SDF-SP: SEMANTIC DUAL-ORDER SPATIOTEMPORAL FORECASTING WITH SIMPLICIAL PROJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatiotemporal forecasting must capture temporal dynamics and cross-node interactions under changing contextual conditions. Contextual semantics and historical trajectories provide complementary information, but incorporating node-specific contextual effects and combining divergent historical responses remain challenging. We propose SDF-SP, a framework that integrates semantically conditioned dual-order modelling with retrieval-based forecast correction. Its Dual-Order Spatiotemporal Encoder processes temporal-first and spatial-first branches in parallel, using context–node semantic relevance to modulate temporal features and spatial propagation. A context-conditioned gate fuses the branch representations, which a frozen pretrained language-model backbone uses to generate a base forecast. Retrieval-Augmented Simplicial Residual Projection then refines this forecast using retrieved historical trajectories. Guided by the base forecast, global and node-level projections model joint cross-node responses and individual node trajectories, respectively. Local simplicial constraints restrict interpolation to connected historical responses, and the projected forecasts are adaptively combined according to their deviations from the base forecast. Experiments on six real-world datasets spanning tourism, transportation and energy demonstrate strong forecasting performance across these domains. The code is publicly available at https://anonymous.4open.science/r/SDF-SP/.

1 INTRODUCTION

Spatiotemporal forecasting predicts future observations at multiple related nodes from historical measurements and available contextual information. It requires modelling node-level temporal dynamics and cross-node spatial interactions. Spatiotemporal graph models combine graph propagation with recurrence or temporal convolution (Li et al., 2018; Yu et al., 2018; Wu et al., 2019). Attention and mixture-of-experts architectures further adapt spatial interactions to varying input conditions (Lee & Ko, 2024). Text-derived representations can also complement numerical observations by encoding contextual information (Yu et al., 2025; Liu et al., 2025a).

Two modelling questions arise when contextual information and historical evidence are used for forecasting. First, how can contextual semantics guide node-level temporal modelling and cross-node spatial information exchange (Figure 1(a))? Feature fusion and cross-modal alignment connect textual representations with numerical sequences (Yu et al., 2025; Liu et al., 2025a). However, directly conditioning temporal dynamics and spatial propagation on node-dependent semantic relevance remains a distinct modelling challenge. Second, how can divergent retrieved responses be combined to correct a base forecast (Figure 1(b))? Retrieval-augmented methods access similar historical segments and their observed continuations, combining this evidence through reference construction, similarity weighting, or learned fusion (Han et al., 2025; Ning et al., 2025; Liu et al., 2025d; Zhou et al., 2026). Similar histories can lead to different future responses. In Figure 1(b), retrieved responses suggest different correction directions, which may also differ in cross-node coordination. Combining these responses requires accounting for their local relationships and compatibility with the base forecast.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these questions, we propose SDF-SP, a spatiotemporal forecasting framework that combines a semantically conditioned Dual-Order Spatiotemporal Encoder with Retrieval-Augmented Simplicial Residual Projection. First, the encoder uses context–node semantic relevance to modulate temporal features and spatial edge weights in temporal-first and spatial-first branches, respectively. A context-conditioned gate fuses the branch representations. The fused representation updates the historical sequence, which is tokenized for a frozen pretrained language-model backbone and prediction heads. Second, the projection module uses local relationships among retrieved residual responses to model both joint cross-node responses and individual node trajectories. Cases retrieved using time-series features and known future calendar features form local simplicial supports for weighted global and node-level projection. The final forecast adaptively combines the two projected forecasts according to their node-wise deviations from the base forecast, with the parametric forecaster kept fixed.

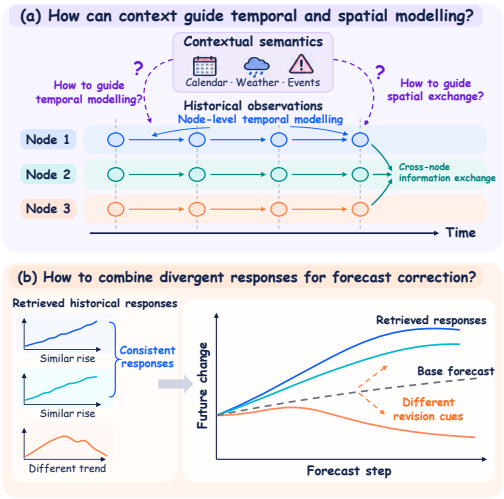

Figure 1: Forecasting motivation. (a) Semantic conditioning of temporal and spatial modelling. (b) Retrieved responses provide different revision cues for the base forecast.

The main contributions of this work are as follows:

- We propose a semantically conditioned Dual-Order Spatiotemporal Encoder that processes temporal-first and spatial-first branches in parallel. Order-specific semantic modulation and context-conditioned fusion capture complementary temporal dynamics and cross-node interactions under varying contexts.
- We propose Retrieval-Augmented Simplicial Residual Projection to correct base forecasts using historical trajectories. Base-guided global and node-level projections combine joint cross-node responses with individual node trajectories. Local simplicial constraints restrict interpolation to related historical responses, while the base forecast guides the choice of response combinations.
- Experiments on six real-world datasets spanning multiple domains demonstrate the effectiveness of SDF-SP and its superior overall forecasting accuracy compared with existing methods. Ablation studies further support the contributions of dual-order encoding and retrieval-based correction.

2 Related Work

2.1 Spatiotemporal Forecasting

Spatiotemporal forecasting on graphs jointly models temporal dynamics and cross-node dependencies. Early models couple graph propagation with recurrent or convolutional temporal modelling (Li et al., 2018; Yu et al., 2018), while adaptive graph methods learn relations beyond predefined connectivity and across time scales (Wu et al., 2019; Cai et al., 2024). Recent methods extend this foundation to richer interactions. MTGODE unifies continuous temporal and spatial propagation (Jin et al., 2023), PDFormer uses propagation-delay-aware attention (Jiang et al., 2023), while TESTAM routes inputs among temporal, static-graph, and dynamic-graph experts (Lee & Ko, 2024). ASTGI constructs observation-level causal neighbourhoods and weights messages by relative spatiotemporal positions (Liu et al., 2026). STBP combines lightweight graph attention with a contextual pattern bank (Liu & Zhang, 2026), while ST-LINK embeds spatial correlations in rotary position embeddings (Jeon et al., 2025). Together, these methods move from fixed topology toward adaptive interactions and changing contexts.

Beyond structural adaptation, semantic-aware methods complement numerical observations with contextual representations. FUSE-Traffic fuses text-derived context and graph features through cross-modal attention (Yu et al., 2025). Language-model-based methods connect numerical sequences to pretrained language representations through text-prototype reprogramming and cross-modal alignment (Jin et al., 2024; Liu et al., 2025a). ExoTimer combines exogenous-variable encoding with textual prompts and cross-modal alignment (Wu et al., 2026), while VoT integrates text-driven reasoning with representation alignment and prediction-level fusion (Wang et al., 2026). These approaches primarily integrate semantic and numerical representations or use text-based reasoning to guide forecasts. SDF-SP instead uses context–node semantic relevance to modulate temporal dynamics and spatial propagation within the temporal-first and spatial-first branches.

2.2 Retrieval-Augmented Forecasting

Retrieval augmentation supplements forecasts with instance-specific historical evidence at inference time. One line augments forecasting inputs or representations. RAF concatenates aligned historical examples with the query context (Tire et al., 2026), RATD guides diffusion denoising with retrieved reference features (Liu et al., 2024a), and TS-RAG models query–candidate interactions through attention-based fusion (Ning et al., 2025). TimeRAG incorporates retrieved sequences into textual prompts (Yang et al., 2025). TimeRAF uses channel prompting to integrate retrieved representations with a frozen forecasting backbone (Zhang et al., 2025), while VoT uses corrected historical reasoning as in-context guidance (Wang et al., 2026). In a complementary direction, GTR retrieves segments from a learned global periodic embedding (Cao et al., 2026).

Another line constructs prediction references or revises model outputs. RAFT retrieves similar histories and uses their observed continuations as a prediction reference (Han et al., 2025), while PIR identifies unreliable forecasts and revises them using local covariates and similarity-weighted historical continuations (Liu et al., 2025d). PFRP adaptively fuses forecasts constructed from retrieved historical continuations with local model outputs (Du et al., 2026). SARAF addresses non-stationarity through stationarity-aware diversity selection and aggregation of retrieved futures (Zhou et al., 2026). Similarity-weighted aggregation determines response combinations primarily from historical relevance. SDF-SP corrects the base forecast through base-guided local projections of historical trajectories at the cross-node and individual-node levels.

3 Methodology

3.1 Problem Formulation

We consider forecasting node observations on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the node and edge sets, respectively. The graph has $N = |\mathcal{V}|$ nodes, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ encodes the given relations among them. At forecast origin t, the historical observations are $\mathbf{X}_t = [\mathbf{x}_{t-T+1}, \ldots, \mathbf{x}_t] \in \mathbb{R}^{N \times T}$, with look-back length T. Here, $\mathbf{x}_t \in \mathbb{R}^N$ contains the observations at all nodes at time t. Let $\mathbf{P} \in \mathbb{R}^{N \times d_p}$ denote node semantic representations derived from node descriptions. The context $\mathcal{C}_t$ includes text-derived semantics and known forecast-horizon covariates available at the forecast origin. The goal is to predict the next H observations at every node:

$$\widehat{\mathbf{Y}}_t = \mathcal{F}(\mathbf{X}_t, \mathcal{G}, \mathbf{P}, \mathcal{C}_t), \qquad \mathbf{Y}_t = [\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+H}] \in \mathbb{R}^{N \times H}. \tag{1}$$

Here, $\mathcal{F}$ denotes the forecasting framework, while $\widehat{\mathbf{Y}}_t$ denotes its multi-step prediction. All elements of $\mathcal{C}_t$ are assumed to be available at forecast time.

3.2 Overview

As shown in Figure 2, SDF-SP produces a semantically conditioned base forecast and then refines it using retrieved historical responses. The Semantic-Context Modulator derives temporal and spatial modulation signals from context–node semantic relevance. The Dual-Order Spatiotemporal Encoder processes temporal-first (TS) and spatial-first (ST) branches in parallel. These signals modulate temporal features in the TS branch and spatial edge weights in the ST branch. A context-conditioned gate fuses the outputs of the two branches. The fused representation provides a residual update to the historical sequence before temporal and spatial tokenization. A frozen pretrained language-model backbone processes the resulting tokens, and prediction heads generate the base forecast.

