GENERALIZABLE SPATIO-TEMPORAL FORECASTING BEYOND THE CLOSED-WORLD SETTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spatio-temporal forecasting is fundamental to many real-world systems, yet deployment environments are inherently varying: temporal distributions may evolve, spatial domains may change, and observation availability may shift from training conditions. Existing models are often tightly coupled to their training environments through node-specific representations, predefined spatial support sets, and reliance on consistently available observations, limiting generalization to changing environments. To this end, we propose **VaryST**, a compact spatio-temporal forecasting model trained from scratch that achieves strong generalization by dynamically constructing predictive context from available long-term observations. To accommodate temporal changes, VaryST derives node representations from long-term histories, capturing current behavior in the context of longer-term patterns. To adapt to spatial changes, it dynamically selects functionally relevant support nodes beyond predefined neighborhoods and integrates their information with recent and long-term target states and global context. To handle varying observation availability, VaryST selectively masks historical patches on which the model relies most during training, encouraging it to exploit alternative temporal and spatial dependencies for accurate forecasting when key observations become unavailable. Extensive experiments on six real-world datasets against seventeen baselines show that VaryST achieves strong in-domain forecasting accuracy, substantially improves zero- and few-shot generalization under spatial and temporal changes, and remains robust to varying observation availability.

1 INTRODUCTION

Spatio-temporal (ST) forecasting predicts the future states of spatially distributed nodes by modeling temporal dynamics and spatial dependencies, supporting applications such as traffic management, energy systems, and environmental monitoring. Despite substantial progress, many existing methods operate within a largely *closed-world setting*, assuming that temporal distributions, spatial domains, and observation availability remain stable between training and deployment.

Real-world deployment, however, is inherently varying, as shown in Figure 1. *Temporal distributions* evolve with seasonal variations and long-term trends. *Spatial domains* change as models transfer to new cities or sensor networks expand. *Observation availability* varies with sensor failures and communication interruptions, producing missing rates and patterns that differ from those encountered during training. These changes raise a central question: **Can a spatio-temporal forecasting model remain effective as where it is deployed, when it is deployed, and what observations are available change?**

Existing methods face several limitations in such vayring deployment environments. Specifically, node-specific embeddings and training-graph-dependent parameters restrict direct transfer to unseen nodes or networks. Besides, many models draw information only from a predefined set of nodes. Even when they dynamically adjust the importance of these nodes, they cannot seek useful information from nodes outside that set. Models may also rely heavily on observations that are consistently available during training, making accurate forecasting difficult when these observations are missing at deployment. These limitations reflect a common challenge: **the forecasting mechanism is tightly coupled to the training environment.**

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.

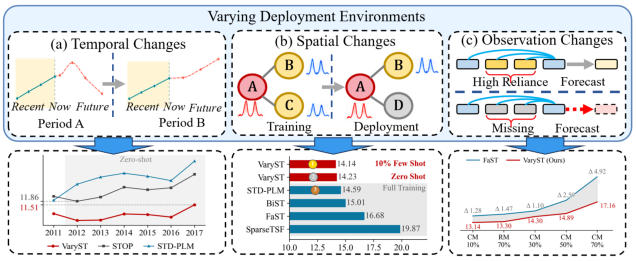


Figure 1: ST models in varying deployment environments

This coupling is problematic because the information useful for forecasting is itself environment-dependent. Temporal shifts require reassessing the target's current behavior; spatial changes may alter which nodes provide useful support; and changes in observation availability may make previously useful information unavailable. We refer to the information selected to support a forecast as predictive evidence, and its organization as predictive context. We therefore argue that **a generalizable model should dynamically construct its predictive context from currently available observations, rather than inherit it from the training environment.** This requires characterizing the target node's current behavior, identifying where relevant evidence can be found, and organizing that evidence for the current prediction.

Long-term observations provide the basis for dynamically constructing predictive context. A short recent window captures immediate dynamics but offers limited evidence for distinguishing persistent behavior from transient fluctuations. Longer histories reveal recurring patterns and operating regimes, while providing alternative cues when recent observations are incomplete. They therefore help characterize the target node's current behavior and identify useful predictive support in the current deployment environment. Their value thus extends beyond a longer input horizon to enabling predictive context to be reconstructed as deployment conditions change.

Building on this perspective, we propose **VaryST**, a compact framework trained from scratch for generalizable **ST** forecasting in **Vary**ing deployment environments. VaryST dynamically constructs target-specific predictive contexts from available long-term observations. Specifically, Long-Term Observation-Derived Representations characterize node behavior without fixed identities, while Dynamic Functional Support Node Retrieval identifies relevant support nodes beyond predefined neighborhoods. Target-Centric Hierarchical Contextual Decoding organizes recent and long-term target states, retrieved supports, and global context across multiple temporal scales and spatial scopes, providing a unified interface for inductive inference. To improve robustness to varying observation availability, Importance-Guided Observation Perturbation (IGOP) masks highly relied-upon observations during training, encouraging the model to reorganize predictive evidence. Together, these designs enable VaryST to reconstruct predictive context as deployment conditions change. Our main contributions are summarized as follows:

- We identify training-environment coupling as a common challenge across temporal shifts, spatial changes, and varying observation availability, and propose **VaryST** to address it through dynamic predictive context construction.
- We propose Long-Term Observation-Derived Representations and Dynamic Functional Support Node Retrieval to characterize nodes without fixed identities and identify relevant support beyond predefined neighborhoods, enabling adaptation to temporal shifts and spatial changes.
- We develop Target-Centric Hierarchical Contextual Decoding to organize multi-scale target-specific evidence through a unified inductive interface, together with Importance-Guided Observation Perturbation for robustness to varying observation availability.
- Extensive experiments on six real-world datasets against seventeen baselines demonstrate strong in-domain forecasting, zero-/few-shot generalization across temporal and spatial changes, and robustness to unseen observation conditions.

2 Related Work

2.1 Spatio-Temporal Forecasting

A central challenge in spatio-temporal forecasting is modeling temporal dependencies within each node and spatial dependencies across nodes. Recurrent models such as LSTM (Hochreiter & Schmidhuber, 1997) capture temporal dynamics. DCRNN (Li et al., 2018), STGCN (Yu et al.,

2018), Graph WaveNet (Wu et al., 2019), and AGCRN (Bai et al., 2020) combine recurrent or temporal convolutional modeling with graph-based spatial modeling, while ASTGCN (Guo et al., 2019), ASTGNN (Guo et al., 2021) and PDFormer (Jiang et al., 2023) incorporate attention to capture spatio-temporal dependencies. These models achieve strong in-domain performance. However, some rely on node-specific embeddings or adjacency parameters tied to the training network, limiting their direct application to unseen nodes or networks. Recently, pretrained language model (PLM)-based approaches and spatio-temporal foundation models have demonstrated strong forecasting and generalization capabilities. OFA (Zhou et al., 2023), ST-LLM (Liu et al., 2024a), STG-LLM, (Liu et al., 2024b) and STD-PLM (Huang et al., 2025) adapt pretrained language models to spatio-temporal modeling, while UniST (Yuan et al., 2024) learns transferable knowledge from diverse spatio-temporal data. These advances highlight the value of pretrained representations and transferable knowledge. However, such capabilities alone do not ensure that a model can identify alternative predictive evidence when node behavior changes or previously available observations become missing. Some studies investigate out-of-distribution (OOD) spatio-temporal forecasting to improve robustness across environments, including spatial and temporal shifts in STONE (Wang et al., 2024) and STOP (Ma et al., 2025b). While these studies focus on particular deployment challenges, we consider a broader perspective involving changes in spatial domains, temporal distributions, and observation availability.

2.2 Long-Context Spatio-Temporal Modeling

PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024c), and SparseTSF (Lin et al., 2026) demonstrate the value of extended temporal dependencies but do not explicitly incorporate spatial graph structure. SSTBAN (Guo et al., 2023), STID (Shao et al., 2022), BiST (Ma et al., 2025a), and FaST (Zhao et al., 2026) explore efficient spatio-temporal modeling, while RAST (Ruan et al., 2026) retrieves historical patterns to augment prediction. STMAE (Sun et al., 2024) and BigST (Han et al., 2024) further exploit historical representations through separate pretraining stages. These studies motivate efficient use of historical information for forecasting. These studies motivate efficient historical modeling. VaryST further exploits long-term observations directly to guide spatial support selection and predictive evidence organization.

3 Preliminaries

Spatio-Temporal Data. We consider a spatio-temporal system defined over a spatial graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the set of N spatial nodes and $\mathcal{A}$ represents their spatial relations. At time t, the observations are denoted by $\mathbf{X}_t \in \mathbb{R}^{N \times C}$, where C is the number of features. Their availability is indicated by a binary mask $\mathbf{M}_t \in \{0, 1\}^{N \times C}$, where $M_{t,i,c} = 1$ if the corresponding observation is available.

Varying Deployment Environments. We characterize deployment environments along three dimensions: temporal distributions, spatial domains, and observation availability. These dimensions may differ between training and deployment or change during deployment. *Evolving temporal distributions* refer to changes in observation distributions over time, $P_t(\mathbf{X}) \neq P_{t'}(\mathbf{X})$, due to seasonality, long-term trends, or changing operating regimes. *Changing spatial domains* refer to changes in the underlying graph, $\mathcal{G} \neq \mathcal{G}'$, including the addition or removal of nodes and spatial relations, or transfer to a different network. *Varying observation availability* refers to changes in the distribution of observation masks, $P(\mathbf{M}) \neq P'(\mathbf{M})$, reflecting different missing rates or patterns.

