STAR^2: Spatio-Temporal Adaptive Retrieval and Refinement Transformer for Traffic Prediction

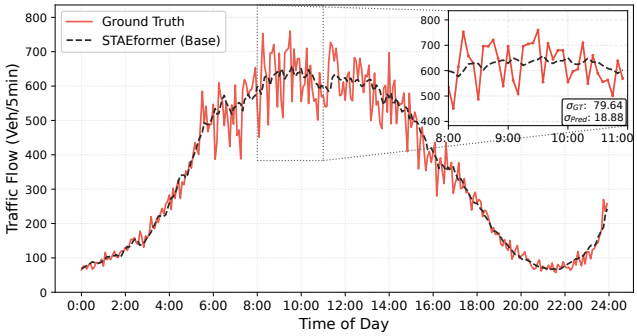
Xinyi Bao, and Yu Fang*
Tongji University, Shanghai, China
Email: XinyiBao@tongji.edu.cn, fangyu@tongji.edu*

***Abstract*—Accurate traffic flow forecasting is a cornerstone of Intelligent Transportation Systems (ITS). Although advanced parametric models have achieved significant progress, they encounter two critical bottlenecks: (i) optimization bias, where models prioritize routine patterns to minimize global error, rendering them inadequate in capturing sharp fluctuations; and (ii) a persistent long-tail error distribution, where a minority of "hard samples" contribute the bulk of the overall prediction error. To address these issues, we propose STAR^2, a novel Spatio-Temporal Adaptive Retrieval and Refinement framework. Our architecture integrates a coarse-to-fine two-stage retrieval strategy that reduces computational complexity from $O(N \cdot M)$ to $O(M+N \cdot K_g)$ to ensure real-time efficiency, an error estimator that quantifies backbone uncertainty via high-level spatio-temporal embeddings, and an adaptive refiner that dynamically fuses historical patterns with a frozen backbone to prevent blind refinement. Extensive evaluations on six real-world datasets indicate that STAR^2 yields outstanding predictive capabilities, notably for hard samples, while promoting higher computational efficiency to facilitate robust forecasting in complex road networks.**

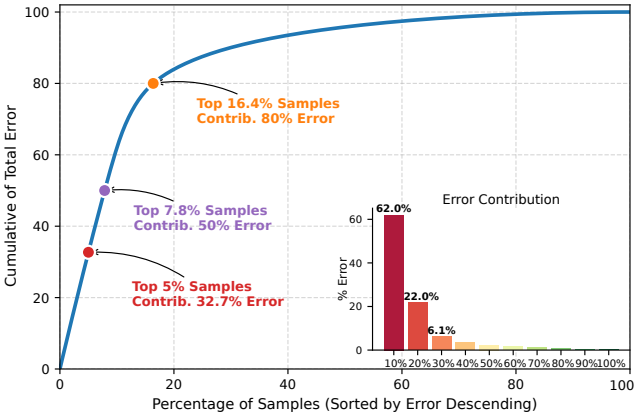
***Index Terms*—Traffic prediction, Spatio-temporal Transformer, Retrieval-augmented**

I. Introduction

Accurate traffic flow prediction is a pivotal task in Intelligent Transportation Systems (ITS). With the explosive growth of large-scale traffic data, data-driven approaches have gradually superseded traditional statistical models, such as ARIMA [1], becoming the dominant paradigm in this field. In the early stages of deep learning deployment, researchers primarily employed hybrid architectures to capture spatio-temporal correlations. A common practice involved utilizing Graph Neural Networks (GNNs) to model road network topology, combined with Recurrent Neural Networks (RNNs) or Temporal Convolutional Networks (TCNs) [2] to capture temporal dynamics. Representative works include DCRNN [3], which utilizes diffusion convolution to capture spatial dependencies, and Graph WaveNet [4], which introduces an adaptive adjacency matrix to learn latent spatial structures. Recently, Transformer-based architectures have established State-of-the-Art (SOTA) status, leveraging the powerful self-attention mechanism [5] to demonstrate superior performance in modeling long-range spatio-temporal dependencies. Models such as PDFormer [6] and STAEformer [7] have achieved remarkable prediction accuracy on mainstream benchmarks by designing sophisticated spatio-temporal embeddings and attention layers.



(a) Visualization of over-smoothing by the STAEformer backbone on typical long-tail difficult samples (PEMS04 node 209).



(b) Long-tail error distribution on METR-LA.

