TOWARDS EXPANDING-NODE SPATIAL-TEMPORAL FORECASTING: A STRUCTURED NODE INTERACTION PROMPTING PERSPECTIVE

Anonymous authors
Paper under double-blind review

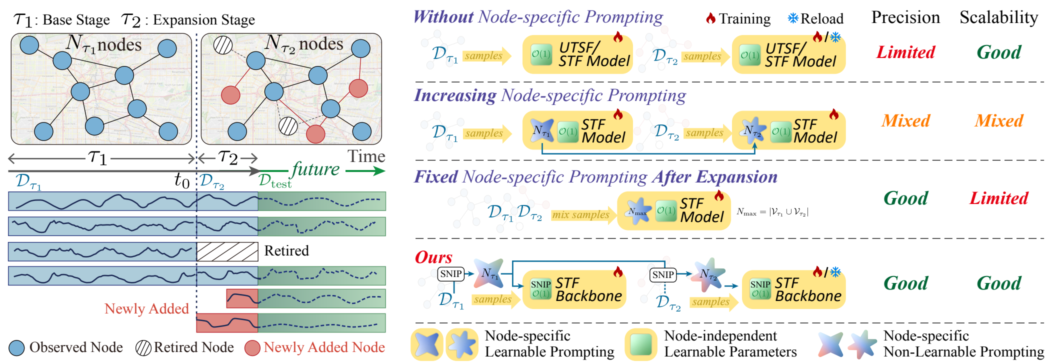
ABSTRACT

The rapid expansion of sensor systems, such as traffic networks, climate monitoring, and energy scheduling, poses new challenges for spatial-temporal series forecasting. While existing models have achieved strong performance under the fixed-node assumption, they rely on node-dependent parameters and fail to adapt when the network evolves, i.e., when old nodes are removed and new nodes with limited history are added. This expanding-node forecasting scenario introduces two critical challenges: (1) learning heterogeneous node representations without coupling learnable parameters to node count, and (2) enabling effective adaptation to new nodes with scarce observations. To tackle these challenges, we propose SNIP (Structured Node Interaction Prompting), a model-agnostic framework that constructs static spatial-temporal priors from historical observations and topology, and dynamically refines them during model training. Specifically, SNIP generates structured priors from three perspectives: periodic patterns across nodes, spatial-temporal interactions under time delays and graph structural information. These priors are projected into model as node promptings and then dynamically refined. For new nodes, SNIP initializes priors by similarity-weighted mixtures of old nodes and updates them with limited history, enabling efficient few-shot adaptation. Extensive experiments on multiple datasets demonstrate that SNIP outperforms state-of-the-art baselines in expanding-node scenarios. Beyond accuracy, SNIP provides plug-and-play generality and computational efficiency, bridging the gap between fixed-node precision and expanding-node adaptability in spatial-temporal forecasting.

1 INTRODUCTION

Spatial-temporal forecasting is crucial in cyber-physical systems such as traffic networks, climate monitoring, and energy scheduling. Despite recent advances, most models still rely on the ***fixed-node assumption***: training and inference are performed on a static node set, with parameters explicitly tied to node count. However, real systems are rarely static. Nodes may be *added* (e.g., new road sensors, weather stations) or *removed* (e.g., failures, replacements). This gives rise to the task of ***expanding-node forecasting***, where node sets evolve across periods, new nodes have scarce history, and some old nodes disappear, rendering traditional models ineffective.

To address this challenge, three lines of solutions have emerged (Figure 1): (1) **Node-independent parameterization.** Univariate time-series forecasting models forecast each node separately, which is scalable but neglects cross-variable dependencies. Others remove node embeddings and rely solely on sequence interactions (Liu et al., 2024; Ma et al., 2025a), while attention-based prompting (Hu et al., 2024) alleviates this partially but remains constrained by short horizons. (2) **Node-scaled Prompting.** Continual learning methods expand embeddings as new nodes appear (Chen & Liang, 2025), but usually assume abundant expansion data, which is unrealistic under scarcity. They also overlook node removal, causing wasted parameters and reduced flexibility. (3) **Fixed expanded parameterization.** A recent work, STEV (Ma et al., 2025b), introduces the Expanding-variate Time Series (EVTSF) forecasting task and mitigates imbalance with a flat scheme and shared subgraph. Nonetheless, it still relies on predefined embeddings for all expanded nodes. As a result, further network changes require costly retraining, limiting scalability.