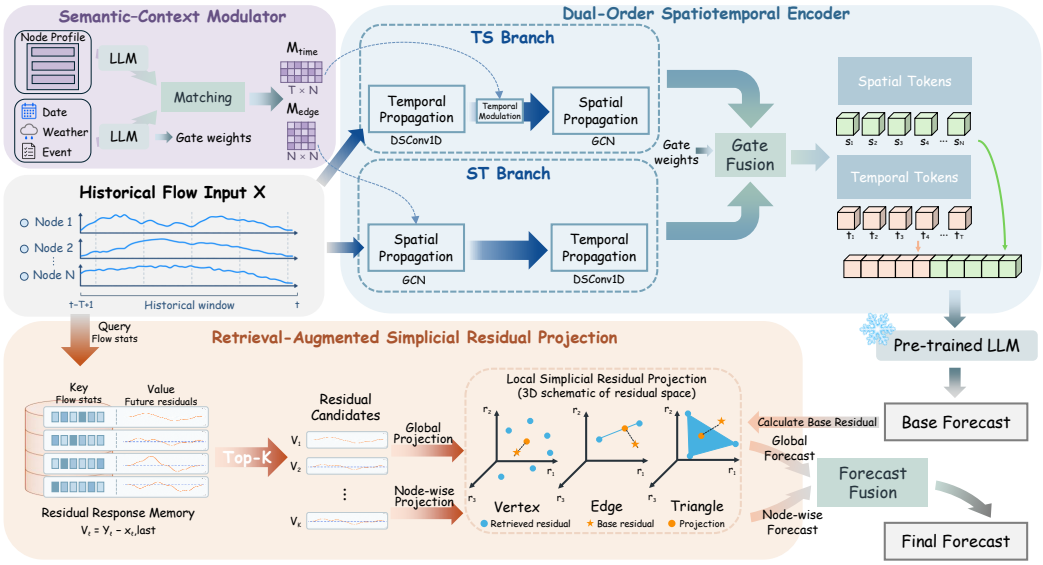

Figure 2: Overview of SDF-SP. Semantic conditioning guides dual-order encoding, and a frozen language-model backbone produces the base forecast. Retrieved responses support global and node-level simplicial projections, whose forecasts are adaptively combined. The sketches illustrate projection onto vertices, edges and triangles.

After base-model training, Retrieval-Augmented Simplicial Residual Projection refines the predictions of the fixed base forecaster. It retrieves historical residual responses from a memory constructed exclusively from training data. The base forecast is expressed as a residual relative to the last observed state. The Global Projection path in Figure 2 performs weighted projection of joint responses after node-scale adjustment, while Node-wise Projection models each node's trajectory separately. Both paths use the same retrieved cases but construct local simplicial supports in their own response spaces. Adding back the last observed state gives the Global Forecast and Node-wise Forecast shown in Figure 2. Forecast Fusion combines these two forecasts into the final prediction.

3.3 Semantic-Context Modulator

The Semantic-Context Modulator conditions temporal features and spatial edge weights through separate context–node matching signals. A pretrained language model extracts and organises prediction-relevant information from contextual and node-related texts. A text embedding model encodes the resulting information into semantic representations.

A context encoder maps the context embedding to a latent representation $\mathbf{Z}$. A temporal query and two endpoint queries derived from $\mathbf{Z}$ are matched with shared node semantic keys. The temporal query yields a node-specific semantic relevance score ψ_i, while the endpoint queries yield scores $m_i^{(1)}$ and $m_i^{(2)}$ for each node. In parallel, a separate projection maps $\mathbf{Z}$ to a context-dependent temporal modulation profile $\mathbf{U} \in \mathbb{R}^T$ over the historical window. The temporal and spatial modulation signals are

$$\begin{aligned} [\mathbf{M}_{\text{time}}]_{\tau,i} &= \tanh(U_\tau + \psi_i)\,, \\ [\mathbf{M}_{\text{edge}}]_{ij} &= B_{ij}\tanh\Big(m_i^{(1)} + m_j^{(2)}\Big)\,. \end{aligned} \tag{2}$$

Here, $\tau \in \{1, \ldots, T\}$ indexes the position of an observation within the historical window. Both U_τ and ψ_i depend on the current context; U_τ is shared across nodes, whereas ψ_i is shared across historical positions. The fixed similarity B_{ij} is computed as the cosine between the original node semantic embeddings, with $B_{ii} = 0$.

In the TS branch, $1 + [\mathbf{M}_{\text{time}}]_{\tau,i}$ scales the full temporal representation at each node–time pair using a factor shared across feature channels. In the ST branch, $1 + [\mathbf{M}_{\text{edge}}]_{ij}$ reweights existing graph

edges before spatial propagation. Thus, the same semantic inputs condition temporal features and spatial interactions through distinct matching signals. Appendix C.1 gives the projection, matching and normalisation details used to generate these signals.

3.4 Dual-Order Spatiotemporal Encoder

We retain temporal-first (TS) and spatial-first (ST) branches with a shared temporal operator $\mathcal{T}$ and spatial operator $\mathcal{S}$. The branches differ in both computation order and the placement of semantic conditioning. Embedding the historical observations gives an initial representation $\mathbf{H}_0 \in \mathbb{R}^{T\times N\times d}$, where d is the hidden width. The temporal operator uses causal depthwise-separable convolution, while the spatial operator aggregates information from neighbouring nodes through graph convolution.

In the TS branch, temporal propagation is followed by the node–time modulation from Section 3.3, then spatial propagation over the fixed graph. In the ST branch, semantic edge modulation first reweights the graph, after which spatial propagation precedes temporal modelling:

$$\begin{aligned}\mathbf{H}_{\mathrm{TS}} &= \mathcal{S}\big((1+\mathbf{M}_{\mathrm{time}})\odot\mathcal{T}(\mathbf{H}_0),\bar{\mathbf{A}}\big)\,,\\ \mathbf{H}_{\mathrm{ST}} &= \mathcal{T}\big(\mathcal{S}\big(\mathbf{H}_0,\mathrm{RowNorm}\big([\bar{\mathbf{A}}\odot(1+\mathbf{M}_{\mathrm{edge}})]_+\big)\big)\big)\,.\end{aligned} \tag{3}$$

The matrix $\bar{\mathbf{A}}$ is obtained by adding self-loops to the input adjacency and normalising it. Here, $[\cdot]_+$ clips entries at zero, and $\mathrm{RowNorm}$ divides each entry by its row sum plus 10^{-6}. The element-wise product $\odot$ applies $\mathbf{M}_{\mathrm{time}}$ to temporal features with broadcasting over feature channels and $\mathbf{M}_{\mathrm{edge}}$ to graph weights before spatial propagation. Thus, context–node semantic relevance acts on node-wise temporal representations in the TS branch and cross-node propagation in the ST branch.

A context-conditioned gate combines the two order-specific representations using non-negative, sample-level weights g_{TS} and g_{ST} that sum to one:

$$\mathbf{H}_{\mathrm{fuse}} = \mathrm{LN}(g_{\mathrm{TS}}\mathbf{H}_{\mathrm{TS}} + g_{\mathrm{ST}}\mathbf{H}_{\mathrm{ST}})\,. \tag{4}$$

Here, LN denotes LayerNorm, and the gate is computed from the context representation $\mathbf{Z}$. One pair of weights is shared across all nodes and historical positions within each sample.

The fused representation is projected to a scalar residual update at each node–time pair and added to the historical sequence. Temporal and spatial tokenizers encode the updated sequence into T temporal tokens and N spatial tokens, respectively. A frozen pretrained language-model backbone processes their concatenation as continuous tokens. Mean pooling and horizon-specific linear heads then produce the base forecast $\widehat{\mathbf{Y}}_{\mathrm{base}}$. Appendix C details the operators, tokenizers, backbone configuration and training procedure.

3.5 Retrieval-Augmented Simplicial Residual Projection

Similar histories can be followed by different joint responses across nodes. We use retrieved responses to correct the base forecast at two levels: weighted global projection models joint cross-node responses, while node-level projection allows each node to follow a different local combination of historical trajectories. Both projections are non-parametric, and their forecasts are combined using node-wise adaptive weights after base-model training.

Residual response memory. Each training window contributes a retrieval key $\mathbf{k}_\ell$ and a residual response $\mathbf{V}_\ell$ to the training-only memory $\mathcal{M} = \{(\mathbf{k}_\ell, \mathbf{V}_\ell)\}_{\ell=1}^{L}$:

$$\mathbf{V}_\ell = \mathbf{Y}_\ell - \mathbf{x}_{\ell,\mathrm{last}}\mathbf{1}_H^{\top}. \tag{5}$$

Here, L is the memory size, $\mathbf{x}_{\ell,\mathrm{last}}$ is the last observed state across all nodes, and $\mathbf{1}_H$ broadcasts it over the forecast horizon. Each key concatenates per-node historical statistics (level, variability and recent changes, including sample-normalised counterparts) with known future calendar features, then applies feature-wise standardisation using training-memory statistics. For a query q, we retrieve the K nearest keys under squared Euclidean distance and denote their indices by $\mathcal{I}_q$. Appendix D.1 gives the retrieval details.

Global and node-level supports. The two projections use the same retrieved cases but different response coordinates. Let $\mathcal{L}_g(\mathbf{V}) = \mathrm{vec}(\mathbf{D}^{-1}\mathbf{V})$ describe the global view, where the positive diagonal matrix $\mathbf{D}$ contains node scales computed from training-memory residuals. For node i, $\mathcal{L}_i(\mathbf{V}) = \mathbf{V}_{i,:} \in \mathbb{R}^H$ retains its unscaled trajectory. In each view $v \in \{g, 1, \ldots, N\}$, the candidates $\mathbf{z}^v_{q,j} = \mathcal{L}_v(\mathbf{V}_j)$, $j \in \mathcal{I}_q$, form a local simplicial support:

$$|\mathcal{K}^v_q| = \bigcup_{\sigma \in \mathcal{K}^v_q} \mathrm{conv}\{\mathbf{z}^v_{q,j} : j \in \sigma\}, \qquad |\sigma| \le 3. \tag{6}$$

Edges connect mutual nearest neighbours within an adaptive radius; three pairwise connected vertices form a triangular face. These relations are constructed separately in each view. Thus, each projection admits local vertices, edges and triangles. Appendix D.2 specifies the scales and neighbourhoods.

Base-guided global projection. Write the base residual as $\mathbf{B}_q = \widehat{\mathbf{Y}}_{\mathrm{base},q} - \mathbf{x}_{q,\mathrm{last}}\mathbf{1}_H^\mathsf{T}$. For a simplex σ, let $\Delta_\sigma = \{\boldsymbol{\alpha} : \alpha_j \ge 0, \sum_{j\in\sigma}\alpha_j = 1\}$ denote its interpolation weights. The global path jointly selects a local simplex and its weights by minimising the discrepancy from the base residual over all nodes and forecast steps:

$$\begin{aligned}(\sigma^{g\star}_q, \boldsymbol{\alpha}^{g\star}_q) &\in \arg\min_{\substack{\sigma\in\mathcal{K}^g_q \\ \boldsymbol{\alpha}\in\Delta_\sigma}} \left\| \mathbf{D}^{-1}\left(\mathbf{B}_q - \sum_{j\in\sigma}\alpha_j\mathbf{V}_j\right)\right\|_F^2, \\ \widehat{\mathbf{Y}}_{\mathrm{global},q} &= \mathbf{x}_{q,\mathrm{last}}\mathbf{1}_H^\mathsf{T} + \sum_{j\in\sigma^{g\star}_q}\alpha^{g\star}_{q,j}\mathbf{V}_j.\end{aligned} \tag{7}$$

Here, $\|\cdot\|_F$ is the Frobenius norm. This is equivalent to Euclidean projection onto $|\mathcal{K}^g_q|$ in the scaled, vectorised coordinates. One weight vector is shared across all nodes and horizons, retaining the cross-node structure of the selected historical responses. The reconstruction uses residuals in their original units, thereby undoing the global scaling before adding the query's last observed state.