Fig. 1: Statistical and empirical analysis of prediction failures in traffic forecasting.

Despite the impressive performance of these Transformer-based models, they encounter non-trivial limitations in practical applications. First, these models are susceptible to **optimization bias**. Since training objectives (e.g., MAE or MSE) are typically calculated over the entire dataset, models prioritize fitting dominant, routine patterns to minimize global error [8]. Consequently, when encountering "hard samples" such as sudden congestion [9], accidents, or extreme weather conditions, the model predictions often exhibit over-smoothing [10], failing to capture drastic traffic fluctuations (as shown in Fig. 1(a)). Second, compressing historical information into fixed parameters creates a **memory bottleneck** [11]. This hinders the model's ability to generalize to rare, long-tail

patterns, as it lacks a mechanism to explicitly retrieve relevant historical contexts. Our empirical analysis on the METR-LA dataset, illustrated in Fig. 1(b), underscores the severity of this issue: in METR-LA, the top 7.8% of "hard samples" contribute to 50% of the total error. This indicates that these edge cases effectively bottleneck the further improvement of SOTA models.

To this end, we propose **STAR**^{2} (**S**patio-**T**emporal **A**daptive **R**etrieval and **R**efinement Transformer for Traffic Prediction), a novel retrieval-augmented framework designed to explicitly leverage historical data for correcting prediction bias. To address the computational challenges of retrieving from massive databases, we design a **coarse-to-fine two-stage retrieval strategy** that ensures real-time efficiency. Furthermore, we introduce an **uncertainty-aware adaptive refiner**, allowing the model to selectively "recall" precise historical results when encountering hard samples. By synergizing the generalization capabilities of deep learning with specific references from external memory, our approach achieves performance gains and effectively alleviates the bottlenecks faced by existing SOTA models.The code is available at https://github.com/lkkio/STAR2.

II. Related Work

A. Spatio-temporal Traffic Forecasting

Traffic forecasting, a cornerstone of Intelligent Transportation Systems (ITS), aims to predict future traffic states (e.g., speed, flow) based on historical observations. Existing methodologies have evolved from statistical models to deep learning-based architectures.

Early research primarily relied on statistical methods such as ARIMA [1] and its variants, VAR, and Kalman Filtering. These methods struggle to capture complex non-linear spatio-temporal dependencies. With the advent of deep learning, researchers began leveraging CNNs and RNNs for traffic data [3], [4], [9]. For instance, ST-ResNet [12] partitions cities into grids and employs CNNs to capture spatial correlations, while FC-LSTM [13] utilizes Long Short-Term Memory networks to handle temporal dependencies. However, these methods are often restricted to Euclidean space, failing to account for the complex non-Euclidean topology of road networks [14].

Recently, Transformer-based architectures have dominated the field due to their advantages in capturing long-range global dependencies. Initial efforts focused on dynamic spatio-temporal modeling; for instance, Traffic Transformer [15] first applied multi-head attention to traffic tasks, while GMAN [16] advanced this by designing dedicated spatial and temporal blocks. To better integrate underlying road topologies, ASTGNN [17] combined dynamic graph convolutions with attention mechanisms. Moving beyond general correlations, subsequent research began to incorporate intrinsic physical properties and data heterogeneity: PDFormer [6] introduced a delay-aware mechanism to account for spatial propagation lags, and ST-ExpertNet [18] employed a Mixture-of-Experts (MoE) architecture to handle distribution shifts between weekdays and holidays. As a representative state-of-the-art (SOTA) model, STAEformer [7] decouples complex spatio-temporal correlations by learning spatio-temporal adaptive embeddings, achieving superior accuracy with reduced complexity.

B. Retrieval-Augmented Methods

Retrieval-Augmented Generation (RAG) [19], [20] was originally developed to address hallucinations and memory bottlenecks [11] in Large Language Models (LLMs) by enhancing generation through external knowledge base retrieval. Inspired by this, the retrieval-augmented paradigm has been extended to time-series forecasting. Early explorations such as RATD [21] and RATF [22] attempted to retrieve similar raw time-series segments or latent features as reference information, validating the feasibility of utilizing historical data as "external memory." PIR [23] proposed an instance-aware refinement method that, unlike prefix-retrieval, focuses on identifying hard samples with large biases and retrieving similar historical sequences to correct the base model's output. In the specific domain of spatio-temporal forecasting, RAST [24] further extends retrieval to spatio-temporal data, representing the current state-of-the-art.

