizon 6, and by 60.61% at horizon 12, and the gain widens as the decision window grows because the reactive plan accumulates more uncorrected shocks. Emergency fulfilment drops from the 36% to 54% range under the reactive plan to the 16% to 21% range under the forecast-driven plan, while the in-network service level rises from the 46% to 64% range to the 79% to 84% range, so the same network absorbs far more of the realised demand once forecasts steer the reservation step. Transfer ratios also decline under the forecast-driven plan, signalling that cross-node moves shift from reactive firefighting to planned rebalancing before demand materialises.

Beyond horizon-level aggregates, the simulation surfaces a small set of persistently high-stress nodes, notably SOS001L12P, SOS005L04P, ATN01K24P, and SOS002L09P, that continue to concentrate residual stress even under the forecast-driven plan. Enterprise operators can use this signal to allocate additional safety capacity at structural hubs, raise monitoring thresholds, and pre-provision redundant transfer paths, thereby turning DTSGFormer outputs into a direct input for capacity planning. Taken together, the case study shows that the accuracy of DTSGFormer translates into operational value through three EIS-relevant levers: capacity reservation, cross-node transfer coordination, and emergency fulfilment control.

5.5. Trade-offs between model complexity and performance

Enterprise adoption depends on both headline accuracy and how complexity translates into cost along training, inference, and maintenance axes, and DTSGF is structured around an asymmetric complexity split in which the teacher pool, the memory-enhanced graph module, and the routing loss are active only at training

Table 7. Cost calibration used in the business case study on SupplyGraph.

Cost component	Value
Capacity unit cost	1.00
Idle capacity penalty	0.15
Inter-node transfer cost per hop-unit	0.35
Emergency fulfillment cost per unit	1.25

Table 8. Operational outcomes of the reactive plan and the forecast-driven plan on SupplyGraph across three planning horizons. Lower cost, lower emergency ratio, and higher in-network service level are preferable. In the column header, SL abbreviates service level.

Horizon	Plan	MAE	Cost/Unit	In-network SL	Emergency Ratio	Transfer Ratio
H = 3	Reactive plan	657.43	0.5391	63.52%	36.48%	4.50%
	Forecast-driven plan	293.39	0.2406	83.72%	16.28%	2.01%
H = 6	Reactive plan	722.62	0.5830	61.83%	38.17%	7.16%
	Forecast-driven plan	306.83	0.2475	83.79%	16.21%	3.04%
H = 12	Reactive plan	903.91	0.8042	46.39%	53.61%	4.83%
	Forecast-driven plan	356.03	0.3168	78.89%	21.11%	1.90%

Table 9. Complexity and performance trade-off axes for DTSGF and their recommended settings.

Design axis	Dominant cost	Primary benefit	Recommendation
Teacher pool size	Training wall time	Regime coverage	Compact pool of spatially diverse teachers
Student depth L	Training and inference	Long-horizon accuracy	$L = 3$ by default, $L = 2$ when 15-minute responsiveness dominates
Temperature T	Negligible direct cost	Softness of teacher signal	Moderate value, grows with horizon
Teacher weight α	Negligible direct cost	Distillation emphasis	$\alpha \approx 0.4$ at short horizons, $\alpha \approx 0.8$ at long horizons