Base-guided node-level projection. For each node i, we independently project its H-step base residual onto the local support constructed from that node's retrieved trajectories:

$$\begin{aligned}(\sigma^{i\star}_q, \boldsymbol{\alpha}^{i\star}_q) &\in \arg\min_{\substack{\sigma\in\mathcal{K}^i_q \\ \boldsymbol{\alpha}\in\Delta_\sigma}} \left\| [\mathbf{B}_q]_{i,:} - \sum_{j\in\sigma}\alpha_j[\mathbf{V}_j]_{i,:}\right\|_2^2, \\ [\widehat{\mathbf{Y}}_{\mathrm{node},q}]_{i,:} &= x_{q,\mathrm{last},i}\mathbf{1}_H^\mathsf{T} + \sum_{j\in\sigma^{i\star}_q}\alpha^{i\star}_{q,j}[\mathbf{V}_j]_{i,:}.\end{aligned} \tag{8}$$

Each node can select a different simplex and weight vector from the shared retrieved cases, with its weights held constant across the forecast horizon. Stacking these trajectories gives $\widehat{\mathbf{Y}}_{\mathrm{node},q}$, allowing node-specific corrections alongside the joint global forecast. For both paths, the solver evaluates admissible vertices, edges and triangles and retains the closest projected point; Appendix D.3 gives the closed-form simplex projections and numerical details.

Adaptive fusion. Let $\widehat{\mathbf{Y}}_{g,q} = \widehat{\mathbf{Y}}_{\mathrm{global},q}$ and $\widehat{\mathbf{Y}}_{n,q} = \widehat{\mathbf{Y}}_{\mathrm{node},q}$ denote the reconstructed forecasts in their original measurement units. For each query q and node i, we measure the mean absolute change from the base forecast over the H prediction steps:

$$d_{v,q,i} = \frac{1}{H}\sum_{h=1}^{H}\left|[\widehat{\mathbf{Y}}_{v,q}]_{i,h} - [\widehat{\mathbf{Y}}_{\mathrm{base},q}]_{i,h}\right|, \qquad v \in \{g, n\}. \tag{9}$$

These quantities measure correction magnitudes, without requiring future observations. We assign a larger weight to the forecast with the smaller change from the base forecast:

$$(w_{g,q,i}, w_{n,q,i}) = \begin{cases}\left(\dfrac{d_{n,q,i}}{d_{g,q,i}+d_{n,q,i}}, \dfrac{d_{g,q,i}}{d_{g,q,i}+d_{n,q,i}}\right), & d_{g,q,i}+d_{n,q,i} > 0, \\ (\frac{1}{2}, \frac{1}{2}), & d_{g,q,i}+d_{n,q,i} = 0.\end{cases} \tag{10}$$

The final prediction is

$$[\widehat{\mathbf{Y}}_q]_{i,h} = w_{g,q,i}[\widehat{\mathbf{Y}}_{g,q}]_{i,h} + w_{n,q,i}[\widehat{\mathbf{Y}}_{n,q}]_{i,h}. \tag{11}$$

The weights vary across queries and nodes but are shared across prediction steps. The base forecast serves as the projection target and the reference for correction magnitude; it is not added as a third fusion component. The base forecaster remains fixed. Local geometric constraints apply to each projected component before fusion. Appendix D.3 details projection and reconstruction; Appendix D.4 gives the inference procedure.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate SDF-SP on six real-world datasets spanning tourism, transportation and energy: Beijing Scenic, Shanghai Scenic, Beijing Metro, Beijing Bus, the Hog electricity subset of Building Data Genome 2 (BDG2; Miller et al., 2020), and SDWPF (Zhou et al., 2024). All methods use the same chronological 70%/10%/20% split within each dataset and predict the next eight steps from 24 historical observations. Appendix A provides detailed dataset descriptions.

Baselines. We compare SDF-SP with 15 baselines in four groups. The time-series models are DLinear (Zeng et al., 2023), TimeMixer (Wang et al., 2024), iTransformer (Liu et al., 2024b) and TimeBridge (Liu et al., 2025c). The spatiotemporal models are DPSTGC (Zheng et al., 2025), ST-MambaSync (Shao et al., 2025) and STEMLP (Jiang et al., 2025). The retrieval-augmented models are RAFT (Han et al., 2025), TS-RAG (Ning et al., 2025), SARAF (Zhou et al., 2026) and PFRP (Du et al., 2026). The LLM-based models are GPT4TS (Zhou et al., 2023), Time-LLM (Jin et al., 2024), CALF (Liu et al., 2025b) and TimeCMA (Liu et al., 2025a). Each baseline follows its method-specific training configuration.

Metrics. We report mean absolute error (MAE) and root mean squared error (RMSE) in each dataset's original measurement units; lower values indicate better performance. All reported MAE and RMSE values are averaged over three independent runs with different random seeds. Table 1 reports eight-step results, while Appendix E reports four-step results.

4.2 MAIN RESULTS

Table 1 shows that SDF-SP achieves strong forecasting performance on all six datasets spanning tourism, transportation and energy. It ranks first in ten of the twelve MAE and RMSE comparisons and second in the remaining two. These results cover visitor flows, transit demand, building electricity consumption and wind power, despite differences in node semantics, temporal patterns and spatial relationships. Four-step results also show strong performance (Appendix E).

The two stages of SDF-SP address complementary forecasting needs. The dual-order encoder models temporal dynamics and spatial interactions through temporal-first and spatial-first processing. Context–node semantic relevance guides feature modelling within nodes and information exchange across nodes. Historical response retrieval then refines the base forecast using observed continuations. Global and node-level projections account for coordinated responses across nodes and individual node trajectories, respectively.

4.3 MODEL ANALYSIS

4.3.1 ABLATION STUDIES

We assess dual-order encoding by comparing the full encoder with three variants on Beijing Scenic, Beijing Metro and SDWPF (Table 2): w/o Dual-order, ST only and TS only. The first variant directly tokenizes the history; the single-order variants retain their respective semantic modulation. The full encoder yields the lowest MAE and RMSE for the base forecast on all three datasets. Its advantage over both single-order variants supports the complementary contributions of the two semantically conditioned branches.

Table 1: Eight-step forecasting results on six datasets. Lower is better. Best and second-best values are in **bold** and underlined, respectively. Blocks separate time-series, spatiotemporal, retrieval-augmented and LLM-based baselines.

	Beijing Scenic		Shanghai Scenic		Beijing Metro		Beijing Bus		BDG2		SDWPF	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
DLinear (2023)	660.31	1539.57	784.98	1689.55	49.71	132.66	44.46	71.08	22.38	53.53	310.09	427.19
TimeMixer (2024)	543.98	1148.49	696.85	1558.50	42.22	131.85	42.04	69.41	29.53	64.36	328.57	450.28
iTransformer (2024)	400.87	1079.47	760.17	1794.03	89.90	211.50	61.48	99.91	15.96	42.62	367.14	488.31
TimeBridge (2025)	389.14	1078.93	740.66	1751.96	65.85	155.74	61.75	101.95	15.75	41.23	330.11	446.53
DPSTGC (2025)	442.51	1038.04	682.31	1513.59	58.81	156.77	80.81	172.32	15.83	41.67	324.57	435.78
ST-MambaSync (2025)	392.03	1059.79	612.37	1143.18	43.04	132.25	68.60	157.42	16.42	41.77	338.57	449.30
STEMLP (2025)	398.20	1140.22	731.30	1692.74	45.22	**125.37**	44.13	76.42	15.96	42.48	315.66	429.81
RAFT (2025)	451.72	1186.16	743.99	1629.83	78.28	168.74	56.19	91.05	17.12	42.93	322.73	445.04
TS-RAG (2025)	402.80	1180.17	749.51	1754.51	55.45	164.26	46.22	78.50	**13.69**	40.35	325.69	445.36
SARAF (2026)	418.75	1086.43	705.28	1583.61	89.64	188.45	58.95	94.81	15.95	41.95	326.29	451.48
PFRP (2026)	408.00	1089.00	683.93	1466.96	90.76	205.12	54.93	89.82	17.52	45.74	331.87	442.08
GPT4TS (2023)	415.65	1158.14	767.20	1774.54	104.42	236.19	72.67	119.74	16.77	43.23	316.16	435.03
Time-LLM (2024)	414.68	1138.42	687.51	1569.38	193.36	365.15	143.27	223.70	16.10	42.38	316.56	428.65
CALF (2025)	383.34	1068.49	692.86	1566.47	108.91	249.61	62.70	105.42	16.52	43.04	311.95	432.60
TimeCMA (2025)	380.90	975.25	710.99	1570.46	141.22	273.54	95.76	140.55	16.83	42.14	372.87	470.83
SDF-SP	**370.03**	**901.63**	**423.74**	**784.15**	**40.97**	128.79	**39.75**	**65.62**	14.62	**37.99**	**297.67**	**407.59**

We then evaluate retrieval correction across all four configurations by comparing their forecasts before and after correction. Retrieval reduces MAE in every case. With the full encoder, MAE decreases by 8.14–26.89%, and RMSE also decreases on all three datasets. The corrected forecasts remain the most accurate among the four variants, showing that retrieval provides further gains while retaining the advantage of dual-order encoding.

Table 2: Eight-step ablation results. Best values within each dataset and stage are bold.

		Base		Base + Retrieval	
Dataset	Encoder	MAE	RMSE	MAE	RMSE
Beijing Scenic	w/o Dual-order	600.77	1278.67	390.86	953.54
	ST only	631.76	1441.91	397.01	983.54
	TS only	545.67	1211.50	390.83	970.75
	Dual-order	**501.75**	**1065.58**	**370.03**	**901.63**
Beijing Metro	w/o Dual-order	71.38	187.05	42.36	141.54
	ST only	67.59	170.61	46.71	147.63
	TS only	58.24	148.69	41.53	130.85
	Dual-order	**56.04**	**140.46**	**40.97**	**128.79**
SDWPF	w/o Dual-order	333.71	433.74	317.01	439.51
	ST only	400.40	508.96	341.30	464.95
	TS only	349.21	445.28	323.36	446.08
	Dual-order	**324.03**	**426.35**	**297.67**	**407.59**

4.3.2 Semantic Conditioning

Semantic injection. We compare three semantic injection strategies (Figure 3). The "Without modulation" variant removes feature and edge modulation and uses the fixed graph. The "Input concatenation" variant appends node and context semantics to node–time features and projects the result to the original width. The "Order-specific modulation" variant conditions TS features and ST edges. All variants retain the shared operators, context-conditioned gate, backbone and prediction heads. Concatenation reduces MAE and RMSE on all three datasets, supporting the use of node and context semantics in the input features. Order-specific modulation yields the largest reductions throughout, showing the additional benefit of conditioning temporal features and spatial propagation. Reductions are $100(E_{\mathrm{w/o}} - E)/E_{\mathrm{w/o}}$.

Appendix F.2 reports a backbone replacement comparison.

4.3.3 Retrieval Correction Analysis

We compare three correction rules: similarity-weighted aggregation, full convex-hull projection and local simplicial projection (Figure 4; Table 10). Local simplicial projection yields the lowest MAE and RMSE on all three datasets. The gains over full convex-hull projection support restricting interpolation to locally connected historical responses.

We compare global-only, node-only and combined projections to assess the two correction scales. Combining the scales yields the lowest MAE and RMSE on all three datasets, supporting complementary roles for cross-node and node-specific correction. Local simplicial and Global + Node denote the full method in the respective comparisons.

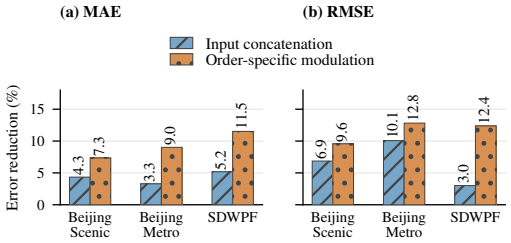

Figure 3: Semantic injection: error reductions (%) over Without modulation. Higher is better (Table 7).