Generalizable Spatio-Temporal Forecasting. Given observations $\mathbf{X}_{t-T+1:t}$ of $\mathcal{G}$ over the past T time steps and their availability masks $\mathbf{M}_{t-T+1:t}$, we aims to predict the future Q steps:

$$\hat{\mathbf{X}}_{t+1:t+Q} = f_\theta \left(\mathbf{X}_{t-T+1:t}, \mathbf{M}_{t-T+1:t}; \mathcal{G} \right). \tag{1}$$

Our goal is to learn a forecasting model that generalizes beyond its training environment to deployment environments with different temporal distributions, spatial graphs, or observation availability. For spatial and temporal generalization, we consider both zero-shot forecasting without parameter updates and few-shot adaptation using limited target-environment training data. For changes in observation availability, we evaluate generalization to unseen missing rates and patterns without retraining.

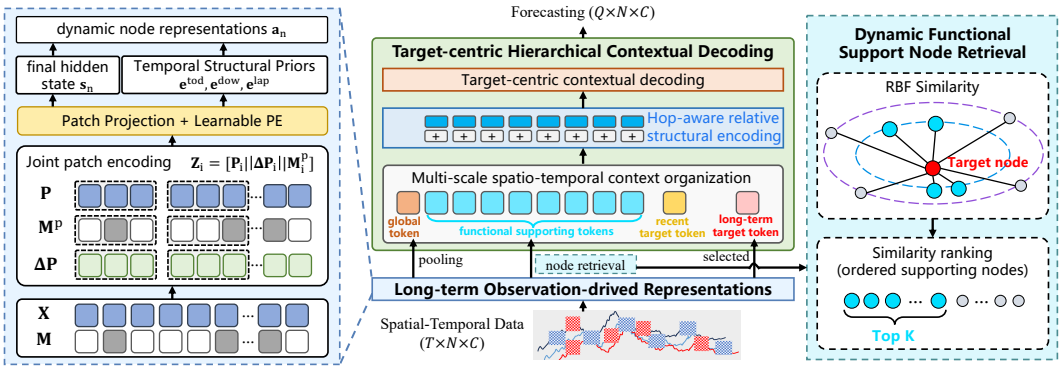


Figure 2: The architecture of VaryST. It contains Long-term Observation-derived Representations, Dynamic Functional Support Node Retrieval, and Target-centric Hierarchical Contextual Decoding.

4 METHODOLOGY

We propose VaryST, a compact forecasting framework trained from scratch that dynamically selects and organizes predictive evidence for varying deployment environments. *Long-Term Observation-Derived Representations* characterize each node's current state without fixed node-specific embeddings and guide *Dynamic Functional Support Node Retrieval*, allowing predictive evidence to adapt to changing spatial domains and temporal regimes. *Target-Centric Hierarchical Contextual Decoding* organizes recent and long-term target states, retrieved functional supports, and global information into a compact context spanning complementary temporal scales and spatial scopes, and further incorporates target-relative structural information through hop-aware encoding. Finally, *Importance-Guided Observation Perturbation* encourages the model to exploit alternative temporal and spatial evidence, improving robustness to varying observation availability.

4.1 LONG-TERM OBSERVATION-DERIVED REPRESENTATIONS

VaryST represents each node through its available long-term observations rather than a fixed node-specific embedding, enabling node representations to adapt to the current deployment environment.

Linear interpolation and temporal differences. First-order differences help characterize temporal fluctuations, but computing them directly from incomplete observations may introduce artificial changes around missing positions. Therefore, given observations $\mathbf{X}$ and availability mask $\mathbf{M}$, we first apply linear interpolation $\widetilde{\mathbf{X}} = \mathrm{Interp}(\mathbf{X}, \mathbf{M})$, and then compute $\Delta\widetilde{\mathbf{X}}_t = \widetilde{\mathbf{X}}_t - \widetilde{\mathbf{X}}_{t-1}$, initializing the first difference to zero.

Joint patch encoding. We partition $\widetilde{\mathbf{X}}$, $\Delta\widetilde{\mathbf{X}}$, and $\mathbf{M}$ into $T_p = \lceil T/L_p \rceil$ temporally aligned patches. For node v_n, the i-th patch representation is $\mathbf{z}_{n,i} = [\mathbf{P}_{n,i}||\Delta\mathbf{P}_{n,i}||\mathbf{M}^p_{n,i}]$, where $||$ denotes feature-wise concatenation. The patch sequence is encoded via a lightweight causal Transformer with learnable temporal order embedding $\mathbf{E}^{pos} \in \mathbb{R}^{T_p\times d}$ of patches:

$$\mathbf{H}_n = \mathrm{Enc}\left(\mathrm{Proj}(\mathbf{Z}_n) + \mathbf{E}^{pos}\right) \in \mathbb{R}^{T_p\times d}. \tag{2}$$

Its final hidden state $\mathbf{s}_n = \mathbf{H}_n[T_p] \in \mathbb{R}^d$ summarizes all preceding historical patches and serves as the dynamic node representation.

Temporal and structural priors. We further incorporate lightweight node-ID-free temporal and structural priors, including time-of-day, day-of-week, and Laplacian positional embeddings (Huang et al., 2025). The resulting dynamic node representation is

$$\mathbf{a}_n = \mathbf{s}_n + \mathbf{e}^{tod}_t + \mathbf{e}^{dow}_t + \mathbf{e}^{lap}_n. \tag{3}$$

Because each a_n is derived from observations rather than a learned node identity, the same encoder can construct representations for unseen nodes and update those of existing nodes as their operating regimes change. These dynamic node representations are then used for functional support retrieval and target-centric contextual decoding.

4.2 Dynamic Functional Support Node Retrieval

Predefined graph neighborhoods or spatial dependencies learned over fixed training nodes may become unreliable when the spatial domain changes, informative nodes lie outside the predefined neighborhood, or previously useful observations become unavailable. We therefore retrieve support nodes dynamically from their current representations rather than inheriting a fixed support set from training.

For each target node v_n, VaryST measures its affinity with every candidate node v_j using their dynamic representations $\mathbf{a}_n$ and $\mathbf{a}_j$. Specifically, we employ an RBF similarity:

$$\kappa_{n,j} = \exp\left(-\|\mathbf{a}_n - \mathbf{a}_j\|_2^2\right), \qquad j \neq n. \tag{4}$$

Because $\mathbf{a}_n$ captures the current observation-derived state together with temporal and structural priors, $\kappa_{n,j}$ reflects functional compatibility under the current context rather than predefined spatial proximity. We omit the RBF bandwidth because the similarity is used only for ranking, for which any fixed positive bandwidth preserves the same ordering.

The K nodes with the highest similarities form the functional supporting set of target node v_n:

$$\mathcal{S}_n = \mathrm{TopK}_{j \neq n}\left(\kappa_{n,j}\right) = \{v_{n,1}, \ldots, v_{n,K}\}. \tag{5}$$

The retrieved nodes are ordered by decreasing similarity, and their representations are subsequently used as supporting tokens in the target-centric decoder.

Unlike predefined graph neighborhoods, the retrieved support set adapts to current node representations and can change with temporal regimes, spatial domains, and observation availability, allowing VaryST to identify alternative predictive evidence when previously relied-upon nodes become less informative or unavailable. Since retrieval requires neither node-specific identifiers nor training-graph-specific support sets, it is naturally applicable to unseen graphs. Dynamic retrieval thus determines which nodes enter the predictive context, while the subsequent decoder determines how they influence the target.

4.3 Target-Centric Hierarchical Contextual Decoding

After retrieving functionally relevant support nodes, VaryST organizes predictive evidence into a target-centric hierarchical context spanning complementary temporal scales and spatial scopes.

Multi-scale spatio-temporal context organization. For a target node v_n, we first construct a global token by pooling the dynamic representations of all nodes:

$$\mathbf{g} = \mathrm{MeanPool}(\mathbf{a}_1, \ldots, \mathbf{a}_N) \in \mathbb{R}^d. \tag{6}$$

To complement the long-term target representation $\mathbf{a}_n$, we explicitly preserve its recent dynamics by projecting the latest R observations and first-order differences:

$$\mathbf{r}_n = \mathrm{MLP}_r\left([\widetilde{\mathbf{X}}_{t-R+1:t,n} \| \Delta\widetilde{\mathbf{X}}_{t-R+1:t,n}]\right). \tag{7}$$

Given the retrieved supporting set $\mathcal{S}_n$, the target-specific context is organized as:

$$\mathbf{C}_n^0 = [\underbrace{\mathbf{g}}_{\text{global token}} \| \underbrace{\mathbf{a}_{n,1}, \ldots, \mathbf{a}_{n,K}}_{\text{functional supporting tokens}} \| \underbrace{\mathbf{r}_n}_{\text{recent target token}} \| \underbrace{\mathbf{a}_n}_{\text{long-term target token}}]. \tag{8}$$

This hierarchy combines the target's immediate and long-term dynamics with functional spatial support and global system context, providing complementary evidence across temporal scales and spatial scopes.

Hop-aware relative structural encoding. Functional retrieval identifies support nodes according to their current functional similarity, but does not distinguish their structural relations to the target. To preserve such information, we encode the target-relative graph distance of each supporting node. For $v_{n,j} \in \mathcal{S}_n$, we compute its shortest-path hop distance to the target,

$$h_{n,j} = \mathrm{HopDist}(v_n, v_{n,j}), \tag{9}$$

and map the discretized distance to a learnable phase vector $\boldsymbol{\phi}_{h_{n,j}} \in \mathbb{R}^{d/2}$. We then apply a rotary transformation to the corresponding supporting token:

$$\bar{\mathbf{a}}_{n,j} = \mathbf{R}(\boldsymbol{\phi}_{h_{n,j}})\mathbf{a}_{n,j}, \qquad j = 1, \ldots, K, \tag{10}$$

where $\mathbf{R}(\cdot)$ applies a block-diagonal rotation to each pair of feature dimensions. Importantly, only the supporting-node tokens are structurally encoded; the global, recent-target, and long-term-target tokens remain unchanged. Since hop distance describes a target-relative relation rather than node identity or absolute graph position, the same encoding can be directly reused on unseen graphs, providing an inductive structural prior for cross-graph decoding.