(a) Expanding-Node Spatial-Temporal Forecasting (b) Comparison among existing solutions and ours SNIP

Figure 1: Examples of expanding-node spatial-temporal forecasting and different solutions. (a) Sensor nodes may added, retired or replaced in the expansion stage. (b) Comparison of three existing solution paradigms with our proposed SNIP framework.

In summary, while node-specific learnable parameters enhance forecasting accuracy (Shao et al., 2022; Liu et al., 2023; Dong et al., 2024), they either lack flexibility for new nodes when fixed or suffer from poor fitting under data scarcity when expanded, leading to a trade-off between accuracy and scalability. As a result, this raises a fundamental question:

Can effective node identification features be computed directly from historical observations, without relying on learnable node-dependent model parameters?

However, two critical challenges emerge: ***(Challenge 1) How to ensure that constructed features sufficiently capture inter-node heterogeneity and correlation***, preserving predictive accuracy comparable to learnable embeddings? ***(Challenge 2) How to refine these features dynamically to remain accurate under dynamic enviroments***, especially when new nodes arrive with only scarce observations?

To address these challenges, we propose SNIP (Structured Node Interaction Prompting), a model-agnostic prompting framework guided by structured priors and refined dynamically. Specifically, SNIP addresses the first challenge by computing priors from historical sequences through dimensionality reduction, which inherently preserves heterogeneity and correlation. Using PCA-based periodic features and Spectral embeddings of time-delayed interactions and graph topology, it effectively encodes node-specific heterogeneity without learnable embeddings. To tackle the second challenge, SNIP incorporates a dynamic refinement module that continuously adapts static priors through diffusion-based graph convolutions, thereby maintaining accuracy under dynamic evolving. Moreover, for new nodes with scarce observations, SNIP introduces a similarity-weighted initialization scheme that transfers priors from old nodes, providing effective embeddings that enable rapid few-shot adaptation. Through these two strategies, SNIP achieves parameter-node decoupling while maintaining both predictive accuracy and adaptability in expanding-node forecasting. Our contributions can be summarized as follows:

- We identify the problem of expanding-node spatial-temporal forecasting, where sensor networks evolve across periods, and highlight its core challenges of parameter-node coupling, data scarcity for new nodes, and preserving node heterogeneity. We further approach this problem from the perspective of structured node interactions.
- We propose SNIP (Structured Node Interaction Prompting), a framework that combines static prior construction (periodic, topological and time-delayed node interaction features) with dynamic refinement to build effective and flexible node promptings. In addition, we design a similarity-weighted initialization scheme to endow new nodes with initial embeddings, enabling efficient adaptation under few-shot conditions.
- A concrete instantiation of SNIP, termed SNIPformer, is further proposed. Extensive experiments on four datasets demonstrate that SNIP outperforms state-of-the-art baselines. Moreover, it serves as a plug-and-play module that enables classical spatial-temporal models to adapt flexibly and effectively to expanding-node forecasting.

2 Related Work

Spatial-temporal forecasting (STF) is central to applications such as traffic, energy, and climate. Early works combined recurrent or convolutional networks with graph modules to model temporal and spatial dependencies. With the advent of Spatio-Temporal Graph Neural Networks (STGNNs) and Transformers (Li et al., 2018; Wu et al., 2019; Guo et al., 2022), research has focused on capturing complex inter-node correlations via multi-view graphs or attention (Diao et al., 2024; Jiang et al., 2023). More recent advances explore adaptive embeddings (Shao et al., 2022; Zheng et al., 2025a) and hybrid neural modules (Sun et al., 2024; Lee & Ko, 2024) to balance efficiency and accuracy.

Node Prompting in STF. A consistent trend in these developments is the introduction of node-specific embeddings as additional identity information. By assigning learnable parameters to each node, models can capture inter-node heterogeneity beyond raw time series, which has shown strong forecasting performance (Liu et al., 2023; Dong et al., 2024). Such embeddings function as prompts that guide spatio-temporal modules, and have become an implicit consensus for achieving state-of-the-art accuracy. However, this design inherently ties model parameters to node count, limiting scalability in evolving networks. Recent work has further explored attention-based prompting mechanisms, such as STGP (Hu et al., 2024) and EAC (Chen & Liang, 2025), but these methods still rely on directly fitting prompts from data, which is challenging and assumes the availability of sufficient training samples.