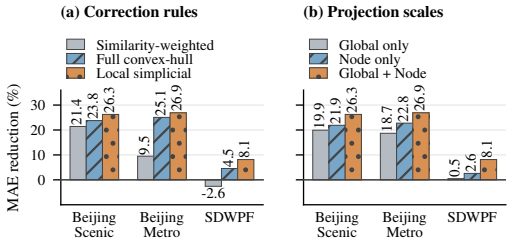

Figure 4: Retrieval analysis: MAE reductions (%) relative to Base. Higher is better.

4.3.4 Applicability to Other Forecasters

To assess applicability beyond SDF-SP, we apply retrieval correction to 15 baseline forecasters (Table 3). We reuse each model's trained checkpoint, keeping its architecture and parameters fixed. The module takes the model's forecast as the projection target and uses historical responses retrieved from the training set for correction. MAE decreases for 11/15, 15/15 and 11/15 models on Beijing Scenic, Beijing Metro and SDWPF, respectively. RMSE decreases for 10/15, 14/15 and 6/15 models, respectively. We report relative error changes as $100(E_{+\mathrm{RAG}} - E)/E$; absolute metrics are provided in Table 11.

Table 3: Relative error changes (%) with our retrieval correction. Negative values indicate improvement.

	Beijing Scenic		Beijing Metro		SDWPF	
Model	ΔMAE	ΔRMSE	ΔMAE	ΔRMSE	ΔMAE	ΔRMSE
DLinear	−26.38%	−17.23%	−15.97%	+3.63%	+1.37%	+4.24%
TimeMixer	−13.85%	+3.47%	−8.12%	−17.75%	−7.29%	−4.79%
iTransformer	+12.14%	+11.37%	−40.59%	−26.43%	−7.97%	−3.40%
TimeBridge	−7.37%	−10.91%	−25.04%	−9.86%	−0.49%	+2.80%
DPSTGC	−16.62%	−8.60%	−24.45%	−4.67%	−0.80%	+3.46%
ST-MambaSync	−4.62%	−6.22%	−10.55%	−39.01%	−2.07%	+2.71%
STEMLP	−6.94%	−12.84%	−16.94%	−28.58%	+1.86%	+4.33%
RAFT	−1.24%	−4.54%	−37.31%	−21.03%	−1.55%	−1.26%
TS-RAG	+12.18%	+4.17%	−22.56%	−18.47%	−2.46%	−0.66%
SARAF	+8.39%	+10.80%	−42.73%	−38.93%	−2.13%	−2.35%
PFRP	−6.43%	−8.89%	−39.10%	−21.39%	−1.07%	+2.82%
GPT4TS	+10.32%	+7.09%	−40.19%	−25.67%	−0.70%	+4.70%
Time-LLM	−8.10%	−10.45%	−63.68%	−50.07%	+1.76%	+5.21%
CALF	−3.65%	−7.24%	−46.82%	−31.57%	+2.97%	+4.94%
TimeCMA	−6.31%	−4.92%	−57.79%	−39.42%	−9.08%	−1.45%

The gains span time-series, spatiotemporal, retrieval-augmented and LLM-based forecasters, demonstrating applicability across the tested architectural families. These improvements are obtained with the original checkpoints, supporting the use of our RAG module as a plug-and-play correction component. Operating directly on forecast outputs allows the module to incorporate historical responses without modifying internal representations or retraining the forecaster. Some retrieval-enhanced baselines outperform SDF-SP on individual metrics. However, SDF-SP achieves the best average rank (2.83) among all 16 methods across the three datasets and two metrics.

5 Conclusion

We presented SDF-SP, a spatiotemporal forecasting framework that combines semantically conditioned dual-order modelling with retrieval-based forecast correction. The encoder learns complementary temporal-first and spatial-first representations, using context–node semantic relevance to guide temporal modelling and spatial propagation. Guided by the base forecast, local simplicial projection combines divergent historical responses at global and node levels to account for cross-node responses and individual node trajectories. Experiments demonstrate strong forecasting performance on all six datasets spanning tourism, transportation and energy. Future work will explore forecast correction when historical responses offer limited coverage of future dynamics.

REFERENCES

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. MSGNet: Learning multi-scale inter-series correlations for multivariate time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11141–11149, 2024.

Fanpu Cao, Lu Dai, Jindong Han, and Hui Xiong. Enhancing multivariate time series forecasting with global temporal retrieval. In *International Conference on Learning Representations*, 2026.

Dazhao Du, Tao Han, and Song Guo. Predicting the future by retrieving the past. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(25):20896–20904, 2026.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O. Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21774–21797, 2025.

Hyotaek Jeon, Hyunwook Lee, Juwon Kim, and Sungahn Ko. ST-LINK: Spatially-aware large language models for spatio-temporal forecasting. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1057–1067, 2025.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):4365–4373, 2023.

Liming Jiang, Baiyi Liu, Huanyu Wang, Shaomiao Chen, and Wei Liang. STEMLP: A spatial-temporal embedding multi-layer perceptron for traffic flow prediction. *Information Sciences*, 721:122602, 2025.

Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate time series forecasting with dynamic graph neural ODEs. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9168–9180, 2023.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Hyunwook Lee and Sungahn Ko. TESTAM: A time-enhanced spatio-temporal attention model with mixture of experts. In *International Conference on Learning Representations*, 2024.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

Aoyu Liu and Yaying Zhang. A general spatio-temporal backbone with scalable contextual pattern bank for urban continual forecasting. In *International Conference on Learning Representations*, 2026.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. TimeCMA: Towards LLM-empowered multivariate time series forecasting via cross-modality alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18): 18780–18788, 2025a.

Jingwei Liu, Ling Yang, Hongyan Li, and Shenda Hong. Retrieval-augmented diffusion models for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, pp. 2766–2786, 2024a.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. CALF: Aligning LLMs for time series forecasting via cross-modal fine-tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18):18915–18923, 2025b.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-Tao Xia. TimeBridge: Non-stationarity matters for long-term time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39815–39840, 2025c.

Xvyuan Liu, Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Chenjuan Guo, Bin Yang, and Jilin Hu. ASTGI: Adaptive spatio-temporal graph interactions for irregular multivariate time series forecasting. In *International Conference on Learning Representations*, 2026.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024b.

Zhiding Liu, Mingyue Cheng, Guanhao Zhao, Jiqian Yang, Qi Liu, and Enhong Chen. Improving time series forecasting via instance-aware post-hoc revision. In *Advances in Neural Information Processing Systems*, volume 38, pp. 35604–35629, 2025d.

Clayton Miller, Anjukan Kathirgamanathan, Bianca Picchetti, Pandarasamy Arjunan, June Young Park, Zoltan Nagy, Paul Raftery, Brodie W. Hobson, Zixiao Shi, and Forrest Meggers. The Building Data Genome Project 2, energy meter data from the ASHRAE Great Energy Predictor III competition. *Scientific Data*, 7:368, 2020.

Kanghui Ning, Zijie Pan, Yu Liu, Yushan Jiang, James Y. Zhang, Kashif Rasul, Anderson Schneider, Lintao Ma, Yuriy Nevmyvaka, and Dongjin Song. TS-RAG: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster. In *Advances in Neural Information Processing Systems*, volume 38, pp. 163170–163199, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. OpenAI technical report.

Zhiqi Shao, Ze Wang, Xusheng Yao, Michael G. H. Bell, and Junbin Gao. ST-MambaSync: Complement the power of Mamba and Transformer fusion for less computational cost in spatial–temporal traffic forecasting. *Information Fusion*, 117:102872, 2025.

Kutay Tire, Ege Onur Taga, Muhammed Emrullah Ildiz, and Samet Oymak. Retrieval augmented time series forecasting. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4294–4302, 2026.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024.

Siyuan Wang, Peng Chen, Yihang Wang, Wanghui Qiu, Chenjuan Guo, Bin Yang, and Yang Shu. Unlocking the value of text: Event-driven reasoning and multi-level alignment for time series forecasting. In *International Conference on Learning Representations*, 2026.

Lan Wu, Xuebin Wang, Chenglong Ge, Ruijuan Chu, and LinYu Wang. ExoTimer: Leveraging large language models for time series forecasting with exogenous variables. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(32):26956–26964, 2026.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019.

Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. TimeRAG: Boosting LLM time series forecasting via retrieval-augmented generation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018.

Chenyang Yu, Xinpeng Xie, Yan Huang, and Chenxi Qiu. FUSE-Traffic: Fusion of unstructured and structured data for event-aware traffic forecasting. In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*, pp. 620–630, 2025.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, 2023.

Huanyu Zhang, Chang Xu, Yi-Fan Zhang, Zhang Zhang, Liang Wang, and Jiang Bian. TimeRAF: Retrieval-augmented foundation model for zero-shot time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 37(9):5654–5665, 2025.

Yingran Zheng, Chao Luo, and Rui Shao. Enhancing traffic flow forecasting with delay propagation: Adaptive graph convolution networks for spatio-temporal data. *IEEE Transactions on Intelligent Transportation Systems*, 26(1):650–660, 2025.

Jingbo Zhou, Xinjiang Lu, Yixiong Xiao, Jian Tang, Jiantao Su, Yu Li, Ji Liu, Junfu Lyu, Yanjun Ma, and Dejing Dou. SDWPF: A dataset for spatial dynamic wind power forecasting over a large turbine array. *Scientific Data*, 11:649, 2024.

Shiqiao Zhou, Holger Schöner, Zipeng Wu, Edouard Fouché, I. A. G. Wilson, and Shuo Wang. Stationarity-aware retrieval-augmented time series forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 6875–6886, 2026.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems*, volume 36, pp. 43322–43355, 2023.

A Datasets and Experimental Setup

A.1 Dataset descriptions

Table 4 summarises the six datasets used in our experiments. The four urban datasets measure visitor or passenger activity, whereas BDG2 and SDWPF extend the evaluation to building electricity consumption and wind power. Each node is a site, station area, building meter or wind turbine, depending on the dataset.

Table 4: Overview of the datasets used in the experiments.

Dataset	Target	Nodes	Sampling	Period
Beijing Scenic	Visitor flow	79	15 min	Jan.–Dec. 2019
Shanghai Scenic	Visitor flow	27	30 min	Jul.–Sep. 2018
Beijing Metro	Entries + exits	329	30 min	May 2019
Beijing Bus	Boardings + alightings	329	30 min	May 2019
BDG2	Electricity consumption	139	1 hour	Jan.–Dec. 2016
SDWPF	Active power	134	1 hour	Jan.–Mar. 2021

Beijing Scenic. This dataset contains visitor-flow observations at 79 scenic sites in Beijing throughout 2019. The forecasting series uses a 15-minute grid from 09:00 to 17:45, giving 36 observations per site per day. Each site forms one node, and the task is to forecast future visitor flows jointly across the sites. Node descriptions are prepared from attraction names, official ratings, categories, main features and intended visitor groups. Daily context descriptions use the compiled Beijing records of calendar, weather, rainfall and events, together with previous-day and historical search-interest summaries (Appendix B).

Shanghai Scenic. This dataset records visitor flows at scenic sites in Shanghai from July to September 2018. The processed version retains 27 sites and aligns their observations to a 30-minute grid from 09:00 to 18:00, giving 19 observations per day over 92 days. Its sampling interval and retained daily observation span differ from those of Beijing Scenic. Node descriptions are prepared from the Shanghai attraction records of ratings, categories, main features and visitor groups. Daily context descriptions use the Shanghai calendar, weather, rainfall and event records, together with previous-day and historical Baidu search-interest summaries.