Target-centric contextual decoding. The resulting context is processed by an L-layer lightweight Transformer decoder:

$$\mathbf{C}_n^L = \mathrm{Decoder}\left(\mathbf{C}_n^0\right). \tag{11}$$

We disable positional encoding in the decoder so that no sequential ordering is imposed on the heterogeneous context tokens; spatial structure is introduced exclusively through the target-relative hop encoding of support nodes. The long-term target token is placed at the final position, and its output state $\mathbf{o}_n = \mathbf{C}_n^L[-1]$ aggregates information from the global, functional-support, and recent-target contexts. Forecasts are produced as:

$$\widehat{\mathbf{Y}}_n = \mathrm{MLP}\left(\mathrm{LN}(\mathbf{o}_n)\right), \qquad \widehat{\mathbf{Y}}_n \in \mathbb{R}^{Q \times C}. \tag{12}$$

4.4 Training with Importance-Guided Observation Perturbation

To improve robustness to varying observation availability, we introduce Importance-Guided Observation Perturbation (IGOP) during training. Rather than masking observations randomly, IGOP deliberately removes historical patches on which the model relies most, creating harder, importance-guided perturbations that encourage the model to exploit alternative temporal and spatial evidence.

Patch Importance Estimation. Given the long-term patch representations of node v_n, $\mathbf{H}_n = [\mathbf{h}_{n,1}, \ldots, \mathbf{h}_{n,T_p}]$, we use the attention scores produced by the joint patch encoder to estimate patch importance. Let $\boldsymbol{\alpha}_n = [\alpha_{n,1}, \ldots, \alpha_{n,T_p}]$, where a larger $\alpha_{n,p}$ indicates stronger dependence on patch p. These scores are used only to guide perturbation construction.

Importance-Guided Observation Perturbation. For each node, IGOP selects the K_m patches with the highest attention-derived importance scores,

$$\mathcal{P}_n^{\mathrm{imp}} = \mathrm{TopK}_{p \in \{1,\ldots,T_p\}}\left(\alpha_{n,p}, K_m\right), \tag{13}$$

All observations within the selected patches are masked to obtain a perturbed observation mask $\mathbf{M}'$. Since each patch corresponds to a contiguous temporal interval, the perturbation naturally produces block-wise observation loss. The interpolated observations and temporal differences are then recomputed under $\mathbf{M}'$ to prevent masked values from leaking into subsequent representations.

Two-Stage Perturbation Training. For each mini-batch, a gradient-free probe first estimates patch importance and constructs $\mathbf{M}'$. VaryST is then optimized on the perturbed history with the original forecasting target:

$$\widehat{\mathbf{Y}} = f_\theta(\mathbf{X}, \mathbf{M}'), \qquad \mathcal{L}_{\mathrm{pred}} = \frac{1}{NQC}\left\|\widehat{\mathbf{Y}} - \mathbf{Y}\right\|_1. \tag{14}$$

The probe participates only in perturbation construction. No reconstruction objective is imposed on the masked patches; instead, forecasting supervision directly encourages the model to reorganize its reliance on the remaining temporal and spatial evidence.

5 Experiments

We evaluate the proposed VaryST on real-world traffic datasets to address the following research questions: **RQ1:** Does VaryST remain competitive under standard in-domain forecasting across different spatio-temporal systems? **RQ2:** Can VaryST remain effective under evolving temporal distributions? **RQ3:** Can VaryST generalize to unseen spatial environments? **RQ4:** Is VaryST robust to varying observation availability? **RQ5:** How do the key designs of VaryST contribute to its effectibeness? **RQ6:** How do model hyperparameters affect model performance? **RQ7:** What is the computational efficiency of VaryST?

5.1 EXPERIMENTAL SETUP

Table 1: Dataset statistics.

Dataset	Nodes	Time Span	Interval	Usage
PEMS03	358	09/01/2018 – 11/30/2018	5 min	RQ1/3/5/6
PEMS04	307	01/01/2018 – 02/28/2018	5 min	RQ1/3/4
PEMS07	883	05/01/2017 – 08/31/2017	5 min	RQ1/3/7
PEMS08	170	07/01/2016 – 08/31/2016	5 min	RQ1/3/4/5/6
PM2.5	12	03/01/2013 – 02/28/2017	60 min	RQ1
PEMSD3-Stream	655	2011 – 2017	5 min	RQ2

Dataset. Table 1 summarizes the benchmarks. PEMS03/04/07/08 cover in-domain and cross-network forecasting, with PEMS04/08 additionally evaluated under random (RM) and block (CM) missingness. PM2.5 evaluates non-traffic generalization, while the 7-year PEMSD3-Stream with an expanding network assesses long-term temporal shift. All benchmarks use a 6:2:2 chronological split to forecast 12 future steps from 2016 historical steps (168 for PM2.5), measured by MAE, RMSE, and MAPE.

Baselines. We compare VaryST with three groups of models for standard forecasting: classical spatio-temporal and time-series models, including ASTGCN (Guo et al., 2019), AGCRN (Bai et al., 2020), SSTBAN (Guo et al., 2023), PDFormer (Jiang et al., 2023), iTransformer (Liu et al., 2024c), and RAST (Ruan et al., 2026); pre-trained LLM-based models, including OFA (Zhou et al., 2023), STLLM (Liu et al., 2024a), STGLLM (Liu et al., 2024b), and STD-PLM (Huang et al., 2025); and extended-history models, where STID (Shao et al., 2022), iTransformer (Liu et al., 2024c), PatchSTG (Fang et al., 2025), BiST (Ma et al., 2025a), FaST (Zhao et al., 2026), and SparseTSF (Lin et al., 2026) are adapted to the same 2016-step history as VaryST. For temporal generalization, we compare with STOP (Ma et al., 2025b), which is specifically designed for temporal distribution shifts, and STD-PLM, which reports strong generalization forecasting ability. For spatial generalization, we compare with STD-PLM, the strongest baseline that supports cross-network transfer. For missing-observation forecasting, we select FaST, STID, and BiST for their strong standard-forecasting performance and efficiency, and use linear interpolation to pre-fill their missing inputs.

Settings. All experiments are conducted on a single NVIDIA RTX A4000 GPU under PyTorch 2.11. VaryST is trained from scratch with the MAE loss, Muon and AdamW optimizers with learning rate 0.001, batch size 128. Unless otherwise specified, we set the history length $T = 2016$, patch length $L_p = 36$, representation dimension $d = 96$, number of retrieved support nodes $K = 64$, recent window length $R = 12$, and decoder depth $L = 3$. For PM2.5, $T = 168$ and $L_p = 12$, while the other settings remain unchanged. Detailed settings are provided in Table 5.

5.2 STANDARD FORECASTING PERFORMANCE (RQ1)

Table 2 reports 12-step forecasting results on four PEMS benchmarks. Overall, long-history models outperform short-history baselines (e.g., PDFormer, RAST) and LLM-based methods (STD-PLM). VaryST achieves the best performance across all four datasets, reducing MAE by 3.16%–4.39%, RMSE by 1.71%–5.72%, and MAPE by 0.67%–8.61% over the strongest baselines. Its advantage over competitive long-history models (STID, FaST,BiST) demonstrates that simply feeding longer sequences is insufficient; VaryST crucially leverages long-term observations to derive dynamic node states and retrieve functional supports.

Table 3: Performance comparison on PM2.5.

Model	MAE	RMSE	MAPE
ASTGCN	27.42	48.70	52.4%
AGCRN	28.20	49.57	<u>50.1%</u>
SSTBAN	30.22	53.13	53.1%
PDFormer	<u>26.65</u>	**47.08**	51.5%
iTransformer	29.56	51.80	55.9%
STD-PLM	27.55	48.88	53.3%
STID†	31.22	54.88	64.1%
iTransformer†	29.97	50.52	58.5%
BiST†	28.11	47.28	54.5%
FaST†	30.66	51.99	59.2%
SparseTSF†	39.64	62.34	85.2%
VaryST	**26.44**	<u>47.18</u>	**49.91%**

†Models adapted to long-term inputs (168 steps).

Table 3 evaluates generalization beyond traffic on PM2.5, which features hourly observations across only 12 stations without an explicit graph. VaryST achieves the best MAE and MAPE alongside a competitive RMSE, demonstrating that dynamic target-centric context construction generalizes effectively to non-traffic domains.

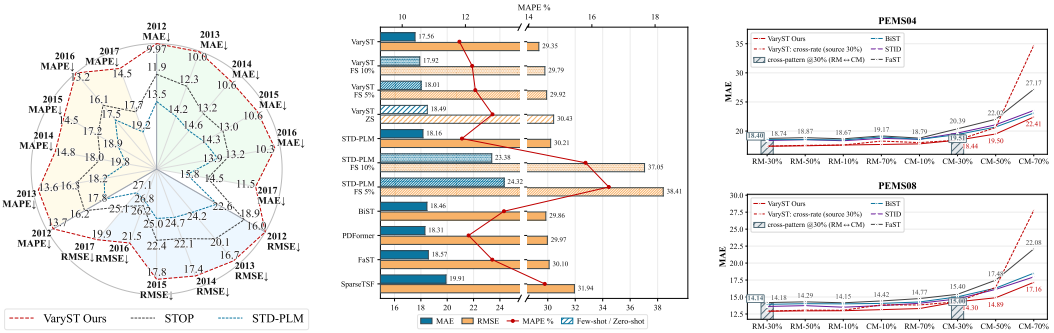
5.3 TEMPORAL GENERALIZATION (RQ2)

To evaluate temporal generalization, models are trained only on 2011 data of PEMSD3-Stream and tested across 2012–2017 without updates (zero-shot transfer). As shown in Figure 3a, VaryST

Table 2: Overall prediction performance comparison on four PEMS datasets. Models are categorized into short-term, LLM based and long-term groups. **Bold**: best, underline: second best.