time while the deployed artefact reduces to the compact student Transformer. Consistent with this split, Table 4 shows that DTSGFormer trains in 17.6 to 22.7 seconds per epoch across one to three student layers and that inference collapses to 0.21 to 0.61 seconds per batch once the teacher pool is detached, so the higher training cost is amortised once during offline preparation while live workloads enjoy stable low-latency inference. The layer-count ablation in Figure 8 then shows monotonic gains from one to three layers with the largest improvements concentrated at the 60-minute horizon and diminishing returns at short horizons, which favours a two-layer student when 15-minute responsiveness dominates and a three-layer student when 30-minute or 60-minute forecasts drive the decision loop. The sensitivity surface in Figure 10 further indicates that the optimal temperature T and balance factor α both grow with the prediction horizon, with 15-minute forecasts favouring moderate T and α near 0.4 and 60-minute forecasts preferring higher T and α near 0.8, so operators serving multiple horizons can index these settings against the target horizon. Enlarging the teacher pool raises training wall time roughly linearly but leaves deployment cost unchanged, so enterprises should start from a compact pool of spatially diverse teachers and grow it only when regime diversity observed during training justifies the additional compute, and Table 9 consolidates these axes alongside their recommended settings so that enterprise architects can align DTSGF parameters with existing procurement decisions where training budget governs the teacher pool and layer count, deployment budget is bounded by student depth alone, and service-level targets shape the horizon-specific hyperparameters.

6. Limitations and future work

Despite the overall strength of DTSGFormer across 15-minute, 30-minute, and 60-minute horizons, several limitations remain. Although DTSGFormer still delivers the lowest errors at every horizon on both METR-LA and PEMS-BAY, its margin over the strongest graph baselines narrows at the 60-minute horizon, and the absolute error metrics themselves grow because recursive expansion of the prediction window amplifies compounding uncertainty and sensor noise, while the layer-count ablation in Figure 8 confirms that simply stacking more Transformer layers offers progressively smaller improvements beyond the three-layer configuration. The teacher pool is also fixed at training time, so new spatial modelling strategies cannot be absorbed after deployment without a retraining cycle, and the current routing mechanism uses hard selection that discards the soft agreement between closely related teachers. In addition, although routing decisions are governed by an explicit similarity rule and the compound Best Choice and Avoid Worst Routing losses, the framework does not yet externalise routing frequencies or regime-specific activation statistics as an operator-facing audit layer, which limits direct inspection of selection behaviour in enterprise decision contexts. Moreover, the empirical evidence remains concentrated on traffic speed forecasting and one supply-chain benchmark, so generalisation to regimes with richer exogenous covariates or with event-driven disruptions such as incidents, weather shocks, and promotional demand spikes has not yet been established. Finally, the current framework outputs point predictions only, which limits its direct use in enterprise decisions that require explicit risk quantification.

Future work targets these limitations along five complementary lines. For long-horizon accuracy, we plan to combine DTSGF with hierarchical or recurrent distillation schemes that use short-horizon forecasts as auxiliary signals for longer horizons, and to incorporate extended-context attention layers that reason over longer historical windows without inflating student depth. For model extensibility, we will study continual learning mechanisms that allow new teachers to be appended after deployment and introduce calibrated soft routing during distillation so that multiple teachers can contribute when their representations agree. For routing transparency, we will expose routing frequencies and regime-specific activation statistics as an explicit audit channel, so that enterprise operators can review which

teachers are selected under which operating conditions and flag anomalous switching behaviour before it propagates into downstream decision modules. For broader validation, we will extend evaluation to additional EIS-relevant benchmarks covering multimodal logistics, demand forecasting with exogenous covariates, and manufacturing optimisation, so that generalisability claims rest on evidence drawn from operational domains beyond traffic alone. For decision-ready outputs, we plan to augment DTSGFormer with uncertainty quantification modules such as conformal prediction or quantile regression heads, so that enterprise decision modules can reason about prediction intervals in addition to point estimates.

7. Conclusion

This paper introduces DTSGF, a novel multi-teacher distillation method that distinguishes itself from previous ensemble, switching, and augmented frameworks by dynamically selecting the optimal expert for each time step and node in the student model through a routing mechanism. This dynamic selection ensures that the student model receives precise guidance tailored to specific spatiotemporal patterns, thereby enhancing its adaptability and performance. Building upon this framework, we design DTSGFormer, a model comprising three layers of Transformers. Extensive experiments on real-world datasets demonstrate that DTSGFormer achieves superior performance with only a fraction of the parameters compared to existing baseline models, underscoring the efficiency and effectiveness of our approach. Effective traffic management is vital for the sustainable development of cities, as proper transportation planning alleviates congestion, enhances traffic efficiency, and improves environmental quality and the urban image. Given its high computational efficiency and performance, DTSGFormer significantly contributes to addressing these challenges. For future work, we aim to further explore the generalisability of the DTSGF framework, including its application to other tasks such as classification, where its adaptive and dynamic routing capabilities are expected to offer notable advantages.