However, deploying these methods for large-scale real-time traffic forecasting faces significant obstacles. Current approaches often ignore spatial topology by treating nodes independently and rely on exhaustive search strategies with a prohibitive time complexity of $O(N \times T)$, rendering them unsuitable for real-time ITS applications [25]. Moreover, they typically employ a "blind fusion" strategy that indiscriminately augments all samples—potentially introducing noise into simple cases—and often necessitate computationally expensive joint end-to-end training.

III. Methodology

A. Problem Definition

In this study, we represent the road network as a graph $G = (V, E, A)$, where V is a set of N nodes and $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix. The traffic forecasting task aims to predict the future sequence $\mathbf{Y} \in \mathbb{R}^{T_{out} \times N \times C}$ given the historical observations $\mathbf{X} \in \mathbb{R}^{T_{in} \times N \times C}$ over T_{in} time steps, where C is the number of traffic features. To enhance the prediction, we construct an external memory $\mathcal{M} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^{M}$ containing M historical input-output pairs.

B. Framework Overview

As illustrated in Figure 2, our proposed STAR^2 framework follows a retrieve-and-refine paradigm, which consists of three main stages:

1) **Frozen Backbone Prediction**. The input sequence $\mathbf{X}$ is first processed by a frozen pre-trained backbone (i.e., STAEformer). This stage generates the initial backbone prediction $\hat{\mathbf{Y}}_{bb}$ and extracts the high-level spatio-temporal embedding $\mathbf{H}$, which serves as the context for error estimation.
2) **Two-stage Efficient Retrieval**. Simultaneously, we employ a coarse-to-fine two-stage retrieval strategy to

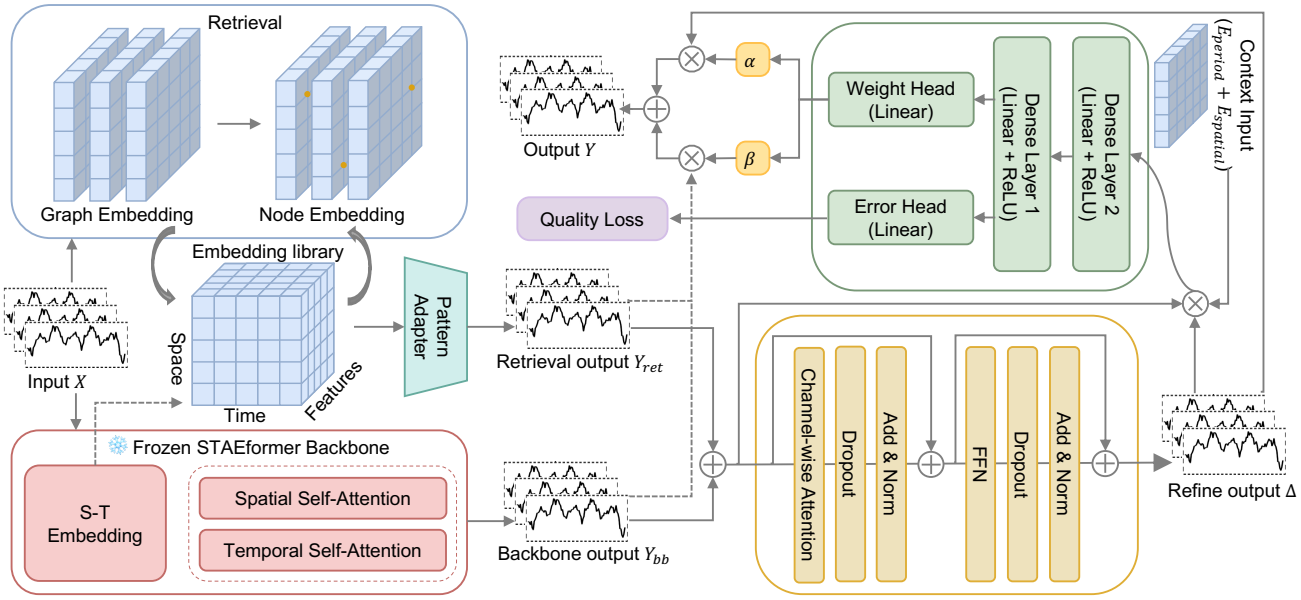


Fig. 2: Framework overview.

search the external memory $\mathcal{M}$. By performing hierarchical matching at both graph and node levels, the framework efficiently recalls the most relevant historical ground-truth patterns $\mathcal{Y}_{ret}$ as external references while maintaining real-time forecasting efficiency.