Beijing Metro. This dataset is constructed from smart-card records collected in May 2019 and covers 329 metro stations. For each station and half-hour interval, the target is the sum of entries and exits. The series retains all 48 half-hour intervals per day. Node descriptions combine station and line information, documented station functions, and passenger-flow profiles computed from training data. Context descriptions use the official holiday calendar and publicly announced events available by the forecast origin, aligned to half-hourly timestamps.

Beijing Bus. This dataset uses bus smart-card records collected in May 2019. The 329 nodes are bus-stop areas, with nearby stops sharing a name grouped into one area. The target is the sum of boardings and alightings within each half-hour interval. These counts measure passenger activity at the selected stop areas. Node descriptions combine route membership records for each stop area with training-period boarding and alighting profiles. Context descriptions use the same Beijing holiday calendar and published event records as the metro dataset, with descriptions adapted to bus travel.

BDG2. We use the 2016 electricity subset from the Hog site of Building Data Genome 2 (Miller et al., 2020). The processed dataset contains hourly readings from 139 building electricity meters over 8,784 hours. The building-meter series are treated as nodes. Node descriptions combine official building-use metadata with training-period electricity-use patterns across daytime, nighttime, weekdays and weekends. Context descriptions combine the forecast-window calendar and US holiday schedule with summaries of the preceding 24 hours of Hog-site weather.

SDWPF. We use the first quarter of 2021 from the full SDWPF dataset (Zhou et al., 2024), covering 134 turbines in one wind farm. The original 10-minute observations are averaged to form

hourly series, yielding 2,160 time points. The prediction target is each turbine's generated active power in kW. Turbine locations describe the spatial layout of the array. Node descriptions are constructed from official turbine coordinates and elevations, relative positions within the farm and nearest-turbine distances. Hourly context descriptions summarise historical wind speed, wind direction, temperature and humidity over the preceding 24 hours.

Spatial graph construction. For the four urban datasets, we compute Haversine distances d_{ij} from node coordinates and assign Gaussian edge weights $A_{ij} = \exp[-d_{ij}^2/(2\sigma^2)]$. The two scenic datasets retain all off-diagonal edges, with σ set to the median of the original full distance matrix. For Beijing Metro and Beijing Bus, $\sigma = 5/\sqrt{2}$ km, equivalent to $A_{ij} = \exp[-(d_{ij}/5)^2]$; weights below 0.1 are set to zero. For BDG2, each meter is connected to the five meters with the strongest positive Pearson correlations, computed from jointly observed training readings. We take the symmetric union of these connections, with correlation coefficients as edge weights. For SDWPF, the graph is the symmetric union of each turbine's eight nearest neighbours, identified by Euclidean distance. The same Gaussian kernel assigns edge weights, with $\sigma \approx 981.460$ m set to the median distance over the directed neighbour pairs. All graphs initially have zero diagonals; the model then adds self-loops and applies symmetric normalisation.

A.2 Temporal split and evaluation protocol

All six datasets use a chronological 70%/10%/20% split. The earliest 70% forms the training set, the following 10% forms the validation set, and the latest 20% forms the test set. The same chronological split is used for all compared methods within each dataset.

Each forecasting example uses 24 historical observations to predict the next eight observations. Partition membership is determined by the forecast targets, which must remain within one partition. Any historical context carried across a partition boundary is restricted to observations preceding the forecast origin. Four-step and eight-step forecasting results are evaluated separately.

SDF-SP constructs its retrieval memory exclusively from training-set examples. Retrieval-key standardisation and the node scales used in global projection are estimated from this training memory and then held fixed for validation and testing. The validation set selects the base checkpoint through early stopping. MAE and RMSE are evaluated in each dataset's original measurement units.

A.3 Baseline models

We compare with 15 baseline models in four groups.

Time-series models. **DLinear** (Zeng et al., 2023) decomposes a series into trend and seasonal components, applies a linear predictor to each, and sums their forecasts. **TimeMixer** (Wang et al., 2024) uses MLPs to mix seasonal and trend components across temporal scales and combines predictions from multiple scales. **iTransformer** (Liu et al., 2024b) represents each variate as a token, using attention to capture relationships across variates and feed-forward layers to learn each variate's representation. **TimeBridge** (Liu et al., 2025c) uses integrated attention to reduce short-term non-stationarity within each variate and cointegrated attention to capture long-term relationships across variates.

Spatiotemporal models. **DPSTGC** (Zheng et al., 2025) combines delay-aware directed graph attention, adaptive graph convolution and temporal gated convolution to model delayed propagation and spatiotemporal dependencies. **ST-MambaSync** (Shao et al., 2025) combines Mamba state-space modelling with Transformer attention to capture spatiotemporal patterns in traffic sequences. **STEMLP** (Jiang et al., 2025) integrates Fourier-based temporal periodicity embeddings and spatial embeddings derived from predefined and adaptive graphs through a joint MLP architecture.

Retrieval-augmented models. **RAFT** (Han et al., 2025) retrieves training histories with patterns similar to the input and incorporates their observed future values into forecasting. **TS-RAG** (Ning et al., 2025) retrieves relevant historical segments with pretrained time-series encoders and uses an Adaptive Retrieval Mixer to combine retrieved patterns with the representations of a time-series foundation model. **SARAF** (Zhou et al., 2026) balances similarity and diversity when selecting

historical references, using stationarity estimates to control retrieval diversification and aggregation of retrieved futures. **PFRP** (Du et al., 2026) retrieves similar patterns from a global memory bank to construct forecasts, then adaptively combines them with outputs from a local prediction model.

LLM-based models. **GPT4TS** (Zhou et al., 2023) adapts a pretrained language Transformer to time-series tasks while freezing its self-attention and feed-forward layers. **Time-LLM** (Jin et al., 2024) reprograms time-series patches with text prototypes and prepends contextual prompts before passing them through a frozen LLM, then projects its representations into forecasts. **CALF** (Liu et al., 2025b) aligns temporal and textual branches through cross-modal input matching, intermediate-feature regularisation and output consistency. **TimeCMA** (Liu et al., 2025a) encodes numerical series and textual prompts in separate branches, then aligns the two modalities to enrich the representations used for forecasting.

A.4 Hardware and software environment

Experiments were conducted on a Windows 11 workstation equipped with one NVIDIA GeForce RTX 5090 D GPU and 32 GB of GPU memory. The software environment used Python 3.9.25 and PyTorch 2.7.1+`cu128`, with the CUDA 12.8 runtime and cuDNN 9.7.1. The NVIDIA driver version was 591.86. NumPy 1.24.3 and Transformers 4.57.6 were used for numerical processing and pretrained-model interfaces, respectively.

The base-forecaster architecture and optimisation settings are detailed in Appendix C; retrieval and projection settings are given in Appendix D. All methods use the same dataset partitions and forecasting task, with method-specific optimisation configurations.

B Semantic Input Preparation

Using Beijing Scenic as an example, we describe how static attraction attributes and daily city-level records are converted into semantic inputs. Qwen3-8B[1] organises this information into natural-language descriptions; Qwen3-Embedding-8B[2] then encodes the prepared descriptions into vectors. Both steps are performed offline, and the forecasting model uses the precomputed embeddings.

B.1 Node attributes and semantic summaries

Each attraction record contains its name, official attraction rating, category, main features and intended visitor groups. These attributes complement the numerical visitor-count history. A node description summarises the attraction type and visitor groups, together with potential sensitivity to holidays, events and weather.

Figure 5 shows Beijing World Park's original attributes and final prepared node summary. The accompanying prompt illustrates the intended transformation from the attributes to a concise description for semantic matching with context.

B.2 Daily context and semantic descriptions

Each daily record contains the date, day of week, weather, rainfall, event description, previous-day mean attraction search index and historical mean search index up to the previous day. The location is set to Beijing. These fields describe city-level conditions shared by all attraction nodes. The two search indices use past days only. Weather and rainfall inputs use only observations available up to the forecast origin.

The prompt asks Qwen3-8B to describe how the event, weather, rainfall, day of week and search interest may jointly influence visitor flows, including the overall direction and strength of the effect.

Figure 6 shows the input record, prompt and generated context description for Beijing on 1 January 2019. The description summarises the potential effects of these conditions on visitor flows.

[1] Official model card: `https://huggingface.co/Qwen/Qwen3-8B`.

[2] Official model card: `https://huggingface.co/Qwen/Qwen3-Embedding-8B`.

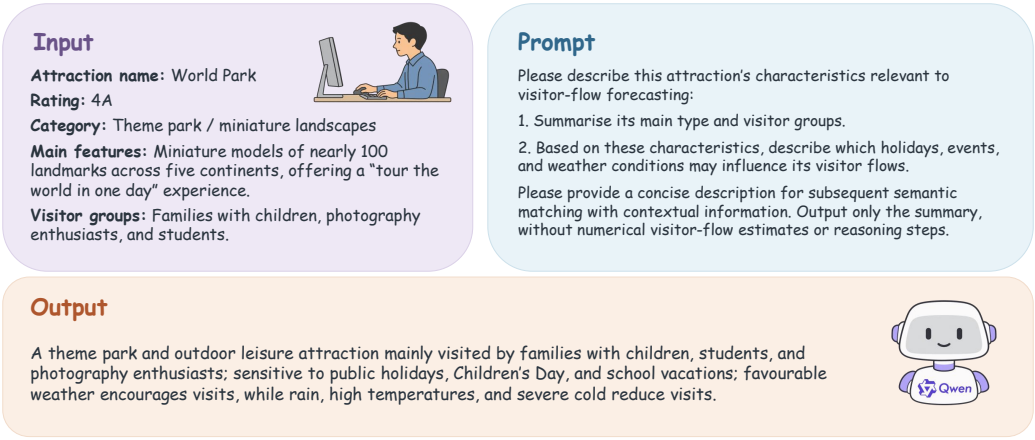

Figure 5: Node semantic preparation illustrated with Beijing World Park.

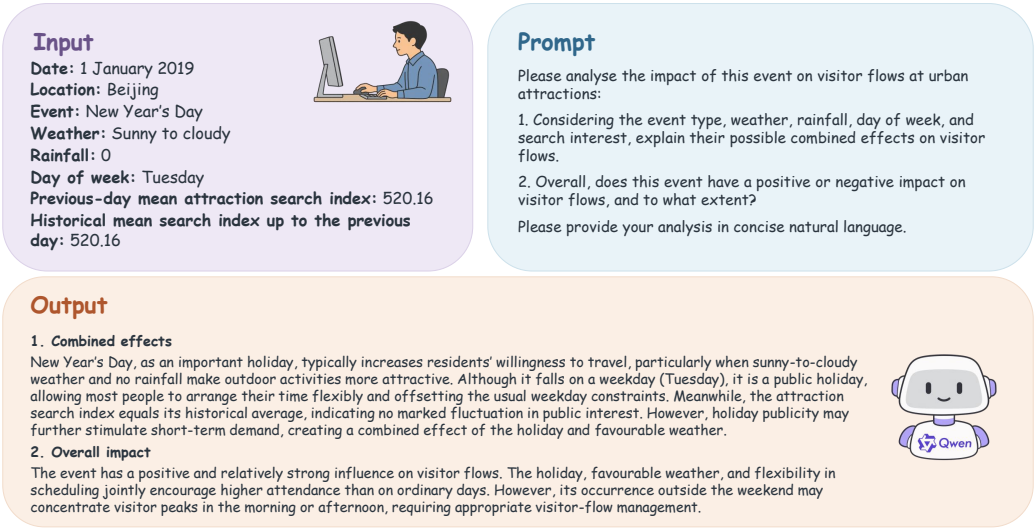

Figure 6: Context semantic preparation for Beijing on 1 January 2019.