Note

1. The PEMS04 dataset is a spatiotemporal dataset from the Caltrans Performance Measurement System (PEMS), focused on traffic data from California's District 4, covering the San Francisco Bay Area. It is publicly available at <https://dot.ca.gov/programs/traffic-operations/mpr/pems-source> and is widely used in traffic prediction research for its detailed records of traffic speed, flow, and occupancy levels across numerous freeway sensors.

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grant 72293563, the Basic Scientific Research Project of Liaoning Provincial Department of Education under Grant JYTZD2023050, the Natural Science Foundation of Liaoning Province, China under Grant 2024-MS-175 and the Dalian Scientific and Technological Talents Innovation Support Plan under Grant 2022RG17. No potential competing interest was reported by the author(s).

Author contributions

CRedit: **Guangyu Wang:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft; **Yujie Chen:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization; **Ming Gao:** Writing – review & editing; **Jingmin An:** Writing – review & editing; **Zhiqiao Wu:** Writing – review & editing; **Jiafu Tang:** Writing – review & editing; **Bowei Chen:** Writing – review & editing.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work was supported by the National Natural Science Foundation of China [72293563]; Natural Science Foundation of Liaoning Province [2024-MS-175]; Dalian Scientific and Technological Talents Innovation Support Plan [2022RG17]; Basic Scientific Research Project of Liaoning Provincial Department of Education [JYTZD2023050].