Model	PEMS03			PEMS04			PEMS07			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Models with Short-term History (Input Horizon = 12)												
ASTGCN	17.85	29.88	17.65%	22.42	34.75	15.87%	25.98	39.65	11.84%	18.86	28.55	12.50%
AGCRN	15.98	28.25	15.23%	19.83	32.26	12.97%	22.37	36.55	9.12%	15.95	25.22	10.09%
SSTBAN	15.90	26.11	17.27%	18.88	31.10	12.56%	20.17	33.45	8.93%	14.39	24.19	10.06%
PDFormer	14.74	25.59	15.35%	18.31	29.97	12.10%	19.83	32.87	8.53%	13.58	23.51	9.05%
iTransformer	19.48	31.20	17.84%	22.56	35.21	16.29%	24.70	37.91	11.40%	20.05	31.90	11.99%
RAST	15.20	25.81	16.12%	18.39	29.93	12.43%	19.52	32.73	8.23%	14.16	23.33	9.27%
Pre-trained LLM-based Models												
OFA	20.96	33.43	19.11%	27.37	42.99	17.97%	30.53	47.51	12.98%	21.89	34.63	13.30%
STLLM	17.25	27.25	22.96%	19.00	30.35	13.55%	21.48	34.07	10.20%	14.67	23.50	10.63%
STGLLM	15.26	24.11	15.73%	20.00	32.11	13.69%	21.98	35.02	9.72%	15.53	24.74	10.15%
STD-PLM	14.59	25.36	14.92%	18.16	30.21	11.89%	19.25	32.84	8.06%	13.31	23.19	8.84%
Models with Long-term History												
STID†	15.56	25.37	15.25%	18.40	30.10	13.19%	19.11	31.71	8.45%	13.42	22.48	9.37%
iTransformer†	17.25	27.70	18.62%	20.82	32.79	14.42%	22.19	34.77	10.00%	15.31	23.94	10.61%
PatchSTG†	16.92	28.30	17.13%	18.67	30.44	13.10%	20.00	32.29	9.51%	13.73	23.03	9.58%
BiST†	15.01	24.64	14.87%	18.46	29.86	13.22%	19.55	32.53	8.87%	13.71	22.79	9.57%
FaST†	16.68	26.98	16.12%	18.57	30.10	12.85%	18.92	31.92	8.05%	14.22	23.21	10.31%
SparseTSF†	19.87	33.24	22.40%	19.91	31.94	14.51%	21.43	34.66	9.56%	16.12	25.74	11.52%
VaryST	**13.95**	**22.73**	**13.59%**	**17.56**	**29.35**	**11.81%**	**18.21**	**31.05**	**7.64%**	**12.89**	**21.99**	**8.50%**

† Models adapted to long-term inputs by us for fair comparison (2016 steps).



(a) Per-year zero-shot metrics on PEMSD3-Stream (2012–2017).

(b) Zero-shot vs. in-domain MAE on PEMS04.

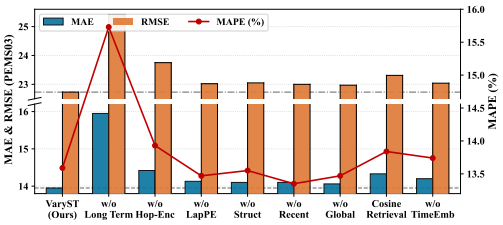
(c) MAE under RM and CM at 10–70% rates.

Figure 3: Distribution-shift robustness of VaryST.

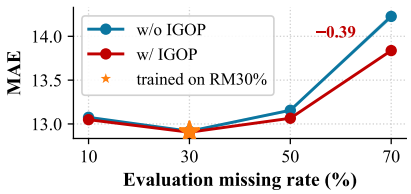
consistently outperforms STOP—the strongest temporal-robust baseline—across all years, reducing MAE by 16.22%–21.97%, RMSE by 15.34%–21.27%, and MAPE by 15.43%–18.08%. While STOP seeks invariant representations via distributionally robust optimization, VaryST adapts directly to evolving dynamics by deriving node states and dynamically retrieving functional supports from current long-term observations.

5.4 Spatial Generalization (RQ3)

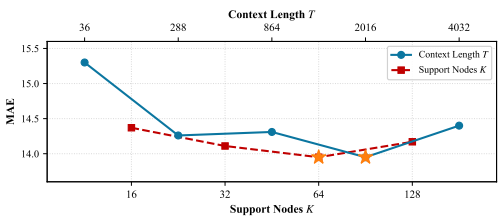
We evaluate spatial generalization via cross-network transfer across four PEMS benchmarks under zero-shot and few-shot (1%, 5%, 10% target data) settings (full 12 directions in Appendix C.2). Figure 3b highlights results on PEMS04. Under zero-shot transfer, VaryST degrades by only 3.68%/5.30%/8.89% (RMSE/MAE/MAPE) relative to its in-domain reference, whereas STD-PLM degrades by 52.36%/63.86%/91.33%. This gap stems from VaryST constructing representations and functional supports dynamically from current observations rather than static graph topology. With few-shot adaptation, VaryST with 5%–10% target data already surpasses competitive full-data in-domain models and consistently outperforms STD-PLM across all budgets.



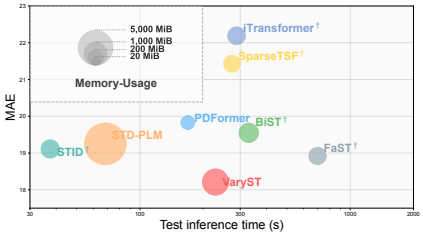
(a) Ablation study on PEMS03.



(b) IGOP ablation under missing-rate shift.



(c) Hyperparameter sensitivity on PEMS03.



(d) Efficiency on PEMS07.

Figure 4: Ablation, hyperparameter and efficiency studies.

5.5 Robustness to Observations Availability(RQ4)

We evaluate cross-network transfer across four PEMS benchmarks under zero-shot and few-shot (1%, 5%, 10%) settings (Appendix C.2). On PEMS04 (Figure 3b), zero-shot VaryST degrades by only 3.68%/5.30%/8.89% (RMSE/MAE/MAPE) relative to in-domain, whereas STD-PLM degrades by 52.36%/63.86%/91.33%, as VaryST dynamically constructs spatial supports from observations rather than static topology. With 5%–10% target data, VaryST already surpasses full-data in-domain baselines and outperforms STD-PLM across all budgets.

5.6 Ablation Studies (RQ5)

Ablation Study On PEMS03 (Figure 4a), removing long-term history increases MAE from 13.95 to 15.95, validating the role of long observations in dynamic node representations. For spatial context construction, replacing RBF similarity with cosine raises MAE to 14.33, while removing hop-aware encoding raises it to 14.42, justifying our affinity metric and target-relative structural design. For perturbation training, IGOP trained at a 30% missing rate reduces MAE by 0.39 at a 70% test missing rate on PEMS08 without retraining (Figure 4b). Full results are in Appendix C.4.

Hyperparameter Sensitivity We evaluate context length $T \in \{36, 288, 864, 2016, 4032\}$ and support count $K \in \{16, 32, 64, 128\}$ (Figure 4c). Extending history from 36 to the default 2016 steps reduces MAE by 8.8% on PEMS03 and 12.8% on PEMS08. For support retrieval, increasing K from 16 to the default 64 lowers MAE by 2.9% on PEMS03, confirming $K = 64$ as an optimal trade-off. Full results are in Appendix C.5.

5.7 Efficiency Analysis (RQ7)

Figure 4d evaluates inference efficiency on PEMS07 (883 nodes). VaryST achieves the lowest MAE (18.21), reducing error by 3.8% with faster inference than FaST†, while outperforming STD-PLM with substantially lower memory footprint. This efficiency stems from our compact predictive context: long-context encoding compresses each node's history into a single representation, limiting the decoder to process only $K + 3$ tokens per target.

6 Conclusion

This paper presents VaryST, which addresses generalizable spatio-temporal forecasting by dynamically constructing predictive contexts. Trained from scratch, VaryST demonstrates superior in-domain accuracy, robust zero- and few-shot generalization across spatial and temporal shifts, and resilience to varying observation availability.

AI USE STATEMENT

Generative AI tools were used only to assist with refining discussions, language polishing, and limited code-related support. All AI-assisted content was reviewed by the authors, who take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

The code for the experiments is included in the supplementary material. The training scripts, and the complete command-line configuration of every experiment are also provided. All datasets are public benchmarks (Table 1).

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17804–17815, 2020.

Yuchen Fang, Yuxuan Liang, Bo Hui, Zezhi Shao, Liwei Deng, Xu Liu, Xinke Jiang, and Kai Zheng. Efficient large-scale traffic forecasting with transformers: A spatial data management perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 307–317, 2025.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5415–5428, 2021.

Shengnan Guo, Youfang Lin, Letian Gong, Chenyu Wang, Zeyu Zhou, Zekai Shen, Yiheng Huang, and Huaiyu Wan. Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting. In *IEEE International Conference on Data Engineering*, pp. 1585–1596. IEEE, 2023.

Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. BigST: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks. *Proceedings of the VLDB Endowment*, 17(5):1081–1090, 2024. doi: 10.14778/3641204.3641217.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Yiheng Huang, Xiaowei Mao, Shengnan Guo, Yubin Chen, Junfeng Shen, Tiankuo Li, Youfang Lin, and Huaiyu Wan. STD-PLM: Understanding Both Spatial and Temporal Properties of Spatial-Temporal Data with PLM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11817–11825, 2025.

Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4365–4373, 2023.

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.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and C. L. Philip Chen. SparseTSF: Lightweight and Robust Time Series Forecasting via Sparse Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(1):170–183, 2026. doi: 10.1109/TPAMI.2025.3602445.

Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*, 2024a.

Lei Liu, Shuo Yu, Runze Wang, Zhenxun Ma, and Yanming Shen. How can large language models understand spatial-temporal data? *arXiv preprint arXiv:2401.14192*, 2024b.

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*, 2024c.

Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. BiST: A Lightweight and Efficient Bi-directional Model for Spatiotemporal Prediction. *Proceedings of the VLDB Endowment*, 18(6):1663–1676, 2025a.

Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. Robust spatio-temporal centralized interaction for ood learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 42165–42192, 2025b.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Weilin Ruan, Xilin Dang, Ziyu Zhou, Sisuo Lyu, and Yuxuan Liang. A retrieval augmented spatio-temporal framework for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 39163–39172, 2026.

Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun 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*, pp. 4454–4458, 2022.

Jiarui Sun, Yujie Fan, Chin-Chia Michael Yeh, Wei Zhang, and Girish Chowdhary. Revealing the power of masked autoencoders in traffic forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 4071–4075, 2024. doi: 10.1145/3627673.3679989.

Binwu Wang, Jiaming Ma, Pengkun Wang, Xu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang. Stone: A spatio-temporal ood learning framework kills both spatial and temporal shifts. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2948–2959, 2024.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019. doi: 10.24963/ijcai.2019/264.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018. doi: 10.24963/ijcai.2018/505.

Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. UniST: A Prompt-Empowered Universal Model for Urban Spatio-Temporal Prediction. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4095–4106, 2024.

Yiji Zhao, Zihao Zhong, Ao Wang, Haomin Wen, Ming Jin, Yuxuan Liang, Huaiyu Wan, and Hao Wu. FaST: Efficient and Effective Long-Horizon Forecasting for Large-Scale Spatial-Temporal Graphs via Mixture-of-Experts. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1975–1986, 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, 2023.

APPENDIX

Roadmap. This appendix provides supplementary methodology, comprehensive experimental setups, and complete empirical results. Specifically, Appendix A.1 presents the pseudocode for the training and inference pipelines of VaryST, along with a complexity analysis of each component. Appendix B elaborates on dataset statistics, hyperparameter configurations, evaluation metrics, and implementation details. Due to space limitations in the main paper, Appendix C details the full experimental results, covering temporal and spatial generalization, missing-observation forecasting under rate and pattern shifts, ablation studies, and hyperparameter sensitivity analyses.

A ALGORITHMS

A.1 PSEUDO CODE

Algorithm 1 VaryST training and inference. Lines 3–6 run only during training; at inference the forward pass executes once with $\mathbf{M}' = \mathbf{M}$.

Require: history $\mathbf{X}_{t-T+1:t}$, mask $\mathbf{M}$, graph $\mathcal{G}$, horizon Q; patch length L_p, number of supports K, recent steps R, masked patches K_m

Ensure: prediction $\widehat{\mathbf{Y}}_{t+1:t+Q}$

// Stage 1: importance-guided probe (training only)

1: $\mathbf{M}' \leftarrow \mathbf{M}$

2: **if** training **then**

3: $\quad \boldsymbol{\alpha}_n \leftarrow$ patch-attention scores of the long-term encoder ▷ gradient-free probe

4: $\quad \mathcal{P}_n^{\mathrm{imp}} \leftarrow \mathrm{TopK}(\boldsymbol{\alpha}_n, K_m)$; $\quad \mathbf{M}' \leftarrow \mathrm{Mask}(\mathbf{M}, \mathcal{P}^{\mathrm{imp}})$

5: **end if**

// Stage 2: long-term dynamic representations

6: $\widetilde{\mathbf{X}} \leftarrow \mathrm{Interp}(\mathbf{X}, \mathbf{M}')$; $\quad \Delta\widetilde{\mathbf{X}} \leftarrow$ first-order differences of $\widetilde{\mathbf{X}}$

7: **for** $n = 1, \ldots, N$ **do**

8: $\quad \mathbf{Z}_n \leftarrow [\mathrm{Patch}(\widetilde{\mathbf{X}}_n) \,\|\, \mathrm{Patch}(\Delta\widetilde{\mathbf{X}}_n) \,\|\, \mathrm{Patch}(\mathbf{M}'_n)]$

9: $\quad \mathbf{a}_n \leftarrow \mathrm{Enc}(\mathrm{Proj}(\mathbf{Z}_n) + \mathbf{E}^{pos})[T_p] + \mathbf{e}_t^{tod} + \mathbf{e}_t^{dow} + \mathbf{e}_n^{lap}$

10: **end for**

// Stage 3: functional support retrieval and target-centric decoding

11: $\mathbf{g} \leftarrow \mathrm{MeanPool}(\mathbf{a}_1, \ldots, \mathbf{a}_N)$

12: **for** each target node v_n **do**

13: $\quad \mathcal{S}_n \leftarrow \mathrm{TopK}_{j \neq n}\big(\exp(-\|\mathbf{a}_n - \mathbf{a}_j\|_2^2)\big)$; $\quad \mathbf{r}_n \leftarrow \mathrm{MLP}_r\big([\widetilde{\mathbf{X}}_{t-R+1:t,n} \,\|\, \Delta\widetilde{\mathbf{X}}_{t-R+1:t,n}]\big)$

14: $\quad \mathbf{C}_n^0 \leftarrow [\mathbf{g} \,\|\, \mathbf{a}_{n,1}, \ldots, \mathbf{a}_{n,K} \,\|\, \mathbf{r}_n \,\|\, \mathbf{a}_n]$

15: $\quad \mathbf{C}_n^L \leftarrow \mathrm{Decoder}(\mathbf{C}_n^0)$ with rotated keys $\widetilde{\mathbf{k}}_{n,j} = \mathbf{R}(\boldsymbol{\varphi}_{h_{n,j}})\mathbf{k}_j$, $v_j \in \mathcal{S}_n$

16: $\quad \mathbf{o}_n \leftarrow \mathbf{C}_n^L[-1]$; $\quad \widehat{\mathbf{Y}}_n \leftarrow \mathrm{MLP}(\mathrm{LN}(\mathbf{o}_n))$

17: **end for**

// Stage 4: forecasting loss

18: **if** training **then**

19: $\quad \theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{pred}}$, $\quad \mathcal{L}_{\mathrm{pred}} = \frac{1}{NQC}\big\|\widehat{\mathbf{Y}} - \mathbf{Y}\big\|_1$

20: **end if**

A.2 COMPLEXITY ANALYSIS

Notation. Let N be the number of nodes, B the number of sliding windows per training iteration, and $P = BN$ the number of target instances (one per window–node pair). Let T and C denote the input length and the number of input features, u the unit-window length and $U = T/u$ the number of unit windows, d the token width of the temporal encoder, D the backbone width, H the number of backbone layers, and K the number of retrieved support nodes. With the global token and the recent prompt enabled, each target instance is decoded from a sequence of length $\ell = K + 3$.

Temporal encoding. Each window-to-node pair is encoded independently: the unit projection (two-layer MLP over each unit window) costs $O(TCd)$ per pair, the self-attention over the U

unit tokens costs $O(U^2d + Ud^2)$, the single-query readout attention costs $O(Ud^2)$, and the output projection costs $O(dD)$. With U fixed (U=56 in our experiments), the total per iteration is $O\big(BN\,(TCd + Ud^2)\big)$, linear in B, N, and T.

Support Node retrieval. For every target instance, similarity to all N candidate nodes of its window is one D-dimensional inner product, followed by a top-K selection, giving $O(BN^2D)$ per iteration. This is the only step super-linear in N. It is parameter-free and lies outside the gradient graph: candidate keys are detached, and the top-K indices serve only as gather indices, so the step incurs no backward cost. Gradients reach the retrieved nodes through their *values*, gathered from the gradient-carrying node states.

Context assembly and decoding. Gathering the K support tokens, the additive structural and temporal encodings, and the recent prompt token cost $O(BNKD)$ in total. The backbone then processes each target instance as a sequence of length $\ell = K + O(1)$, *independent of* N: a Transformer layer costs $O(\ell^2D + \ell D^2)$ per instance, and the hop-bucket key rotation multiplies the attention-score computation only by the constant number of buckets (10). Over all $P = BN$ instances and H layers, decoding costs $O\big(BN \cdot H\,(\ell^2D + \ell D^2)\big)$. Graph-dependent quantities (Laplacian eigenvectors, hop buckets, and the hop matrix) are computed once per graph, outside the training loop.

Where the quadratic term lives. A graph Transformer attending over all N nodes pays $O(N^2D)$ *inside every backbone layer*, in both the forward and the backward pass, and materialises $N \times N$ attention scores in memory. In Methodology, the quadratic term appears exactly once, in the retrieval matmul above: a single parameter-free, gradient-free matrix product per iteration. All gradient-carrying computation—temporal encoding and the backbone—scales linearly with N at fixed ℓ, and the attention footprint per instance is ℓ^2 rather than N^2.