B.3 Embedding and use in forecasting

Before encoding, thinking blocks and special tokens are removed from the generated responses. Qwen3-Embedding-8B maps each prepared description to a 4,096-dimensional vector. Attraction summaries provide the node semantic representations, which are fixed for each attraction. Daily descriptions provide the contextual semantic representations, which vary by date and are shared across attraction nodes. The Semantic-Context Modulator uses these two representations to produce node-dependent temporal and spatial modulation signals.

C Base Forecaster Implementation

C.1 Semantic-context modulator

Let $\mathbf{e}^c$ denote the context embedding in $\mathcal{C}_t$, and let $\mathbf{p}_i^\top$ denote row i of the node semantic matrix $\mathbf{P}$. The context encoder f_c first applies affine-free LayerNorm. A bias-free linear projection maps the input from 4,096 to 8 dimensions, followed by GELU. A biased linear projection maps this output to 32 dimensions, followed by GELU and dropout with probability 0.1. Node keys use a separate

affine-free LayerNorm, one bias-free linear projection from 4,096 to $r = 16$ dimensions, and ℓ_2 normalisation. These keys are shared by the temporal query and the two endpoint queries. The context representation, node keys and queries are

$$\begin{aligned} \mathbf{Z} &= f_c(\mathbf{e}^c), \\ \mathbf{K}_i &= \mathrm{L2Norm}(\mathbf{W}_k \,\mathrm{LN}(\mathbf{p}_i)) \,, \\ \mathbf{q}^u &= \mathbf{W}_q^u \mathbf{Z}/\sqrt{r}, \qquad u \in \{\mathrm{time}, 1, 2\}. \end{aligned} \tag{12}$$

Here, $\mathbf{W}_k$ and $\mathbf{W}_q^u$ are learned projections, and $\mathrm{L2Norm}$ produces unit-length keys. The temporal modulation profile and context–node matching scores used in Eq. (2) are

$$\begin{aligned} \mathbf{U} &= \mathbf{W}_{\mathrm{time}}\mathbf{Z} + \mathbf{b}_{\mathrm{time}}, \\ \psi_i &= (\mathbf{q}^{\mathrm{time}})^\mathsf{T}\mathbf{K}_i, \\ m_i^{(1)} &= (\mathbf{q}^1)^\mathsf{T}\mathbf{K}_i, \qquad m_i^{(2)} = (\mathbf{q}^2)^\mathsf{T}\mathbf{K}_i. \end{aligned} \tag{13}$$

The temporal projection has learned weights $\mathbf{W}_{\mathrm{time}}$ and bias $\mathbf{b}_{\mathrm{time}}$, producing $\mathbf{U} \in \mathbb{R}^T$. The fixed matrix $B_{ij} = \cos(\mathbf{p}_i, \mathbf{p}_j)$ has its diagonal set to zero, excluding self-loops from direct semantic rescaling.

For spatial propagation, the given non-negative adjacency is normalised once after adding self-loops:

$$\bar{\mathbf{A}} = \mathbf{D}_A^{-1/2}(\mathbf{A} + \mathbf{I})\mathbf{D}_A^{-1/2}, \qquad [\mathbf{D}_A]_{ii} = \sum_j (A_{ij} + I_{ij}). \tag{14}$$

Here, $\mathbf{A}$ is the input adjacency defined in Section 3.1, and $\mathbf{I}$ is the identity matrix. The fixed matrix $\bar{\mathbf{A}}$ is used directly in the TS branch. In the ST branch, the spatial modulation produces

$$\widetilde{A}_{ij} = \max\{\bar{A}_{ij}(1 + [\mathbf{M}_{\mathrm{edge}}]_{ij}), 0\} \,, \qquad [\mathbf{A}_{\mathrm{ST}}]_{ij} = \frac{\widetilde{A}_{ij}}{\sum_k \widetilde{A}_{ik} + 10^{-6}}. \tag{15}$$

Thus, the model adjusts the weights of the fixed graph according to context.

The temporal modulation head, all three contextual query projections, and the context-conditioned branch-gate head are initialised to zero. This initialisation sets the temporal and edge modulation factors to one and gives equal weights to the two computation orders. At initialisation, the ST branch applies row normalisation to the graph, while the TS branch uses $\bar{\mathbf{A}}$ directly.

C.2 Shared temporal and spatial operators

The encoder operates on tensors of shape $T \times N \times d$ for each sample. A shared linear map followed by LayerNorm embeds each scalar observation to obtain $\mathbf{H}_0$. Both branches reuse the same temporal and spatial modules, with $d = 64$ and dropout probability 0.1. The temporal operator applies LayerNorm, a causal depthwise convolution, a pointwise convolution and a residual feed-forward network:

$$\mathcal{T}(\mathbf{H}) = \mathbf{H} + f_t(\mathrm{PWConv}(\mathrm{DWConv}_{\mathrm{causal}}(\mathrm{LN}_t(\mathbf{H})))) \,. \tag{16}$$

The depthwise convolution has kernel size five and uses left-only padding of four positions. It operates independently along each node's historical sequence. The pointwise convolution mixes feature channels with kernel size one. The network f_t comprises linear maps $d \to 2d \to d$, with GELU after the first map and dropout after GELU and the second map.

The spatial operator aggregates zero- and one-hop representations at each historical position:

$$\begin{aligned} \bar{\mathbf{H}} &= \mathrm{LN}_s(\mathbf{H}), \\ \mathcal{S}(\mathbf{H}, \mathbf{A}') &= \mathbf{H} + f_s\left[\bar{\mathbf{H}} \,\|\, \mathbf{A}'\bar{\mathbf{H}}\right] . \end{aligned} \tag{17}$$

Here, multiplication by $\mathbf{A}'$ acts on the node axis, and $\|$ denotes feature-channel concatenation. The matrix $\mathbf{A}'$ is $\bar{\mathbf{A}}$ in the TS branch and $\mathbf{A}_{\mathrm{ST}}$ in the ST branch. The network f_s uses linear maps $2d \to 2d \to d$, with the same GELU and dropout placement as f_t. The TS factor $1 + \mathbf{M}_{\mathrm{time}}$ multiplies the entire output of $\mathcal{T}$, including its residual connection.

C.3 CONTEXT-CONDITIONED FUSION AND TOKENIZATION

The sample-level gate is computed from the context representation as

$$(g_{\mathrm{TS}}, g_{\mathrm{ST}}) = \mathrm{softmax}[\tanh(\mathbf{W}_g \mathbf{Z} + \mathbf{b}_g)]\,. \tag{18}$$

The affine gate has two outputs, whose weights and biases are initialised to zero. After fusion in Eq. (4), a separate LayerNorm and linear map produce a scalar correction for each node–time pair:

$$\begin{aligned} \Delta X_{i,\tau} &= \mathbf{w}_o^{\mathsf{T}}\, \mathrm{LN}_o([\mathbf{H}_{\mathrm{fuse}}]_{\tau,i,:}) + b_o, \\ \mathbf{X}' &= \mathbf{X} + \alpha \Delta \mathbf{X}. \end{aligned} \tag{19}$$

Here, $\mathbf{X}$ and $\Delta\mathbf{X}$ have shape $N \times T$, and the learnable scale α is initialised to one. The term $\Delta\mathbf{X}$ is the additive correction to the historical sequence.

The temporal tokenizer maps each row of $(\mathbf{X}')^{\mathsf{T}}$ from N values to the backbone width d_{model}. The spatial tokenizer maps each row of $\mathbf{X}'$ from T values to the same width. Each tokenizer consists of a linear map followed by LayerNorm. Their T temporal tokens and N spatial tokens are concatenated in that order:

$$\begin{aligned} \mathbf{z} &= \mathrm{Mean}_{\mathrm{token}}\left[\mathcal{B}_{\mathrm{frozen}}\left(\phi_t((\mathbf{X}')^{\mathsf{T}}) \,\|_{\mathrm{token}}\, \phi_s(\mathbf{X}')\right)\right], \\ [\widehat{\mathbf{Y}}_{\mathrm{base}}]_{:,h} &= \mathbf{W}_h \mathbf{z} + \mathbf{b}_h, \qquad h = 1, \ldots, H. \end{aligned} \tag{20}$$

The operator $\|_{\mathrm{token}}$ concatenates along the token axis, and mean pooling includes all $T + N$ output tokens. Each horizon-specific head maps the pooled representation to all N nodes.

The backbone is GPT-2 (Radford et al., 2019) with 12 layers, 12 attention heads and width $d_{\mathrm{model}} = 768$. The backbone receives continuous embeddings through its embedding-input interface and retains its positional embeddings and causal attention mask. All backbone parameters remain frozen, while gradients propagate through the backbone to the trainable input modules. Dropout follows the model's training or evaluation mode.

C.4 BASE-MODEL TRAINING

Training minimises the mean Huber loss over nodes, horizons and training samples:

$$\begin{aligned} \mathcal{L}_{\mathrm{base}} &= \frac{1}{|\mathcal{D}_{\mathrm{tr}}| N H} \sum_{\ell \in \mathcal{D}_{\mathrm{tr}}} \sum_{i=1}^{N} \sum_{h=1}^{H} \ell_\kappa\Big([\widehat{\mathbf{Y}}_{\mathrm{base},\ell} - \mathbf{Y}_\ell]_{i,h}\Big), \\ \ell_\kappa(e) &= \begin{cases} \frac{1}{2} e^2, & |e| \le \kappa, \\ \kappa(|e| - \frac{1}{2}\kappa), & |e| > \kappa. \end{cases} \end{aligned} \tag{21}$$

Here, $\mathcal{D}_{\mathrm{tr}}$ indexes the training samples, and $\kappa = 1$. The context projections, shared propagation operators, gate, tokenizers and output heads are trained jointly. The Huber loss is the sole base-model training objective. Optimisation uses Adam with learning rate 10^{-3} and batch size 24. Training runs for at most 2,500 epochs, with early stopping after 50 epochs without improvement in validation Huber loss. The checkpoint with the lowest validation loss supplies the base forecasts. The selected checkpoint remains fixed during memory construction, projection and adaptive fusion.

D RETRIEVAL AND SIMPLICIAL PROJECTION DETAILS

D.1 RETRIEVAL KEYS AND TRAINING-ONLY MEMORY

Each memory value is computed from a training window using Eq. (5). The retrieval descriptor combines eight historical statistics per node with eight calendar features per forecast step. For a history $\mathbf{X} \in \mathbb{R}^{N \times T}$, define the sample-normalised sequence $Z^x_{i,\tau} = (X_{i,\tau} - \mu_X)/\max(\sigma_X, 10^{-6})$. Both μ_X and σ_X are computed over all nodes and historical positions within that sample. Table 5 lists the eight statistics. The short-history window used to compute recent changes contains six steps.

For the Beijing Scenic daytime sampling configuration, let $s = (m_{\mathrm{day}} - 540)/15$ be the 15-minute slot index, where m_{day} denotes minutes since midnight. For weekday w, day of year d_y and month

Table 5: Historical retrieval statistics for each node i. The final three statistics use the sample-normalised history $\mathbf{Z}^x$. All statistics use observations available at the forecast origin.