ORCID

Ming Gao  <http://orcid.org/0000-0002-8502-1155>

References

- Allen-Zhu, Z., and Y. Li. 2020. "Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning." *arXiv preprint arXiv*. <https://doi.org/10.48550/arXiv.2012.09816>.
- An, J., L. Guo, W. Liu, Z. Fu, P. Ren, X. Liu, and T. Li. 2021. "IGAGCN: Information Geometry and Attention-Based Spatiotemporal Graph Convolutional Networks for Traffic Flow Prediction." *Neural Networks* 143:355–367. <https://doi.org/10.1016/j.neunet.2021.05.035>.
- Awad, M., R. Khanna, M. Awad, and R. Khanna. 2015. "Efficient Learning Machines: Theories." *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*: 67–80.
- Bai, L., L. Yao, C. Li, X. Wang, and C. Wang. 2020. "Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting." *Advances in Neural Information Processing Systems* 33:17804–17815.
- Chebatar, Y., and A. Waters. 2016. "Distilling Knowledge from Ensembles of Neural Networks for Speech Recognition." *Interspeech*: 3439–3443. <https://doi.org/10.21437/Interspeech.2016-1190>.
- Chen, Z. Y., and M. Sun. 2024. "Deep Stacking Kernel Machines for the Data-Driven Multi-Item, One-Warehouse, Multiretailer Problems with Backlog and Lost Sales." *INFORMS Journal on Computing, Ijoc*. <https://doi.org/10.1287/ijoc.2022.0365.cd>.
- Chiang, H. S., and Q. Y. Liu. 2025. "Traffic Incident Duration Prediction Based on Deep Learning Methods." *Enterprise Information Systems* 19 (3–4): 2448828. <https://doi.org/10.1080/17517575.2024.2448828>.
- Cho, J. H., and B. Hariharan. 2019. "On the Efficacy of Knowledge Distillation." Proceedings of the IEEE/CVF international conference on computer vision, Seoul, Republic of Korea, 4794–4802.
- Chu, Z., W. Ma, M. Li, and H. Chen. 2024. "Adaptive Decision Spatio-Temporal Neural ODE for Traffic Flow Forecasting with Multi-Kernel Temporal Dynamic Dilation Convolution." *Neural Networks* 179:106549. <https://doi.org/10.1016/j.neunet.2024.106549>.
- Desgourdes, C., and J. Ram. 2024. "The Role of Big Data Analytics for Decision-Making in Projects: Uses and Challenges." *Enterprise Information Systems* 18 (4): 2317153. <https://doi.org/10.1080/17517575.2024.2317153>.
- Fan, J., W. Weng, H. Tian, H. Wu, F. Zhu, and J. Wu. 2024. "Rgdn: A Random Graph Diffusion Attention Network for Traffic Prediction." *Neural Networks* 172:106093. <https://doi.org/10.1016/j.neunet.2023.106093>.
- Fukuda, T., M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran. 2017. "Efficient Knowledge Distillation from an Ensemble of Teachers." *Interspeech*: 3697–3701. <https://doi.org/10.21437/Interspeech.2017-614>.
- Gao, H., R. Jiang, Z. Dong, J. Deng, Y. Ma, and X. Song. 2024. Spatial-Temporal-Decoupled Masked Pre-training for Spatiotemporal Forecasting. *arXiv*:2312.00516.
- Gou, J., B. Yu, S. J. Maybank, and D. Tao. 2021. "Knowledge Distillation: A Survey." *International Journal of Computer Vision* 129 (6): 1789–1819. <https://doi.org/10.1007/s11263-021-01453-z>.
- Guo, S., Y. Lin, N. Feng, C. Song, and H. Wan. 2019. "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting." *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (1): 922–929. <https://doi.org/10.1609/aaai.v33i01.3301922>.
- Gupta, B. B., A. Gaurav, V. Arya, and P. K. Panigrahi. 2025. "Big Data Innovations in Enterprise Information Systems: Strategies Formation for New Generation Entrepreneurs." *Enterprise Information Systems* 19 (1–2): 2449261. <https://doi.org/10.1080/17517575.2024.2449261>.
- Ha, D., A. Dai, and Q. V. Le. 2016. *Hypernetworks*. <https://arxiv.org/abs/1609.09106>, *arXiv*:1609.09106.
- Hinton, G., O. Vinyals, and J. Dean. 2015. *Distilling the Knowledge in a Neural Network*. <https://arxiv.org/abs/1503.02531>, *arXiv*:1503.02531.
- Hong S.M. ZhugeJ. Chen X. ZhengY. ChengC. Zhang J. Wang et al. METAGPT: META PROGRAMMING FOR a MULTI-AGENT COLLABORATIVE FRAMEWORK. International Conference on Learning Representations (ICLR 2024).
- Jagadish, H. V., J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi. 2014. "Big Data and Its Technical Challenges." *Communications of the ACM* 57 (7): 86–94. <https://doi.org/10.1145/2611567>.
- Jiang, R., Z. Wang, J. Yong, P. Jeph, Q. Chen, Y. Kobayashi, X. Song, S. Fukushima, and T. Suzumura. 2023. Spatio-Temporal Meta-Graph Learning for Traffic Forecasting. *arXiv*:2211.14701.
- Jiang, W., J. Han, H. Liu, T. Tao, N. Tan, and H. Xiong. 2024 a. Interpretable Cascading Mixture-of-Experts for Urban Traffic Congestion Prediction. *arXiv*:2406.12923.