3) **Error-Guided Adaptive Refiner**. This stage begins by aligning $\mathcal{Y}_{ret}$ via a pattern adapter to eliminate scale gaps between historical data and the current state. Subsequently, an Adaptive Refiner (built on a Transformer architecture) captures the deep interactions between $\hat{\mathbf{Y}}_{bb}$ and the adapted retrieval patterns to produce a residual correction Δ. Meanwhile, an Error Estimator integrates $\mathbf{H}$, spatio-temporal contexts, and the output features from the Refiner to quantify backbone uncertainty. This process generates dynamic weights α and β, which are used in a three-way gated fusion mechanism to produce the final prediction $\hat{\mathbf{Y}}_{final}$ that specifically compensates for errors on hard samples.

C. Two-stage Retrieval

To enable efficient and accurate retrieval, we utilize a pre-trained and frozen STAEformer encoder to extract robust embeddings from historical traffic data and construct an external spatio-temporal memory database $\mathcal{M}$ that maps representative embeddings to historical ground-truth sequences. To facilitate the coarse-to-fine retrieval process, we design a hierarchical indexing scheme:

- Graph-level Keys ($\mathbf{K}_g$): For each historical sample ($T \times N \times D$), we first apply temporal mean pooling to capture the evolution trend, followed by spatial mean pooling to aggregate information across all nodes. This results in a global vector of shape ($1 \times D$) that represents the macroscopic traffic state of the entire city at a given time.
- Node-level Keys ($\mathbf{K}_n$): To account for spatial heterogeneity, we generate node-specific keys by performing only temporal mean pooling while preserving the spatial dimension. This yields a set of vectors of shape ($N \times D$), where each vector serves as a unique fingerprint for a specific road segment's local history.

A basic retrieval approach that calculates similarities between all query nodes and the entire historical database $\mathcal{M}$ would incur a prohibitive computational cost of $O(N \times M)$, making real-time forecasting infeasible for large-scale road networks. To address this efficiency bottleneck, we first perform a graph-level coarse retrieval to narrow down the search space. Specifically, the query sequence $\mathbf{X}_{query}$ is transformed into a query vector $q_g \in \mathbb{R}^D$ via the frozen encoder followed by dual-level mean pooling. We then identify the most relevant historical time slices by computing the cosine similarity between q_g and the pre-computed graph-level keys $\mathbf{K}_g$:

$$S_g = \text{Sim}(q_g, \mathbf{K}_g)$$

where the similarity is efficiently calculated via the inner product of L2-normalized vectors using FAISS [26]. By selecting the top-K_g indices ($K_g \ll M$), this stage effectively filters out irrelevant historical data and reduces the subsequent search space to a highly concentrated subset of global analogs.

Then, we perform a node-level fine-grained matching within the K_g candidate time slices retrieved in the previous stage. For each individual node $v_i \in V$, we compute the similarity between its specific query feature $q_{n,i}$ and the corresponding node-level keys $k_{n,i}$ stored in the candidate subset:

$$S_{n,i} = \text{Sim}(q_{n,i}, k_{n,i})$$

The process yields a customized reference matrix $\mathcal{Y}_{ret} \in \mathbb{R}^{T_{out} \times N \times C}$, which serves as the non-parametric basis for the subsequent adaptive correction module.

This hierarchical strategy shifts the total retrieval complexity from a product-based $O(N \cdot M)$ to a summation-based $O(M + N \cdot K_g)$, ensuring real-time inference efficiency for large-scale networks.

D. Error-Guided Adaptive Refiner

To mitigate the risk of introducing retrieval noise into simple samples, we propose an Error-Guided Adaptive Refiner. This module enables instance-aware, on-demand correction by balancing backbone's output with retrieved historical patterns. The core of adaptive refinement lies in accurately identifying "hard samples" where the backbone is likely to fail.