Statistic	Definition
Last value	$X_{i,T}$
Historical mean	$T^{-1}\sum_{\tau=1}^{T} X_{i,\tau}$
Historical standard deviation	$\mathrm{std}_\tau(X_{i,\tau})$
Recent change	$X_{i,T} - \frac{1}{5}\sum_{\tau=T-5}^{T-1} X_{i,\tau}$
Early–late difference	$\frac{1}{6}\sum_{\tau=T-5}^{T} X_{i,\tau} - \frac{1}{6}\sum_{\tau=1}^{6} X_{i,\tau}$
Normalised last value	$Z^x_{i,T}$
Normalised historical mean	$T^{-1}\sum_{\tau=1}^{T} Z^x_{i,\tau}$
Normalised recent change	$Z^x_{i,T} - \frac{1}{5}\sum_{\tau=T-5}^{T-1} Z^x_{i,\tau}$

m, each forecast timestamp is encoded as

$$\mathbf{f}_{\mathrm{cal}} = \left[\sin\frac{2\pi s}{36}, \cos\frac{2\pi s}{36}, \sin\frac{2\pi w}{7}, \cos\frac{2\pi w}{7}, \sin\frac{2\pi d_y}{366}, \cos\frac{2\pi d_y}{366}, \frac{m-1}{11}, \frac{s}{35}\right]. \tag{22}$$

Weekdays are indexed from zero to six. These features are computed from the known forecast timestamps. Concatenating the historical statistics and the H timestamp encodings gives a raw descriptor $\mathbf{f} \in \mathbb{R}^{d_f}$, where $d_f = 8N + 8H$. For Beijing Scenic, $N = 79$, $T = 24$ and $H = 8$, giving 696 retrieval coordinates.

Feature-wise training-memory statistics produce the standardised keys and the query-to-memory distance:

$$\begin{aligned} \mathbf{k} &= \frac{\mathbf{f} - \boldsymbol{\mu}_{\mathcal{M}}}{\max(\boldsymbol{\sigma}_{\mathcal{M}}, 10^{-6})}, \\ d_{\mathrm{key}}(q,\ell) &= d_f^{-1}\|\mathbf{k}_q - \mathbf{k}_\ell\|_2^2. \end{aligned} \tag{23}$$

All operations in the first line are feature-wise, and the same statistics are reused for queries. Dividing by d_f preserves the nearest-neighbour ranking. The implementation retrieves $K = 6$ candidates. The retrieval keys comprise historical statistics and calendar features. Historical future observations supply the memory values.

The memory contains only training samples. Validation data are used for base-model early stopping and checkpoint selection. At test time, each query retrieves historical responses from this training-only memory.

D.2 Response coordinates and local simplicial construction

Both projection paths use the same K retrieved memory items. The global path adjusts each node's scale using only training-memory residual responses:

$$\begin{aligned} c_i &= \max(\mathrm{std}_{\ell,h}[\mathbf{V}_\ell]_{i,h}, 1)\,, \\ s_i &= \sqrt{\max(c_i,\ 0.1\,\mathrm{median}_n\, c_n)}, \qquad \mathbf{D} = \mathrm{diag}(s_1, \ldots, s_N). \end{aligned} \tag{24}$$

The standard deviation is computed over memory items and forecast steps with population normalisation. The square root gives partial scale adjustment, with the residual origin preserved. The same fixed $\mathbf{D}$ is applied to candidates and base residuals for every query. In original coordinates, the squared global distance weights node i by s_i^{-2}. The node-level path uses each node's unscaled H-step trajectory.

For either the scaled global view or a node-level view, write the candidate vectors as $\mathbf{z}_j$ and omit the query and view superscripts for clarity. Their distance is $d_{j\ell} = \|\mathbf{z}_j - \mathbf{z}_\ell\|_2$. The adaptive radius and admissible edges are

$$\begin{aligned} \rho &= \gamma\ \mathrm{median}_{j\in\mathcal{I}_q}\left(\min_{\substack{\ell\in\mathcal{I}_q \\ \ell\neq j}} d_{j\ell}\right), \\ \mathcal{E} &= \{\{j,\ell\} : \ell \in \mathcal{N}_k(j),\ j \in \mathcal{N}_k(\ell),\ d_{j\ell} \le \rho,\ j < \ell\}. \end{aligned} \tag{25}$$

Here, $\mathcal{N}_k(j)$ contains the k nearest other candidates. The implementation uses $k = 3$ and $\gamma = 2$. Pairwise squared distances are computed from a Gram matrix and clipped at zero. Diagonal entries are set to infinity during neighbour selection.

Three vertices form a triangular face when all three edges exist:

$$\begin{aligned} \mathcal{F} &= \{\{j, \ell, m\} : j < \ell < m,\ \{j, \ell\}, \{j, m\}, \{\ell, m\} \in \mathcal{E}\}, \\ \mathcal{K} &= \{\{j\} : j \in \mathcal{I}_q\} \cup \mathcal{E} \cup \mathcal{F}. \end{aligned} \tag{26}$$

The complex includes every face of each admitted simplex and has maximum dimension two. Its support can be non-convex or disconnected. All vertices remain available, including isolated candidates. The construction is performed separately in the global and node-level views, yielding view-specific edges and triangles from the shared candidate set.

D.3 Projection solver and reconstruction

The same Euclidean projection solver is used in all views. We denote the closest point in $|\mathcal{K}_q^v|$ by $\mathbf{p}_q^v$; its simplex and interpolation weights solve the corresponding global or node-level objective in Eqs. (7)–(8). For a target $\mathbf{t} = \mathcal{L}_v(\mathbf{B}_q)$, it first selects the nearest candidate vertex, then checks every admissible edge and non-degenerate triangular face. If no edge or triangle improves the distance, the nearest vertex is retained. For an edge with endpoints $\mathbf{a}$ and $\mathbf{c}$, its closest point is

$$\eta = \mathrm{clip}_{[0,1]} \frac{(\mathbf{t} - \mathbf{a})^\mathsf{T}(\mathbf{c} - \mathbf{a})}{\|\mathbf{c} - \mathbf{a}\|_2^2}, \qquad \mathbf{p}_{\mathrm{edge}} = \mathbf{a} + \eta(\mathbf{c} - \mathbf{a}). \tag{27}$$

Edges with squared length at most 10^{-12} are skipped because their endpoints have already been evaluated.

For a triangle with vertices $\mathbf{a}$, $\mathbf{c}$ and $\mathbf{d}$, define $\mathbf{U}_\triangle = [\mathbf{c} - \mathbf{a}, \mathbf{d} - \mathbf{a}]$. Projection onto its affine plane solves

$$\begin{aligned} (\mathbf{U}_\triangle^\mathsf{T}\mathbf{U}_\triangle)\boldsymbol{\eta} &= \mathbf{U}_\triangle^\mathsf{T}(\mathbf{t} - \mathbf{a}), \\ \mathbf{p}_\triangle &= \mathbf{a} + \mathbf{U}_\triangle\boldsymbol{\eta}. \end{aligned} \tag{28}$$

The plane projection is retained when all barycentric weights $(1 - \eta_1 - \eta_2, \eta_1, \eta_2)$ are positive, placing it inside the triangle. Otherwise, the closest point is represented by an edge or vertex already considered by the solver. The nearest retained point over all simplices gives $\mathbf{p}_q^v$, up to numerical tolerances.

Residual distances and projections are computed in 64-bit floating point. Writing $\mathbf{G}_\triangle = \mathbf{U}_\triangle^\mathsf{T}\mathbf{U}_\triangle$, the solver skips a triangle if $\det(\mathbf{G}_\triangle) \le 10^{-10}\max(G_{\triangle,11}G_{\triangle,22}, 1)$. All three barycentric weights must exceed 10^{-8}. A candidate replaces the current solution when its squared distance is strictly smaller. In a tie, the solver retains the first encountered minimiser. Projected vectors are returned in 32-bit floating point.

The two residual estimates and their reconstructed forecasts are

$$\begin{aligned} \mathbf{R}_{g,q} &= \mathbf{D}\,\mathrm{unvec}_{N,H}(\mathbf{p}_q^g), \quad [\mathbf{R}_{n,q}]_{i,:} = \mathbf{p}_q^i, \\ \widehat{\mathbf{Y}}_{\mathrm{global},q} &= \mathbf{x}_{q,\mathrm{last}}\mathbf{1}_H^\mathsf{T} + \mathbf{R}_{g,q}, \quad \widehat{\mathbf{Y}}_{\mathrm{node},q} = \mathbf{x}_{q,\mathrm{last}}\mathbf{1}_H^\mathsf{T} + \mathbf{R}_{n,q}. \end{aligned} \tag{29}$$

Here, $\mathrm{unvec}_{N,H}$ reverses the global vectorisation. Global scaling is undone after projection; node-level trajectories remain in their original coordinates.

D.4 Adaptive fusion and inference

Test-time inference uses the fixed base checkpoint, training-only memory and node scales estimated from that memory. The two forecasts are reconstructed in their original measurement units before computing node-wise correction magnitudes and fusion weights using Eqs. (9)–(11). The weights require neither future observations nor fitted parameters. Equal correction magnitudes yield equal weights. If both magnitudes are zero, both projections coincide with the base forecast; if only one is zero, that projection receives unit weight. This rule favours smaller corrections relative to the base forecast, without treating correction magnitude as prediction error. Algorithm 1 gives the inference procedure.

Algorithm 1 SDF-SP inference with adaptive fusion

Input: Query inputs $(\mathbf{X}_q, \mathcal{G}, \mathbf{P}, \mathcal{C}_q)$; trained base forecaster; training-only memory $\mathcal{M}$; fixed node-scale matrix $\mathbf{D}$
Output: Final forecast $\widehat{\mathbf{Y}}_q$

Compute $\widehat{\mathbf{Y}}_{\mathrm{base},q}$ using the semantic modulator, dual-order encoder and frozen backbone
Form $\mathbf{k}_q$ using training-memory standardisation
Retrieve the K nearest keys and residual responses $\{\mathbf{V}_j : j \in \mathcal{I}_q\}$
Set $\mathbf{B}_q \leftarrow \widehat{\mathbf{Y}}_{\mathrm{base},q} - \mathbf{x}_{q,\mathrm{last}}\mathbf{1}_H^\top$
for each view $v \in \{g, 1, \ldots, N\}$ **do**
    Transform the target and candidates with $\mathcal{L}_v$
    Construct $\mathcal{K}_q^v$ in that view using Eqs. (25)–(26)
    Find $\mathbf{p}_q^v$ by checking vertices, edges and triangles as in Appendix D.3
end for
Reconstruct $\widehat{\mathbf{Y}}_{\mathrm{global},q}$ and $\widehat{\mathbf{Y}}_{\mathrm{node},q}$ using Eq. (29)
for each node $i = 1, \ldots, N$ **do**
    Compute $d_{g,q,i}$ and $d_{n,q,i}$ from the reconstructed forecasts and the base forecast using Eq. (9)
    if $d_{g,q,i} + d_{n,q,i} > 0$ **then**
        Set $w_{g,q,i} \leftarrow d_{n,q,i}/(d_{g,q,i} + d_{n,q,i})$ and $w_{n,q,i} \leftarrow 1 - w_{g,q,i}$
    else
        Set $w_{g,q,i} \leftarrow 1/2$ and $w_{n,q,i} \leftarrow 1/2$
    end if
    Combine the two node trajectories using Eq. (11)
end for
return $\widehat{\mathbf{Y}}_q$

D.5 Computational cost and scope

Exhaustive retrieval costs $O(Ld_f)$ per query. Each support contains K vertex candidates, at most $\binom{K}{2}$ edges and $\binom{K}{3}$ triangles. The global projection uses NH coordinates and has worst-case computational cost $O(K^3NH)$. The N node-level projections each use H coordinates and together have the same asymptotic cost. For $K = 6$, each view checks at most 15 pairs and 20 triples before neighbourhood filtering; this is repeated for $N + 1$ views. Computing correction magnitudes and fusing the forecasts costs $O(NH)$. The fixed node scales are computed once from the training memory, while fusion weights are computed separately for each query and node.