- Jiang, W., J. Han, H. Liu, T. Tao, N. Tan, and H. Xiong. 2024 b. "Interpretable Cascading Mixture-Of-Experts for Urban Traffic Congestion Prediction." *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA: 5206–5217.
- Jin, G., Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, and Y. Zheng. 2024. "Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey." *IEEE Transactions on Knowledge and Data Engineering* 36 (10): 388–5408. <https://doi.org/10.1109/TKDE.2023.3333824>.
- Jin, M., Y. Zheng, Y. F. Li, S. Chen, B. Yang, and S. Pan. 2022. "Multivariate Time Series Forecasting with Dynamic Graph Neural Odes." *IEEE Transactions on Knowledge and Data Engineering* 35 (9): 9168–9180. <https://doi.org/10.1109/TKDE.2022.3221989>.
- Kazemi, S. M., R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. Brubaker. 2019. Time2vec: Learning a Vector Representation of Time. arXiv preprint arXiv:1907.05321. <https://doi.org/10.48550/arXiv.1907.05321>.
- Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu. 2017. "Lightgbm: A Highly Efficient Gradient Boosting Decision Tree." *Advances in Neural Information Processing Systems* 30 : 3146–3154.
- Lan X., X. Zhu, S. Gong, et al. 2018. "Knowledge Distillation by on-The-Fly Native Ensemble." *Advances in Neural Information Processing Systems* 31 : 7517–7527.
- Lee H.S. JinH. Chu H. LimS. Ko 2021 Learning to Remember Patterns: Pattern Matching Memory Networks for Traffic Forecasting, International Conference on Learning Representations (ICLR 2022). <https://openreview.net/forum?id=wwDg3bbYBlq>.
- Lee, H., S. Jin, H. Chu, H. Lim, and S. Ko. 2022. "Learning to remember patterns: pattern matching memory networks for traffic forecasting."
- Lee, H., and S. Ko. 2024. "Testam: A time-enhanced spatio-temporal attention model with mixture of experts."
- Li, R., F. Zhang, T. Li, N. Zhang, and T. Zhang. 2023. "DMGAN: Dynamic Multi-Hop Graph Attention Network for Traffic Forecasting." *IEEE Transactions on Knowledge and Data Engineering* 35 (9): 9088–9101. <https://doi.org/10.1109/TKDE.2022.3221316>.
- Li, Y., R. Yu, C. Shahabi, and Y. Liu. 2018. *Diffusion convolutional recurrent neural network: data-driven traffic forecasting*.
- Lin, J., Q. Ren, X. Lv, H. Xu, and Y. Liu. 2025. "When Multi-View Meets Multi-Level: A Novel Spatio-Temporal Transformer for Traffic Prediction." *Information Fusion* 117:102801. <https://doi.org/10.1016/j.inffus.2024.102801>.
- Linares-Mustarós, S., J. C. Ferrer-Comalat, D. Corominas-Coll, and J. M. Merigó. 2019. "The Ordered Weighted Average in the Theory of Expertons." *International Journal of Intelligent Systems* 34 (3): 345–365. <https://doi.org/10.1002/int.22055>.