First, the retrieved pattern $\mathcal{Y}_{ret}$ is aligned via a linear pattern adapter to eliminate distribution gaps. Subsequently, a Transformer-based refiner captures the deep spatio-temporal interactions between $\hat{\mathbf{Y}}_{bb}$ and the adapted retrieval patterns $\hat{\mathbf{Y}}_{ret}^{adapt}$ to produce a residual correction Δ. To quantify forecasting uncertainty, we employ a hierarchical error estimator. It first derives an error estimation by fusing the high-level spatio-temporal embedding $\mathbf{H}$ with the output features from the refiner through an Error Head. This estimation is then fed into a Weight Head to be mapped into two node-wise gating scores, α and β, via sigmoid activations:

$$\alpha, \beta = \sigma(\text{WeightHead}(\text{ErrorHead}([\mathbf{H}; \text{Refiner}(\cdot)])))$$

where $\alpha \in [0, 1]$ controls the intensity of the learned residual correction Δ, while $\beta \in [0, 1]$ determines the reliance on external historical references.

Based on these estimated uncertainties, the final prediction $\hat{\mathbf{Y}}_{final}$ is formulated via a three-way gated fusion mechanism:

$$\hat{\mathbf{Y}}_{final} = (1 - \beta) \cdot \hat{\mathbf{Y}}_{bb} + \beta \cdot \hat{\mathbf{Y}}_{ret}^{adapt} + \alpha \cdot \Delta$$

This mechanism implements a multi-level dynamic trade-off: for "easy" samples where the backbone exhibits low uncertainty, the framework preserves the original output to maintain parametric stability. For "hard" samples such as sudden congestion or long-tail events, the refiner adaptively shifts the focus toward the retrieved historical analogs via β while performing precise compensation via α and Δ, effectively correcting the backbone's bias using non-parametric memory.

E. Optimization Objective

We employ a multi-task objective to jointly optimize forecasting accuracy and error estimation. The primary forecasting loss $\mathcal{L}_{pred}$ measures the Mean Absolute Error (MAE) between the final refined prediction $\hat{\mathbf{Y}}_{final}$ and the ground truth $\mathbf{Y}_{gt}$:

$$\mathcal{L}_{pred} = \frac{1}{T_{out} \times N} \sum_{t,n} |\hat{y}_{t,n}^{final} - y_{t,n}^{gt}|$$

To supervise the estimator, we define the Quality Loss $\mathcal{L}_{qual}$. Specifically, we define the estimated error $\hat{\mathbf{R}}$ (derived from the Error Head) and construct a pseudo-label $\mathbf{R} = |\hat{\mathbf{Y}}_{bb} - \mathbf{Y}_{gt}|$ representing the frozen backbone's empirical error. The estimator is trained to approximate this residual via Mean Squared Error (MSE):

$$\mathcal{L}_{qual} = \text{MSE}(\hat{\mathbf{R}}, \text{SG}(\mathbf{R}))$$

where $\text{SG}(\cdot)$ denotes the stop-gradient operation to ensure the backbone's error distribution fixed. The total objective is formulated as $\mathcal{L} = \mathcal{L}_{pred} + \lambda \mathcal{L}_{qual}$, where λ is a hyperparameter balancing the two tasks.

Notably, we adopt a parameter-efficient strategy by freezing the pre-trained backbone. Gradients are back-propagated solely through the pattern adapter, refiner, and error estimator. This preserves the backbone's robust representations while focusing optimization on the adaptive correction mechanism.

IV. Experiments

A. Experimental Setup

Datasets. Our model is validated on six real-world traffic benchmarks: METR-LA, PEMS-BAY (from DCRNN [3]), and PEMS03, 04, 07, 08 (from STSGCN [27]). All datasets feature 5-minute sampling intervals (12 frames/hour). Detailed statistics, including sensor counts and time ranges, are presented in Table I.

TABLE I: Summary of Datasets.

Dataset	#Sensors (N)	#Timesteps	Time Range
METR-LA	207	34,272	03/2012 - 06/2012
PEMS-BAY	325	52,116	01/2017 - 05/2017
PEMS03	358	26,209	05/2012 - 07/2012
PEMS04	307	16,992	01/2018 - 02/2018
PEMS07	883	28,224	05/2017 - 08/2017
PEMS08	170	17,856	07/2016 - 08/2016

Baselines. We evaluate our proposed method against several state-of-the-art baselines representing different architectural approaches. These include representative Spatio-Temporal Graph Neural Networks (STGNNs) such as DCRNN [3], STGCN [9], AGCRN [28], GWNet [4], GTS [29], and MTGNN [30]. We also evaluate specialized Transformers (GMAN [16], PDFormer [6], and STAEformer [7]), which leverage attention mechanisms and adaptive embeddings to model spatio-temporal correlations. Finally, we include the efficient and robust baseline STID [31] for comprehensive comparison.