The global projected residual combines at most three locally connected memory responses with weights shared across all nodes and horizons. Each node-level residual also uses at most three trajectories, but different nodes may select different simplices and weights. The final forecast adaptively combines the global and node-level estimates according to their node-wise correction magnitudes. The resulting interpolation weights can vary across nodes and queries but remain constant across the forecast horizon. The fused forecast can lie outside either component's selected simplex. The local constraints govern each component's admissible response combinations.

E Additional Forecasting Results

Table 6 reports four-step forecasting results. The baseline set and metric units are unchanged.

Table 6: Four-step forecasting results. Lower is better. Best and second-best values are in **bold** and underlined, respectively. Blocks separate time-series, spatiotemporal, retrieval-augmented and LLM-based baselines.

	Beijing Scenic		Shanghai Scenic		Beijing Metro		Beijing Bus		BDG2		SDWPF	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
DLinear (2023)	570.22	1281.48	731.31	1587.09	39.67	84.80	39.47	63.86	19.31	48.06	259.66	368.50
TimeMixer (2024)	462.96	929.77	661.23	1503.11	34.76	83.99	40.45	62.74	28.90	62.27	262.24	368.73
iTransformer (2024)	306.39	840.40	659.79	1536.99	77.36	177.55	52.94	86.35	13.22	32.92	340.77	460.04
TimeBridge (2025)	298.84	839.92	649.28	1535.18	55.28	126.41	53.63	88.74	12.35	33.84	280.08	391.78
DPSTGC (2025)	382.36	868.97	607.09	1363.94	53.79	126.86	81.04	172.40	13.25	33.95	277.15	377.62
ST-MambaSync (2025)	305.51	845.36	528.70	928.98	41.44	90.47	65.49	153.16	12.59	33.45	297.16	403.67
STEMLP (2025)	305.99	872.98	635.95	1470.57	40.75	93.89	41.27	70.01	12.12	32.95	257.28	371.36
RAFT (2025)	332.82	897.81	643.95	1454.53	68.85	149.01	50.95	85.10	12.52	37.65	257.47	385.93
TS-RAG (2025)	303.03	891.59	645.71	1497.12	44.43	112.83	40.64	67.52	**9.93**	**29.55**	265.64	382.54
SARAF (2026)	311.52	847.58	628.29	1440.52	74.22	161.14	53.55	89.39	12.16	35.31	259.80	390.96
PFRP (2026)	308.57	854.65	619.73	1430.41	75.80	166.68	49.58	80.96	12.90	34.09	266.57	372.66
GPT4TS (2023)	313.71	895.40	669.71	1553.04	84.00	193.37	60.38	101.17	13.94	33.33	258.33	378.74
Time-LLM (2024)	316.02	878.02	607.12	1311.22	166.74	331.59	110.41	186.70	13.05	33.81	259.80	369.77
CALF (2025)	298.00	832.00	610.44	1395.98	91.37	219.84	52.97	89.63	12.39	33.88	264.92	370.30
TimeCMA (2025)	299.46	798.21	619.92	1372.88	131.58	258.11	88.01	131.25	13.29	33.13	353.73	450.01
SDF-SP	**292.12**	**744.59**	**367.32**	**743.95**	**33.20**	**81.91**	**37.61**	**61.46**	11.95	32.73	**254.58**	**358.48**

F Additional Model Analysis Results

F.1 Semantic injection

Table 7 reports the absolute eight-step MAE and RMSE underlying Figure 3.

Table 7: Eight-step semantic-injection ablation results corresponding to Figure 3. Best values are bold.

	Beijing Scenic		Beijing Metro		SDWPF	
Variant	MAE	RMSE	MAE	RMSE	MAE	RMSE
Without modulation	399.37	997.01	45.03	147.73	336.43	465.17
Input concatenation	382.04	928.57	43.55	132.86	319.00	451.17
Order-specific modulation	**370.03**	**901.63**	**40.97**	**128.79**	**297.67**	**407.59**

F.2 Backbone replacement

GPT-2 outperforms a 12-layer Transformer replacement on all three datasets (Table 8), reducing MAE by 9.73–24.78% and RMSE by 8.79–18.54%. The largest gains occur on Beijing Metro. This comparison supports the pretrained backbone within SDF-SP. SDF-SP uses node and context semantics to modulate temporal features and spatial propagation. The priors learned during language pretraining may help GPT-2 make more effective use of these semantically conditioned representations, contributing to its advantage over the Transformer replacement.

Table 8: Eight-step backbone replacement results on three datasets. Lower MAE and RMSE are better; best values are bold.

	Beijing Scenic		Beijing Metro		SDWPF	
Backbone	MAE	RMSE	MAE	RMSE	MAE	RMSE
12-layer Transformer	409.90	988.57	54.47	158.10	332.85	459.96
GPT-2 (SDF-SP)	**370.03**	**901.63**	**40.97**	**128.79**	**297.67**	**407.59**

F.3 SPATIAL ATTENTION MASKING

We evaluated bidirectional attention among the N spatial tokens on Beijing Scenic, Beijing Metro and SDWPF. Temporal tokens retained causal attention and could not attend to spatial tokens; spatial tokens could attend to all temporal and spatial tokens. Both variants used the same inputs, data split, training settings and positional embeddings, with the GPT-2 parameters frozen. Both variants used global and node-level projections for forecast correction. Table 9 reports the eight-step results. Spatial bidirectional attention slightly increased both errors on Beijing Scenic, improved both metrics on Beijing Metro, and worsened both on SDWPF. These results show no consistent benefit across the three datasets. We therefore retain the native causal mask to keep the implementation simple.

Table 9: Eight-step attention-mask comparison for the complete model. Best values within each dataset are bold.

Dataset	Attention mask	MAE	RMSE
Beijing Scenic	Native causal	**370.03**	**901.63**
	Spatial bidirectional	373.80	924.95
Beijing Metro	Native causal	40.97	128.79
	Spatial bidirectional	**39.97**	**113.70**
SDWPF	Native causal	**297.67**	**407.59**
	Spatial bidirectional	341.22	464.35

F.4 RETRIEVAL CORRECTION

Table 10 reports the absolute eight-step MAE and RMSE underlying Figure 4, including the projection-scale comparisons.

Table 10: Eight-step retrieval-correction comparisons. Local simplicial projection uses both projection scales and is repeated as Global + Node in the lower block. Best values within each block are bold.

	Beijing Scenic		Beijing Metro		SDWPF	
Variant	MAE	RMSE	MAE	RMSE	MAE	RMSE
Similarity-weighted aggregation	394.31	998.73	50.72	133.36	332.46	454.41
Full convex-hull projection	382.51	929.25	42.00	131.78	309.29	425.42
Local simplicial projection	**370.03**	**901.63**	**40.97**	**128.79**	**297.67**	**407.59**
Global only	401.78	999.80	45.57	142.79	322.48	452.39
Node only	391.92	952.84	43.29	137.60	315.66	439.75
Global + Node	**370.03**	**901.63**	**40.97**	**128.79**	**297.67**	**407.59**

F.5 RETRIEVAL CORRECTION ACROSS FORECASTERS

Table 11 reports the original and corrected metrics underlying Table 3. Relative changes are computed separately for each model, dataset and metric from these paired values.

Table 11: Absolute errors before and after adding our retrieval correction. Original denotes the original forecaster; +RAG denotes its corrected output. Lower values are better; the lower error in each Original/+RAG pair is in **bold**. Blocks follow the baseline categories in Table 1.

	Beijing Scenic				Beijing Metro				SDWPF			
	MAE		RMSE		MAE		RMSE		MAE		RMSE	
Model	Original	+RAG	Original	+RAG	Original	+RAG	Original	+RAG	Original	+RAG	Original	+RAG
DLinear	660.31	**486.11**	1539.57	**1274.36**	49.71	**41.77**	**132.66**	137.48	**310.09**	314.33	**427.19**	445.30
TimeMixer	543.98	**468.63**	**1148.49**	1188.37	42.22	**38.79**	131.85	**108.45**	328.57	**304.62**	450.28	**428.70**
iTransformer	**400.87**	449.54	**1079.47**	1202.16	89.90	**53.41**	211.50	**155.59**	367.14	**337.89**	488.31	**471.73**
TimeBridge	389.14	**360.47**	1078.93	**961.27**	65.85	**49.36**	155.74	**140.39**	330.11	**328.48**	**446.53**	459.02
DPSTGC	442.51	**368.96**	1038.04	**948.79**	58.81	**44.43**	156.77	**149.45**	324.57	**321.96**	**435.78**	450.85
ST-MambaSync	392.03	**373.92**	1059.79	**993.87**	43.04	**38.50**	132.25	**80.66**	338.57	**331.55**	**449.30**	461.46
STEMLP	398.20	**370.57**	1140.22	**993.87**	45.22	**37.56**	125.37	**89.54**	**315.66**	321.52	**429.81**	448.42
RAFT	451.72	**446.12**	1186.16	**1132.26**	78.28	**49.07**	168.74	**133.25**	322.73	**317.74**	445.04	**439.42**
TS-RAG	**402.80**	451.85	**1180.17**	1229.40	55.45	**42.94**	164.26	**133.92**	325.69	**317.67**	445.36	**442.43**
SARAF	**418.75**	453.88	**1086.43**	1203.74	89.64	**51.34**	188.45	**115.09**	326.29	**319.35**	451.48	**440.88**
PFRP	408.00	**381.76**	1089.00	**992.17**	90.76	**55.27**	205.12	**161.24**	331.87	**328.33**	**442.08**	454.53
GPT4TS	**415.65**	458.53	**1158.14**	1240.22	104.42	**62.45**	236.19	**175.55**	316.16	**313.96**	**435.03**	455.46
Time-LLM	414.68	**381.08**	1138.42	**1019.47**	193.36	**70.23**	365.15	**182.31**	**316.56**	322.12	**428.65**	450.98
CALF	383.34	**369.36**	1068.49	**991.09**	108.91	**57.92**	249.61	**170.80**	**311.95**	321.23	**432.60**	453.98
TimeCMA	380.90	**356.85**	975.25	**927.29**	141.22	**59.61**	273.54	**165.70**	372.87	**339.01**	470.83	**464.02**

G LLM USAGE

In accordance with the ICLR 2027 policy on large language model (LLM) usage, we disclose that we used an LLM solely for language editing, including improvements to grammar, clarity, and fluency. All research ideas, manuscript content and structure, technical contributions, experimental design, data analysis, and scientific conclusions were conceived, developed, and validated by the human authors. The LLM was not involved in research ideation, experimental design, data analysis, or the creation of scientific content. All LLM-assisted edits were reviewed and revised by the authors to ensure accuracy and correctness. All use of LLMs complies with the ICLR Code of Ethics.