- Liu, S., M. He, Z. Wu, P. Lu, and W. Gu. 2024. "Spatial-Temporal Graph Neural Network Traffic Prediction Based Load Balancing with Reinforcement Learning in Cellular Networks." *Information Fusion* 103:102079. <https://doi.org/10.1016/j.inffus.2023.102079>.
- Liu, Y., T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long. 2024. iTransformer: Inverted Transformers are Effective for Time Series Forecasting. arXiv:2310.06625.
- Liu, Y., W. Zhang, and J. Wang. 2020. "Adaptive Multi-Teacher Multi-Level Knowledge Distillation." *Neurocomputing* 415:106–113. <https://doi.org/10.1016/j.neucom.2020.07.048>.
- Liu, Z. 2022. "Super Convergence Cosine Annealing with Warm-Up Learning Rate." CAIBDA 2022; 2nd International Conference on Artificial Intelligence, Big Data and Algorithms, VDE, Nanjing, China, 1–7.
- Luo, Y., J. Zheng, X. Wang, Y. Tao, and X. Jiang. 2024. "GT-LSTM: A Spatio-Temporal Ensemble Network for Traffic Flow Prediction." *Neural Networks* 171:251–262. <https://doi.org/10.1016/j.neunet.2023.12.016>.
- Murshed, M. G. S., C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain. 2021. "Machine Learning at the Network Edge: A Survey." *ACM Computing Surveys* 54 (8): 170. <https://doi.org/10.1145/3469029>.
- Pang, B., J. Gou, L. M. Camarinha-Matos, J. Z. Zhang, and L. Zhang. 2025. "Scenario-Oriented Data Interoperability: Maximising the Connection Between Data and Users in Collaboration Environments." *Enterprise Information Systems* 19 (1–2): 2445604. <https://doi.org/10.1080/17517575.2024.2445604>.
- Rigatti, S. J. 2017. "Random forest." *Journal of Insurance Medicine* 47 (1): 31–39. <https://doi.org/10.17849/insm-47-01-31-39.1>.
- Romero, A., N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. 2015. FitNets: Hints for Thin Deep Nets. arXiv:1412.6550.
- Seyedan, M., and F. Mafakheri. 2020. "Predictive Big Data Analytics for Supply Chain Demand Forecasting: Methods, Applications, and Research Opportunities." *Journal of Big Data* 7 (1): 53. <https://doi.org/10.1186/s40537-020-00329-2>.
- Shao, Y., Y. Zhao, F. Yu, H. Zhu, and J. Fang. 2021. "The Traffic Flow Prediction Method Using the Incremental Learning-Based Cnn-Ltsm Model: The Solution of Mobile Application." *Mobile Information Systems* 2021:5579451. <https://doi.org/10.1155/2021/5579451>.
- Shao, Z., Z. Wang, X. Yao, M. G. Bell, and J. Gao. 2024. "St-Mambasync: Complement the Power of Mamba and Transformer Fusion for Less Computational Cost in Spatial-Temporal Traffic Forecasting." *Information Fusion* 102872 102872. <https://doi.org/10.1016/j.inffus.2024.102872>.
- Shao, Z., Z. Zhang, F. Wang, and Y. Xu. 2022. "Pre-Training Enhanced Spatial-Temporal Graph Neural Network for Multivariate Time Series Forecasting." *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, Washington, DC, USA, 1567–1577.