Metrics. We utilize MAE, RMSE, and MAPE to measure forecasting accuracy. Consistent with prior studies, we calculate the average error across all 12 predicted horizons for PEMS03, 04, 07, and 08. For METR-LA and PEMS-BAY, we report results at horizons 3, 6, and 12 (i.e., 15, 30, and 60 minutes) to facilitate comparison with baseline models.

Implementation. Our model is implemented in PyTorch on an NVIDIA RTX A5000 GPU, with dataset split ratios of 7:1:2 for METR-LA/PEMS-BAY and 6:2:2 for PEMS03/04/07/08. We set both input and prediction lengths to 12 time steps. Training is conducted for 100 epochs using the Adam optimizer with a learning rate of 0.001, weight decay of 1×10^{-5}, and a batch size of 16, while saving the best model state based on validation performance. The total loss integrates Masked MAE with a Quality Loss weighted at 1.0. The architecture utilizes a 2-layer refiner ($d_{model} = 64, d_{ff} = 128$), and the retrieval mechanism is configured with $k = 5$, $M = 100$, and a 0.1 dropout rate.

TABLE II: Performance on PEMS03, 04, 07 and 08.

Model	PEMS03			PEMS04			PEMS07			PEMS08		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
DCRNN	15.54	27.18	15.62%	19.63	31.26	13.59%	21.16	34.14	9.02%	15.22	24.17	10.21%
STGCN	15.83	27.51	16.13%	19.57	31.38	13.44%	21.74	35.27	9.24%	16.08	25.39	10.60%
AGCRN	15.24	26.65	15.89%	19.38	31.25	13.40%	20.57	34.40	8.74%	15.32	24.41	10.03%
GTS	15.41	26.15	15.39%	20.96	32.95	14.66%	22.15	35.10	9.38%	16.49	26.08	10.54%
GWNet	**14.59**	25.24	15.52%	18.53	29.92	12.89%	20.47	33.47	8.61%	14.40	23.39	9.21%
MTGNN	14.85	**25.23**	**14.55%**	19.17	31.70	13.37%	20.89	34.06	9.00%	15.18	24.24	10.20%
GMAN	16.87	27.92	18.23%	19.14	31.60	13.19%	20.97	34.10	9.05%	15.31	24.92	10.13%
PDFormer	14.94	25.39	15.82%	18.36	30.03	12.00%	19.97	32.95	8.55%	13.58	23.41	9.05%
STAEformer	15.35	27.55	15.18%	18.22	30.18	11.98%	19.14	32.60	8.01%	**13.46**	23.25	8.88%
STID	15.33	27.40	16.40%	18.38	29.95	12.04%	19.61	32.79	8.30%	14.21	23.28	9.27%
STAR^2 (Ours)	14.81	25.66	15.01%	**17.96**	**29.83**	**11.95%**	**19.09**	**32.45**	**7.96%**	**13.46**	**23.14**	**8.86%**

TABLE III: Performance on METR-LA and PEMS-BAY.

METR-LA	Horizon 3 (15 min)			Horizon 6 (30 min)			Horizon 12 (60 min)		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DCRNN	2.67	5.16	6.86%	3.12	6.27	8.42%	3.54	7.47	10.32%
STGCN	2.75	5.29	7.10%	3.15	6.35	8.62%	3.60	7.43	10.35%
AGCRN	2.85	5.53	7.63%	3.20	6.52	9.00%	3.59	7.45	10.47%
GTS	2.75	5.27	7.12%	3.14	6.33	8.62%	3.59	7.44	10.25%
GWNet	2.69	5.15	6.99%	3.08	6.20	8.47%	3.51	7.28	9.96%
MTGNN	2.69	5.16	6.89%	3.05	6.13	8.16%	3.47	7.21	9.70%
GMAN	2.80	5.55	7.41%	3.12	6.49	8.73%	3.44	7.35	10.07%
PDFormer	2.83	5.45	7.77%	3.20	6.46	9.19%	3.62	7.47	10.91%
STAEformer	2.65	5.11	6.85%	2.97	6.00	8.13%	3.34	7.02	9.70%
STID	2.82	5.53	7.75%	3.19	6.57	9.39%	3.55	7.55	10.95%
STAR^2	**2.63**	**5.06**	**6.82%**	**2.94**	**5.96**	**8.11%**	**3.33**	**6.99**	**9.70%**