B Experiment Details

B.1 Dataset Statistics

All six datasets are public benchmarks whose statistics are summarized in Table 1 of the main text. The traffic datasets span 170 to 883 nodes at 5-minute sampling. The air quality dataset PM2.5 has 12 stations at hourly sampling.

PEMSD3-Stream protocol. Per-year statistics are listed in Table 4. We follow the PEMSD3-Stream setting of STOP (Ma et al., 2025b): PeMS District 3 flow over 2011–2017, one slice per year, and each year an independent graph whose station count grows from 655 to 871 as new sensors join. Training and validation use 2011 alone with the window indices released by STOP, the validation split being disjoint from training and used for early stopping, so no window of a later year is seen before testing. Testing uses the held-out windows of a year: 2011 gives the in-domain reference, while 2012–2017 are evaluated zero-shot with the 2011 checkpoint, without retraining. As in STOP, raw flow of zero marks a missing station and is excluded from the loss and the metrics, so all reported numbers are computed on observed positions.

Table 4: Per-year statistics of PEMSD3-Stream.

Year	Nodes	Time Span	Interval
2011	655	07/10/2011 – 08/31/2011	5 min
2012	715	07/10/2012 – 08/31/2012	5 min
2013	786	07/10/2013 – 08/31/2013	5 min
2014	822	07/10/2014 – 08/31/2014	5 min
2015	834	07/10/2015 – 08/31/2015	5 min
2016	850	07/10/2016 – 08/31/2016	5 min
2017	871	07/10/2017 – 08/31/2017	5 min

B.2 HYPERPARAMETERS

Table 5: Default hyperparameters for the VaryST framework.

Hyperparameter	Description	Value or Choices
sample_len	Input history length T	2016 (168 on PM2.5)
output_len	Prediction horizon Q	12
unit_len	Patch length L_p	36 (12 on PM2.5)
train_ratio / val_ratio	Chronological train/validation split	0.6 / 0.2
token_dim	Temporal Patch Token dimension d	96
emb_dim	backbone dim D	96
n_head	backbone attention heads	4
backbone_layers	Number of backbone layers	3
spatial_pe	Structural encodings	lap, hop_rope
node_pe_k	Laplacian PE eigenvectors	16
n_aux	Number of support nodes retrieved per target K	64
use_time_emb	Time-of-day / day-of-week embeddings	True
use_global_token	Global mean-pooled token	True
recent_len	Recent steps in the prompt token	12
norm_mode	Input normalization	RevIN
loss_type	Training loss	MAE
optimizer	Optimizer	Muon + AdamW
lr	Learning rate	0.001
weight_decay	Weight decay	0.0
dropout	Dropout	0.05
batch_size	The batch size for training	128
lr_patience	Learning-rate decay patience	5
patience	Early-stopping patience	11
epoch	Maximum training epochs	500

Table 5 lists the default configuration shared by all PEMS experiments. On PM2.5, the history length and the patch length, are changed to 168 and 12 respectively, as marked in the table. The sensitivity of context length, patch length, and the number of retrieved support nodes is studied in Section C.5.

B.3 METRICS

Throughout this paper, we use the following metrics to evaluate performance:

MAE (Mean Absolute Error): Represents the average absolute difference between the predicted and actual values, providing an interpretable measure of average error magnitude.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{15}$$

where y_i and $\hat{y}_i$ denote the ground truth and predicted values, respectively, and n is the number of evaluated data points.

RMSE (Root Mean Squared Error): The square root of the mean squared difference between the predicted and actual values, expressed in the same units as the original data, and more sensitive to large deviations than MAE.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{16}$$

MAPE (Mean Absolute Percentage Error): Expresses errors as a percentage of the actual values, offering a scale-independent metric that facilitates comparisons across datasets.

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{17}$$

All three metrics are computed only at observed positions, and reported values are averaged over all 12 prediction horizons; MAPE is expressed in percent. For all metrics, lower values indicate better performance.

B.4 Implementation Details

B.4.1 Hardware and Environment

All experiments run on a single NVIDIA RTX A4000 GPU under PyTorch 2.11. With early stopping, training completes in 3.5 hours on PEMS03, 4.2 hours on PEMS04, 4.4 hours on PEMS07, 3.8 hours on PEMS08, 2.7 hours on the PEMSD3-Stream train year, and 44 minutes on PM2.5, where the wall-clock covers the full training loop including validation and test passes. Peak training memory stays within 0.9–2.9 GB on the traffic benchmarks and below 0.1 GB on PM2.5.

B.4.2 Data Splitting and Leakage Prevention

Every dataset is split chronologically into training, validation, and test portions in a 6:2:2 ratio, with no shuffling across time. Zero-shot evaluation applies a model trained fully on the source dataset directly to the target dataset, without updating target parameters. Few-shot transfer fine-tunes on 1%, 5%, or 10% of the target training data, matching Table 9. The fraction is taken over training windows, each retaining the full 2016-step input that ends exactly where its 12-step target begins, rather than over the raw time span, which for our splits is shorter than one window and would leave nothing to train on; the retained windows are evenly spaced across the training portion, so a 1% run performs roughly 1% of the parameter updates of a full run. The same 2016-step input length also fixes the evaluation span: because a window's target can only start where its input ends, the validation and test boundaries are shifted back by 2016 steps, so that the targets of the test windows tile the entire test portion from its first step to its last and every test step is scored. A model that under- or over-fits the windows it was fine-tuned on is therefore judged on the whole test span, never on a span as narrow as its few-shot subset. VaryST carries no node identity: representations are derived online from current observations, and its structural priors are relative relations only, namely hop distances and Laplacian eigenvectors. No entity information from a training graph therefore survives transfer. In the missing-observation experiments the input mask coincides with the loss mask, so the loss is computed only at observed positions; see the next subsubsection.

B.4.3 Baseline Adaptation and Missing-Input Protocol

Dagger†-marked baselines were adapted by us to the same long-term input as VaryST, namely 2016 steps on the traffic benchmarks, while the PM2.5 comparison uses the 168-step input as stated in its table footnote. These adaptations change only the input interface, with no architectural modification. PatchSTG additionally requires sensor latitude and longitude metadata to construct its geospatial partitioning, which the PEMS benchmarks do not provide; we therefore synthesize virtual coordinates for each dataset so that PatchSTG operates with its original geo-partitioning intact. Under missing observations the baselines do not consume observation masks, so their inputs are pre-filled by linear interpolation. VaryST instead consumes the mask natively in its encoder, so its representations are conditioned on which observations are available.

C Full Experiment Results

C.1 Full Temporal Generalization Results

Models are trained and validated on 2011 alone and evaluated zero-shot on each year from 2012 to 2017 as new sensors join the network; metrics are computed only at observed positions and averaged over all 12 prediction horizons. The six-year averages are summarized in Table 6 and the per-year breakdown in Table 7. VaryST attains the best MAE, RMSE, and MAPE in every year, reducing the six-year average MAE from 13.01 to 10.49. VaryST improves over STOP, the strongest baseline, by 19.4%. The gap follows from observation-derived node representations recomputed from current observations, and from retrieval that reselects supporting evidence as temporal patterns drift. The per-year curves appear in Figure 3a of the main text.

Table 6: Average Zero-shot performance on PEMSD3-Stream.

Model	MAE	RMSE	MAPE
STOP	13.01	22.44	16.91%
STD-PLM	14.40	25.06	18.55%
VaryST	**10.49**	**18.22**	**14.05%**

Table 7: Year-by-year performance on PEMSD3-Stream. 2011: In-Domain Reference

Year	**VaryST**			**STOP**			**STD-PLM**		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
2011	**10.64**	**16.47**	**13.84**	12.34	18.96	16.12	11.96	18.28	16.21
2012	**9.97**	**16.00**	**13.65**	11.86	18.87	16.16	13.54	22.57	17.77
2013	**10.01**	**16.68**	**13.60**	12.27	20.05	16.26	14.24	24.16	18.23
2014	**10.61**	**17.39**	**14.80**	13.24	22.07	17.96	14.62	24.73	19.78
2015	**10.55**	**17.80**	**14.53**	12.99	22.40	17.22	14.32	24.97	18.87
2016	**10.30**	**21.52**	**13.23**	13.19	26.20	16.14	13.89	26.77	17.46
2017	**11.51**	**19.93**	**14.50**	14.51	25.05	17.70	15.81	27.15	19.18

C.2 Full Spatial Generalization Results

Table 8 presents 12-step zero-shot transfer performance across all twelve pairwise source–target combinations among PEMS03, PEMS04, PEMS07, and PEMS08, evaluated in MAE, RMSE, and percentage MAPE, with the diagonal serving as the in-domain baseline. In parallel, Table 9 reports few-shot fine-tuning results using 1%, 5%, and 10% of the target training data. VaryST achieves the lowest MAE, RMSE, and MAPE across all twelve zero-shot transfer directions. Notably, its maximum MAE increase over the corresponding in-domain baseline is confined to just 9.9% in the PEMS03→PEMS07 transfer. In fact, transferring from PEMS07 to PEMS03 slightly surpasses the in-domain benchmark, achieving an MAE of 13.78 compared to 13.95. This counter-intuitive advantage aligns with our design: node representations and supports are directly constructed from observations available on the target network rather than rigid source graph structures. Under few-shot adaptation, fine-tuning VaryST on merely 1% of target data already outperforms STD-PLM adapted with 10% data across all twelve directions, while scaling the target budget to 10% yields a modest additional MAE reduction of at most 3.7%. Figure 3b compares zero-shot and in-domain MAE on PEMS04, with comprehensive analyses provided in Section 5.4.

Table 8: Zero-shot cross-dataset transfer (12-step prediction). Rows: source dataset used for full training; columns: target dataset evaluated on. Diagonal cells are the in-domain (IID) baselines. Each cell: MAE / RMSE / MAPE (%).