- Sun, L., M. Liu, G. Liu, X. Chen, and X. Yu. 2024. "Fd-Tgcn: Fast and Dynamic Temporal Graph Convolution Network for Traffic Flow Prediction." *Information Fusion* 106:102291. <https://doi.org/10.1016/j.inffus.2024.102291>.
- Tang, X., H. Chen, Y. Shen, H. Wu, J. Li, and J. Li. 2024. "Adpstgcn: Adaptive Spatial-Temporal Graph Convolutional Network for Traffic Forecasting." *Knowledge-Based Systems* 301:112295. <https://doi.org/10.1016/j.knosys.2024.112295>.
- Tsantekidis, A., N. Passalis, and A. Tefas. 2021. "Diversity-Driven Knowledge Distillation for Financial Trading Using Deep Reinforcement Learning." *Neural Networks* 140:193–202. <https://doi.org/10.1016/j.neunet.2021.02.026>.
- Wang, G., Y. Chen, M. Gao, Z. Wu, J. Tang, and J. Zhao. 2024. A Time Series is Worth Five Experts: Heterogeneous Mixture of Experts for Traffic Flow Prediction. arXiv:2409.17440.
- Wasi, A. T., M. S. Islam, and A. R. Akib. 2024. "Supplygraph: A benchmark dataset for supply chain planning using graph neural networks." *4th Workshop on Graphs and More Complex Structures for Learning and Reasoning (GCLR), 38th AAAI Conference on Artificial Intelligence*. <https://doi.org/10.48550/arXiv.2401.15299>.
- Wasi, A. T., M. S. Islam, and A. R. Akib. 2025. Supplygraph: A benchmark dataset for supply chain planning using graph neural networks. <https://arxiv.org/abs/2401.15299>, arXiv:2401.15299.
- Wu, Z., S. Pan, G. Long, J. Jiang, and C. Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. arXiv:1906.00121.
- Xu, C., and C. Xu. 2024. "Local Spatial and Temporal Relation Discovery Model Based on Attention Mechanism for Traffic Forecasting." *Neural Networks* 176:106365. <https://doi.org/10.1016/j.neunet.2024.106365>.
- Xu, L. D. 2011. "Enterprise Systems: State-Of-The-Art and Future Trends." *IEEE Transactions on Industrial Informatics* 7 (4): 630–640. <https://doi.org/10.1109/TII.2011.2167156>.
- Xu, M., W. Dai, C. Liu, X. Gao, W. Lin, G. J. Qi, and H. Xiong. 2021. Spatial-Temporal Transformer Networks for Traffic Flow Forecasting. arXiv:2001.02908.
- Xu, Q., K. Wu, M. Wu, K. Mao, X. Li, and Z. Chen. 2024. "Reinforced Knowledge Distillation for Time Series Regression." *IEEE Transactions on Artificial Intelligence* 5 (6): 3184–3194. <https://doi.org/10.1109/TAI.2023.3341854>.
- Yan, Y., S. Cui, J. Liu, Y. Zhao, B. Zhou, and Y. H. Kuo. 2025. "Multimodal Fusion for Large-Scale Traffic Prediction with Heterogeneous Retentive Networks." *Information Fusion* 114:102695. <https://doi.org/10.1016/j.inffus.2024.102695>.
- Yang, D., S. Li, Z. Peng, P. Wang, J. Wang, and H. Yang. 2019. "Mf-Cnn: Traffic Flow Prediction Using Convolutional Neural Network and Multi-Features Fusion." *IEICE Transactions on Information and Systems* 102 (8): 1526–1536. <https://doi.org/10.1587/transinf.2018EDP7330>.
- Ye, J., L. Sun, B. Du, Y. Fu, and H. Xiong. 2021. "Coupled Layer-Wise Graph Convolution for Transportation Demand Prediction." *Proceedings of the AAAI conference on artificial intelligence, Virtual conference (online)*, 4617–4625.
- Yin, L., T. Li, G. Li, S. Besklubova, R. Y. Zhong, and S. Zhao. 2024. "Cyber-Physical Internet Based Intelligent Operation Platform for off-Site Construction Supply Chain Management." *Enterprise Information Systems* 18 (6): 2346743. <https://doi.org/10.1080/17517575.2024.2346743>.
- Yu, B., H. Yin, and Z. Zhu. 2018 b. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting." *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden*: 3634–3640.
- Yu, J., and G. Liu. 2021. "Extracting and Inserting Knowledge into Stacked Denoising Auto-Encoders." *Neural Networks* 137:31–42. <https://doi.org/10.1016/j.neunet.2021.01.010>.
- Zhang, W., Y. Yu, Y. Qi, F. Shu, and Y. Wang. 2019. "Short-Term Traffic Flow Prediction Based on Spatio-Temporal Analysis and Cnn Deep Learning." *Transportmetrica A: Transport Science* 15 (2): 1688–1711. <https://doi.org/10.1080/23249935.2019.1637966>.
- Zhang, Y., and J. Yan. 2023. "Crossformer: Transformer Utilizing Cross." *International Conference on Learning Representations (ICLR 2023)*.
- Zheng, C., X. Fan, C. Wang, and J. Qi. 2020. "GMAN: A Graph Multi-Attention Network for Traffic Prediction." *Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA*, 34, 1234–1241.
- Zoph, B., I. Bello, S. Kumar, N. Du, Y. Huang, J. Dean, N. Shazeer, and W. Fedus. 2022. *St-moe: Designing stable and transferable sparse expert models*. <https://arxiv.org/abs/2202.08906>, arXiv:2202.08906.