PEMS-BAY	Horizon 3 (15 min)			Horizon 6 (30 min)			Horizon 12 (60 min)		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DCRNN	1.31	2.76	2.73%	1.65	3.75	3.71%	1.97	4.60	4.68%
STGCN	1.36	2.88	2.86%	1.70	3.84	3.79%	2.02	4.63	4.72%
AGCRN	1.35	2.88	2.91%	1.67	3.82	3.81%	1.94	4.50	4.55%
GTS	1.37	2.92	2.85%	1.72	3.86	3.88%	2.06	4.60	4.88%
GWNet	**1.30**	**2.73**	**2.71%**	1.63	3.73	3.73%	1.99	4.60	4.71%
MTGNN	1.33	2.80	2.81%	1.66	3.77	3.75%	1.95	4.50	4.62%
GMAN	1.35	2.90	2.87%	1.65	3.82	3.74%	1.92	4.49	4.52%
PDFormer	1.32	2.83	2.78%	1.64	3.79	3.71%	1.91	4.43	4.51%
STAEformer	1.31	2.78	2.76%	1.62	3.68	3.62%	1.88	4.34	4.41%
STID	1.31	2.79	2.78%	1.64	3.73	3.73%	1.91	4.42	4.55%
STAR^2	**1.30**	2.76	2.72%	**1.60**	**3.66**	**3.56%**	**1.87**	**4.30**	**4.36%**

B. Performance Evaluation

The comparative results are presented in Table II and Table III, where **bold** and underline indicate the best and second-best performance, respectively. As observed, STAR^2 achieves state-of-the-art performance across the majority of benchmarks. Specifically, it establishes new records on PEMS04, 07, and 08 while maintaining top-tier competitiveness on PEMS03. On the METR-LA and PEMS-BAY datasets, our model consistently outperforms established baselines such as STAEformer and PDFormer, with particularly significant gains in long-term forecasting horizons. These results underscore the model's superior capability in capturing stable spatio-temporal dependencies and effectively alleviating prediction bias through the proposed retrieval-augmented mechanism.

C. Ablation Study

To evaluate the effectiveness of each component, we compare three model variants as follows:

- w/o ErrorHead: Replaces the error estimation branch with fixed fusion weights.
- w/o Refiner: Discards the Transformer-based spatio-temporal refiner.

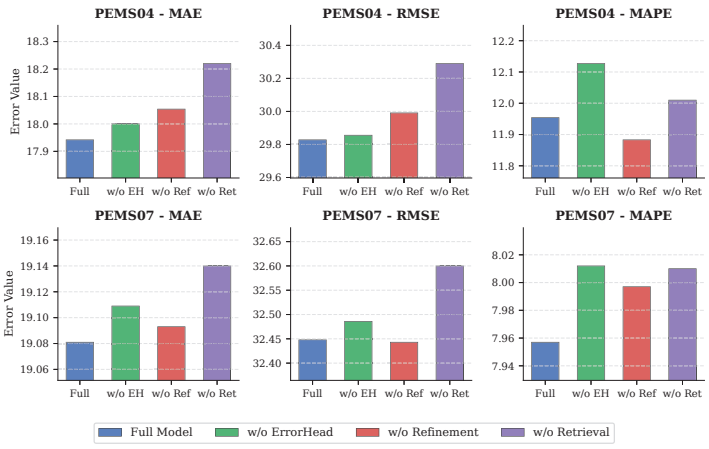


Fig. 3: Ablation Study on PEMS04 and PEMS07.

- w/o Retrieval: Removes the retrieval module, relying solely on frozen backbone outputs.

The ablation results presented in Fig. 3 highlight the critical impact of each module on the overall performance.Notably, removing the retrieval module (*w/o Retrieval*) leads to the most significant degradation, with MAE and RMSE increasing by approximately 1.5% on the PEMS04 dataset. This confirms that retrieved historical patterns provide essential external knowledge that the fixed parametric backbone fails to internalize. The accuracy gap in *w/o Refiner* underscores the module's role in maintaining spatial consistency during pattern integration. The higher error volatility observed in *w/o ErrorHead* validates our dynamic gated mechanism, proving that "on-demand" correction suppresses noise more effectively than static fusion by mitigating the risk of blind refinement.