Model	**Source**	**Target**			
		PEMS03	**PEMS04**	**PEMS07**	**PEMS08**
VaryST	PEMS03	13.95 / 22.73 / 13.59	18.51 / 30.40 / 12.84	20.02 / 32.72 / 8.88	13.81 / 22.70 / 9.31
	PEMS04	14.03 / 22.77 / 13.44	17.56 / 29.35 / 11.81	19.91 / 32.64 / 8.48	13.97 / 22.92 / 9.07
	PEMS07	**13.78 / 22.32 / 13.24**	**18.15 / 29.98 / 12.54**	18.21 / 31.05 / 7.64	**13.46 / 22.44 / 8.86**
	PEMS08	14.88 / 24.45 / 14.61	18.80 / 30.90 / 13.19	**19.53 / 32.65 / 8.54**	12.89 / 21.99 / 8.50
STD-PLM	PEMS03	14.52 / 24.69 / 14.87	30.90 / 49.01 / 20.88	34.79 / 52.85 / 20.27	25.83 / 39.86 / 25.03
	PEMS04	23.90 / 38.93 / 27.38	18.15 / 30.31 / 11.77	35.19 / 53.08 / 23.32	26.64 / 40.38 / 28.48
	PEMS07	24.98 / 38.37 / 42.77	29.44 / 44.95 / 24.05	19.26 / 32.72 / 8.09	22.86 / 35.28 / 15.09
	PEMS08	22.48 / 36.65 / 36.09	28.89 / 44.59 / 22.63	31.25 / 48.41 / 14.18	13.30 / 23.41 / 8.82

Table 9: Few-shot cross-dataset transfer (12-step prediction) with 1%, 5% and 10% of the target training data. Each cell: MAE / RMSE / MAPE (%).

Model	Target	1%			5%			10%		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
Source: PEMS03										
VaryST	PEMS04	**18.39**	**30.39**	**12.60**	**18.04**	**29.93**	**12.33**	**17.91**	**29.78**	**12.19**
	PEMS07	**19.16**	**32.03**	**8.16**	**18.60**	**31.48**	**7.85**	**18.63**	**31.50**	**7.85**
	PEMS08	**13.50**	**22.54**	**8.91**	**13.26**	**22.31**	**8.75**	**13.17**	**22.21**	**8.69**
STD-PLM	PEMS04	46.75	74.69	22.00	24.45	38.70	16.19	23.38	37.08	15.54
	PEMS07	28.35	43.93	12.53	25.53	40.06	11.58	24.34	38.77	10.55
	PEMS08	24.00	36.80	16.20	18.68	29.55	14.00	18.16	29.02	12.17
Source: PEMS04										
VaryST	PEMS03	**14.08**	**22.91**	**13.49**	**14.00**	**22.77**	**13.35**	**13.98**	**22.73**	**13.45**
	PEMS07	**19.27**	**32.14**	**8.11**	**18.74**	**31.62**	**7.86**	**18.65**	**31.49**	**7.82**
	PEMS08	**13.64**	**22.69**	**8.77**	**13.34**	**22.36**	**8.65**	**13.20**	**22.21**	**8.59**
STD-PLM	PEMS03	20.30	32.79	21.97	18.35	30.37	18.00	17.01	27.84	16.67
	PEMS07	27.60	42.84	12.58	25.17	39.70	11.35	24.55	39.07	10.72
	PEMS08	21.99	33.25	15.84	18.47	29.40	12.70	18.05	28.96	11.80
Source: PEMS07										
VaryST	PEMS03	**13.91**	**22.56**	**13.31**	**13.89**	**22.54**	**13.24**	**13.88**	**22.55**	**13.18**
	PEMS04	**18.20**	**30.12**	**12.50**	**17.88**	**29.75**	**12.15**	**17.87**	**29.72**	**12.18**
	PEMS08	**13.27**	**22.31**	**8.77**	**13.09**	**22.18**	**8.60**	**13.02**	**22.09**	**8.62**
STD-PLM	PEMS03	20.41	33.44	26.46	18.35	30.69	18.23	16.86	27.78	16.38
	PEMS04	32.67	53.73	20.74	24.07	38.02	16.76	23.19	36.79	15.88
	PEMS08	21.78	33.36	18.84	18.18	29.20	11.53	18.08	29.17	11.37
Source: PEMS08										
VaryST	PEMS03	**14.78**	**24.57**	**14.41**	**14.56**	**24.10**	**14.21**	**14.57**	**23.95**	**14.16**
	PEMS04	**18.66**	**30.98**	**12.84**	**18.12**	**30.07**	**12.46**	**17.97**	**29.87**	**12.27**
	PEMS07	**19.01**	**32.15**	**8.08**	**18.70**	**31.83**	**7.89**	**18.51**	**31.53**	**7.81**
STD-PLM	PEMS03	19.69	32.93	23.70	18.38	31.25	17.96	16.91	28.28	16.31
	PEMS04	47.90	84.18	23.60	24.44	38.50	16.66	23.57	37.27	15.98
	PEMS07	27.63	42.92	12.31	25.38	39.94	11.14	24.38	38.94	10.59

C.3 Full Missing-Observation Results

The evaluation covers forecasting on PEMS04 and PEMS08 under point missing (RM) and spatio-temporal block missing (CM) at missing rates of 10% to 70%, with all metrics averaged over the 12 horizons at observed positions. VaryST trains one model per rate with importance-guided observation perturbation (IGOP), which removes the history patches that draw the most attention during training, whereas baseline inputs are pre-filled by linear interpolation, following Appendix B.4.3. Table 10 reports same-rate training and testing.

Table 10: Overall prediction performance with missing observations on PEMS04 and PEMS08 under missing rates from 10% to 70%. The best and second-best results are in **bold** and underline, respectively.

Model	10%			30%			50%			70%		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
PEMS04 Point Missing (RM)												
FaST†	18.67	30.29	12.89	18.74	30.37	13.01	18.87	30.50	13.05	19.17	30.78	13.14
STID†	18.34	29.92	13.09	18.38	29.98	12.84	18.53	30.20	12.99	18.75	30.40	13.41
BiST†	18.47	29.94	12.88	18.53	29.94	13.25	18.53	30.10	13.22	19.00	30.42	13.69
VaryST	**17.60**	**29.34**	**11.90**	**17.40**	**29.20**	**11.74**	**17.47**	**29.28**	**11.80**	**17.70**	**29.39**	**11.93**
PEMS04 Spatio-Temporal Block Missing (CM)												
FaST†	18.79	30.32	13.07	20.39	32.74	14.16	22.02	34.70	15.99	27.17	41.55	20.67
STID†	18.59	30.22	13.48	19.66	32.20	13.84	21.08	33.92	14.88	23.54	38.39	16.08
BiST†	18.58	30.13	13.51	19.40	31.39	13.80	20.70	33.06	14.87	23.09	36.81	17.41
VaryST	**17.80**	**29.55**	**12.05**	**18.44**	**30.42**	**12.48**	**19.50**	**31.76**	**13.39**	**22.41**	**36.08**	**15.29**
PEMS08 Point Missing (RM)												
FaST†	14.15	23.26	10.08	14.18	23.36	10.14	14.29	23.65	10.18	14.77	23.97	10.47
STID†	13.50	22.50	9.34	13.61	22.53	9.73	13.73	22.62	9.60	14.08	23.12	10.12
BiST†	13.99	22.74	10.75	13.92	22.77	10.59	14.02	22.94	10.33	14.27	23.24	10.53
VaryST	**12.98**	**22.13**	**8.61**	**12.91**	**21.91**	**8.56**	**12.96**	**21.96**	**8.55**	**13.30**	**22.27**	**8.73**
PEMS08 Spatio-Temporal Block Missing (CM)												
FaST†	14.42	23.75	10.40	15.40	25.00	11.00	17.48	27.69	12.93	22.08	33.78	17.09
STID†	13.74	22.80	9.60	14.73	23.83	10.35	16.12	25.91	12.12	17.96	28.94	12.71
BiST†	14.06	23.12	10.54	15.03	24.14	10.83	16.31	26.19	11.80	18.51	29.11	14.22
VaryST	**13.14**	**22.32**	**8.63**	**14.30**	**23.38**	**9.63**	**14.89**	**24.67**	**9.72**	**17.16**	**28.08**	**11.56**

† Models adapted to long-context inputs (2016 steps) by us for fair comparison.

Under this protocol VaryST attains the lowest metric in all 48 reported cells. Scattered point missing barely affects it: between 10% and 70% RM the same-rate MAE stays within 0.4 on both datasets, because the encoder consumes the availability mask directly and re-derives node representations from the observations that remain. Block missing is the harder regime: as entire segments vanish, the error of every model grows with the rate, and VaryST remains first throughout.

Table 11: Generalization to unseen missing rates on PEMS04 and PEMS08. Models are trained at a 30% missing rate.