D. Case Study

To evaluate the predictive performance in specific scenarios, we visualize the results for a typical hard sample (Node 22) from the PEMS04 dataset. As shown in Fig. 4, the frozen backbone (STAEformer) exhibits significant under-prediction when traffic flow increases sharply, failing to capture the sudden surge. By contrast, our model successfully captures these trends by integrating retrieved historical patterns that align closely with the ground truth. Quantitatively, for this difficult case, our method reduces the MAE from 108.8 to 23.6, achieving an absolute error reduction of 85.2.

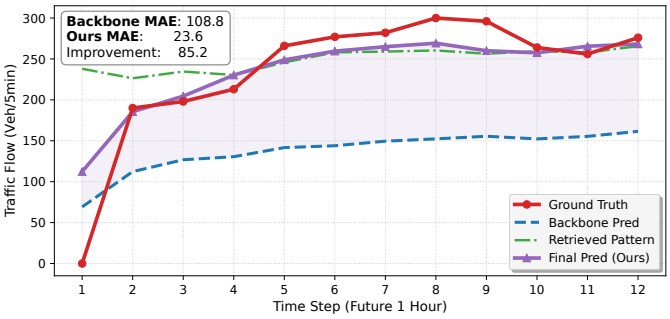


Fig. 4: Case Study on PEMS04 (Node 22) illustrating how the retrieval mechanism corrects the backbone's under-prediction on a hard sample.

V. Conclusion

In this study, we present STAR^2, a novel framework that enhances traffic forecasting accuracy through a two-stage adaptive retrieval and refinement mechanism. By integrating a frozen parametric backbone with a non-parametric memory bank, STAR^2 significantly optimizes the predictive performance on hard samples. Our experimental results validate that the uncertainty-aware adaptive fusion effectively mitigates the risk of blind refinement.

Future research will integrate multi-modal data, such as weather and social events, for richer context, while exploring cross-city generalizability and transfer learning to evaluate framework robustness across diverse urban topologies.

References

[1] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[2] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018. [Online]. Available: https://arxiv.org/abs/1803.01271

[3] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[4] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 4, 2023, pp. 4365–4373.

[7] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, "Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 4125–4129.

[8] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "St-norm: Spatial and temporal normalization for multi-variate time series forecasting," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 269–278.

[9] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[10] A. Khaled, A. M. T. Elsir, and Y. Shen, "Tfgan: Traffic forecasting using generative adversarial network with multi-graph convolutional network," *Knowledge-Based Systems*, vol. 249, p. 108990, 2022.

[11] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," *Advances in neural information processing systems*, vol. 36, pp. 43 322–43 355, 2023.

[12] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[13] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM international Conference on Data Mining*. SIAM, 2017, pp. 777–785.

[14] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[15] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, 2020.

[16] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[17] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, 2021.

[18] H. Wang, J. Chen, Z. Fan, Z. Zhang, Z. Cai, and X. Song, "St-expertnet: A deep expert framework for traffic prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 7512–7525, 2022.

[19] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[20] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-augmented generation for ai-generated content: A survey," *Data Science and Engineering*, pp. 1–29, 2026.

[21] S. Han, S. Lee, M. Cha, S. O. Arik, and J. Yoon, "Retrieval augmented time series forecasting," *arXiv preprint arXiv:2505.04163*, 2025.

[22] J. Liu, L. Yang, H. Li, and S. Hong, "Retrieval-augmented diffusion models for time series forecasting," *Advances in Neural Information Processing Systems*, vol. 37, pp. 2766–2786, 2024.

[23] Z. Liu, M. Cheng, G. Zhao, J. Yang, Q. Liu, and E. Chen, "Improving time series forecasting via instance-aware post-hoc revision," *arXiv preprint arXiv:2505.23583*, 2025.

[24] W. Ruan, X. Dang, Z. Zhou, S. Lyu, and Y. Liang, "Rast: A retrieval augmented spatio-temporal framework for traffic prediction," *arXiv preprint arXiv:2508.16623*, 2025.

[25] R. Tian, H. Zhai, W. Zhang, F. Wang, and Y. Guan, "A survey of spatio-temporal big data indexing methods in distributed environment," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 4132–4155, 2022.

[26] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[27] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 914–921.

[28] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 804–17 815, 2020.

[29] C. Shang, J. Chen, and J. Bi, "Discrete graph structure learning for forecasting multiple time series," *arXiv preprint arXiv:2101.06861*, 2021.

[30] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.

[31] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *Proceedings of the 31st ACM international conference on information & knowledge management*, 2022, pp. 4454–4458.