Model	10%			30%			50%			70%		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
PEMS04 Point Missing (RM), source = 30%												
FaST†	18.69	30.31	12.97	18.74	30.37	13.01	18.88	30.59	13.09	19.22	30.97	13.39
STID†	18.32	29.91	12.80	18.38	29.98	12.84	18.53	30.19	12.91	18.91	30.62	13.19
BiST†	18.49	29.90	13.22	18.53	29.94	13.25	18.66	30.12	13.32	18.98	**30.12**	13.57
VaryST	**17.60**	**29.35**	**12.07**	**17.40**	**29.20**	**11.74**	**17.55**	**29.38**	**11.73**	**18.28**	**30.12**	**12.34**
PEMS04 Spatio-Temporal Block Missing (CM), source = 30%												
FaST†	19.88	32.16	13.63	20.39	32.74	14.16	23.37	36.43	18.95	38.20	57.56	39.79
STID†	19.07	31.42	13.21	19.66	32.20	13.84	22.27	35.67	17.73	34.73	52.86	35.44
BiST†	18.98	30.86	13.42	19.40	31.39	13.80	21.73	34.31	17.31	34.83	**52.23**	36.97
VaryST	**18.02**	**29.91**	**12.35**	**18.44**	**30.42**	**12.48**	**20.61**	**34.15**	**14.31**	**34.71**	58.39	**32.20**
PEMS08 Point Missing (RM), source = 30%												
FaST†	14.11	23.28	10.09	14.18	23.36	10.14	14.29	23.50	10.22	14.68	24.00	10.43
STID†	13.55	22.47	9.70	13.61	22.53	9.73	13.76	22.71	9.82	14.18	23.26	10.02
BiST†	13.88	22.74	10.58	13.92	22.77	10.59	14.03	22.90	10.68	14.42	23.44	10.88
VaryST	**13.05**	**22.05**	**8.73**	**12.91**	**21.91**	**8.56**	**13.07**	**22.05**	**8.62**	**13.83**	**22.87**	**9.02**
PEMS08 Spatio-Temporal Block Missing (CM), source = 30%												
FaST†	14.73	24.16	10.47	15.40	25.00	11.00	18.57	29.78	13.30	32.06	48.39	26.39
STID†	14.05	23.00	9.82	14.73	23.83	10.35	17.31	28.23	12.18	28.72	46.28	21.98
BiST†	14.44	23.40	10.34	15.03	24.14	10.83	17.30	27.90	12.53	**27.43**	**43.88**	**21.04**
VaryST	**13.84**	**22.83**	**9.21**	**14.30**	**23.38**	**9.63**	**16.42**	**27.82**	**11.18**	27.76	48.80	21.84

† Models adapted to long-context inputs (2016 steps) by us for fair comparison.

Table 11 reports cross-rate transfer performance, where models trained at a 30% missing rate are directly evaluated across rates from 10% to 70% without retraining or altering the missing pattern. Under RM, VaryST ranks first across all 24 evaluation cells. Even at the extreme 70% missing rate, its MAE remains 0.63 and 0.35 lower than the runner-up on PEMS04 and PEMS08, respectively. Although its degradation relative to same-rate training reaches +0.58 on PEMS04 and +0.53 on PEMS08, surpassing the baselines' maximum gap of 0.16, scattered observations still suffice to reconstruct its representations. Under CM, VaryST maintains its top ranking up to a 50% missing rate; at 70%, where contiguous evidence largely vanishes, it still achieves the best MAE and MAPE on PEMS04, outperforming the runner-up by 3.2 points in MAPE. On PEMS08 it is second to BiST on MAE, 27.76 against 27.43, and it is last on RMSE on both datasets.

Table 12: Cross-pattern generalization at 30% missing rate on PEMS04 and PEMS08.

Model	PEMS04						PEMS08					
	RM → CM			CM → RM			RM → CM			CM → RM		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
FaST†	20.32	32.40	15.19	19.89	32.16	13.62	16.00	26.00	11.87	14.71	24.13	10.46
STID†	19.96	32.14	14.71	19.06	31.41	13.13	15.33	**24.85**	11.28	**14.02**	**22.95**	9.80
BiST†	20.02	**31.87**	15.06	18.98	30.86	13.38	15.63	25.06	12.33	14.42	23.37	10.31
VaryST	**19.51**	31.94	**13.96**	**18.40**	**30.23**	**12.75**	**15.00**	24.93	**10.43**	14.14	23.15	**9.36**
In-domain reference at 30% (train and test on the target pattern)												
STID†	19.66	32.20	13.84	18.38	29.98	12.84	14.73	23.83	10.35	13.61	22.53	9.73
BiST†	19.40	31.39	13.80	18.53	29.94	13.25	15.03	24.14	10.83	13.92	22.77	10.59

† Models adapted to long-context inputs (2016 steps) by us for fair comparison.

Table 12 reports cross-pattern transfer at a 30% missing rate: a model trained under one pattern is evaluated under the other without further training, and the reference rows repeat the in-domain numbers at 30% from Table 10. VaryST attains the lowest MAE in three of the four directions and the lowest MAPE in all four, as its representations are re-derived from the observations that survive

the switch. Its RMSE is second in both RM→CM directions by at most 0.08, and STID leads MAE and RMSE in the CM→RM direction on PEMS08 (14.14 vs. 14.02 and 23.15 vs. 22.95), consistent with STID modeling each node independently and being least sensitive to the pattern switch.

The rate and pattern sweeps are summarized in Figure 3c of the main text.

C.4 Full Ablation Results

The complete ablation in Table 13 covers 12-step prediction on PEMS03 and PEMS08, reporting MAE, RMSE, MAPE, and MAPE_{10}, where MAPE_{10} excludes ground-truth values below 10. Removing any component raises the MAE and RMSE on PEMS03. The raw MAPE deserves care. Readings below 10 carry little operational value in traffic applications, but their small denominators make relative errors fluctuate, so a few variants look marginally better under this metric. Once these low-value readings are excluded, the full model attains the best MAPE_{10} on both datasets, and their apparent advantages prove to be small-value artifacts rather than genuine improvements. The hop-aware structural attention causes the largest PEMS03 degradation at 0.47 MAE, and removing the global token costs 0.33 MAE on PEMS08, identifying structural modulation and global-state completion as the main sources of in-domain accuracy. The PEMS03 breakdown appears in Figure 4a of the main text.

Table 13: Ablation study of each component on PEMS03 and PEMS08 (12-step prediction).

Variant	PEMS03				PEMS08			
	MAE	RMSE	MAPE	MAPE_{10}	MAE	RMSE	MAPE	MAPE_{10}
VaryST (full)	**13.95**	**22.73**	13.59	**11.00**	12.89	21.99	8.50	**7.29**
w/o Laplacian PE	14.13	23.02	13.47	11.04	12.91	22.00	8.57	7.31
w/o Hop-RoPE	14.42	23.75	13.93	11.40	12.93	22.07	8.56	7.32
w/o structural encoding	14.10	23.05	13.55	11.11	12.92	22.06	8.54	7.30
w/o recent-12 token	14.10	23.00	**13.35**	**10.99**	**12.86**	**21.97**	**8.47**	**7.29**
w/o long context	15.95	25.42	15.73	13.04	15.02	25.51	9.81	8.45
w/o global token	14.06	22.97	13.47	11.04	13.22	22.23	8.67	7.47
Cosine retrieval	14.33	23.31	13.84	11.32	12.88	22.02	8.49	7.30
w/o tod/dow embedding	14.20	23.04	13.74	11.22	12.98	22.07	8.62	7.35

C.5 Full Hyperparameter Study Results

Table 14 presents the complete hyperparameter study. We evaluate 12-step prediction on PEMS03 and PEMS08 by sweeping each hyperparameter against the default configuration in Table 5: context length T across 36, 288, 864, 2016, and 4032 steps; patch length across 12, 36, and 96 steps; and the number of retrieved support nodes K across 16, 32, 64, and 128. Figure 4c illustrates the corresponding curves for PEMS03.

Context length exerts the strongest influence, characterized by a sharp initial error reduction that rapidly plateaus. Increasing T from 36 to 288 steps substantially lowers MAE by 1.04 on PEMS03 and 1.18 on PEMS08. Beyond 288 steps, the error declines gradually toward the one-week default ($T = 2016$), reaching MAEs of 13.95 and 12.89, respectively. Extending the context to two weeks ($T = 4032$) yields no further gain on PEMS08 and degrades MAE by 0.45 on PEMS03, confirming the default as the optimal operating point. This behavior directly reflects our representation design: compressing a 2016-step history into a single observation-derived token per node enriches the evidence underlying each token without expanding the $(K+3)$-token decoder sequence, up to the point of diminishing informational returns.

Spatial retrieval exhibits a similar economy. Even with only 16 support nodes, the MAE remains within 0.42 of the optimum on both benchmarks, whereas scaling K to 128 brings no consistent benefit. The RBF retrieval effectively concentrates relevant spatial signals into a small subset of nodes, while formatting them as prompt tokens preserves linear decoding complexity in K rather than quadratic scaling in total node count. Finally, patch length is the least sensitive hyperparameter; varying it across 12, 36, and 96 steps shifts MAE by at most 0.35 on both datasets, demonstrating that the representation is largely agnostic to the exact historical segmentation.

Table 14: Hyperparameter study on PEMS03 and PEMS08. The best and second-best results are in **bold** and underline, respectively.

Setting	PEMS03			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Input context length						
36	15.30	24.37	14.88	14.78	24.91	9.67
288	14.26	23.08	13.87	13.60	22.79	9.02
864	14.31	23.37	13.65	13.40	22.53	8.90
2016 (default)	**13.95**	**22.73**	**13.59**	**12.89**	21.99	**8.50**
4032	14.40	23.74	13.87	12.89	**21.97**	8.51
Patch length						
12	14.03	22.90	**13.47**	13.23	22.25	8.80
36 (default)	**13.95**	**22.73**	13.59	**12.89**	**21.99**	**8.50**
96	14.26	23.25	13.95	13.13	22.24	8.75
Number of auxiliary nodes						
16	14.37	23.33	13.94	12.89	22.06	8.49
32	14.11	23.02	13.64	12.93	22.02	8.54
64 (default)	**13.95**	**22.73**	**13.59**	12.89	21.99	8.50
128	14.17	23.14	13.72	**12.88**	**21.92**	**8.48**