STF under dynamic node expansion. In real-world systems, nodes may be added or removed over time, violating the fixed-node assumption in classical STF. To address this, recent works explored several directions. One approach decomposes data into univariate series or removes node-specific embeddings, which improves scalability but ignores spatial dependencies. Others directly learn from raw inputs or attention-based prompts (Liu et al., 2024; Hu et al., 2024), but accuracy drops due to insufficient heterogeneity modeling. Continual learning methods (Wang et al., 2023; Chen & Liang, 2025) expand embedding sets through prompt-tuning, yet typically assume abundant new data. OOD-generalization based methods (Wang et al., 2024; Ma et al., 2025a) emphasize robustness but lose accuracy when fine-tuning is feasible. A recent EVTSF paradigm, STEV (Ma et al., 2025b), mitigates imbalance via flattening and contrastive learning, but still relies on node-dependent parameters and costly retraining, limiting flexibility.

In contrast, our SNIP builds non-learnable priors and refines them dynamically, decoupling parameters from nodes while retaining node-specific effectiveness, and can be seamlessly integrated into existing STF models.

3 Preliminary

We consider a spatial-temporal network at time period τ, denoted as $\mathcal{G}_\tau = (\mathcal{V}_\tau, \mathcal{E}_\tau)$, where $\mathcal{V}_\tau = \{v_1, v_2, ..., v_{N_\tau}\}$ is the node set (e.g., road sensors, climate monitors), and $\mathcal{E}_\tau$ denotes the edges (e.g., road links, physical connections). The adjacency matrix is $\boldsymbol{A}_\tau \in \mathbb{R}^{N_\tau \times N_\tau}$, $N_\tau = |\mathcal{V}_\tau|$, representing the spatial relationships among nodes. Each node records C features within a temporal window of length L, forming a spatial-temporal series $\mathbf{X}_\tau \in \mathbb{R}^{L \times N_\tau \times C}$.

Definition (Expanding-node Spatial-Temporal Series). We define two consecutive periods. Period-1 **(base stage)** is $\tau_1 = [t_0 - L_1 + 1, t_0]$, with data $\mathcal{D}_{\tau_1} = (\mathcal{G}_{\tau_1}, \mathbf{X}_{\tau_1})$, where $|\mathcal{V}_{\tau_1}| = N_{\tau_1}$. L_1 denotes the length of base stage, and t_0 is the final time slice of this stage. Period-2 **(expansion stage)** is $\tau_2 = [t_0 + 1, t_0 + L_2]$, with data $\mathcal{D}_{\tau_2} = (\mathcal{G}_{\tau_2}, \mathbf{X}_{\tau_2})$, where $|\mathcal{V}_{\tau_2}| = N_{\tau_2}$. L_2 is the length of expansion stage. During the transition from the base stage to the expansion stage, nodes may be *added* or *removed*, which can be formalized as $\mathcal{V}_{\tau_2} = (\mathcal{V}_{\tau_1} \setminus \mathcal{V}_{\text{del}}) \cup \mathcal{V}_{\text{new}}, \mathcal{V}_{\text{del}} \subseteq \mathcal{V}_{\tau_1},\ \mathcal{V}_{\text{new}} \cap \mathcal{V}_{\tau_1} = \emptyset$. Moreover, to enable timely forecasting on newly deployed nodes, the available data in the expansion stage is typically very limited, i.e., $L_2 \ll L_1$, which leads the problem of data scarcity, particularly for newly added nodes.

Problem (Expanding-node Spatial-Temporal Forecasting). The goal of expanding-node spatial-temporal forecasting is to train a model f using data from both periods, such that f : $(\mathbf{X}_{t-T+1:t}, \mathcal{G}_\tau; \Theta) \mapsto \hat{\mathbf{Y}}_{t+1:t+T'}$, where $\mathbf{X}_{t-T+1:t} \in \mathbb{R}^{T \times N_\tau \times C}$ is the input sequence of length T, and $\hat{\mathbf{Y}}_{t+1:t+T'} \in \mathbb{R}^{T' \times N_\tau \times C}$ is the predicted sequence of length T'. A key requirement is that

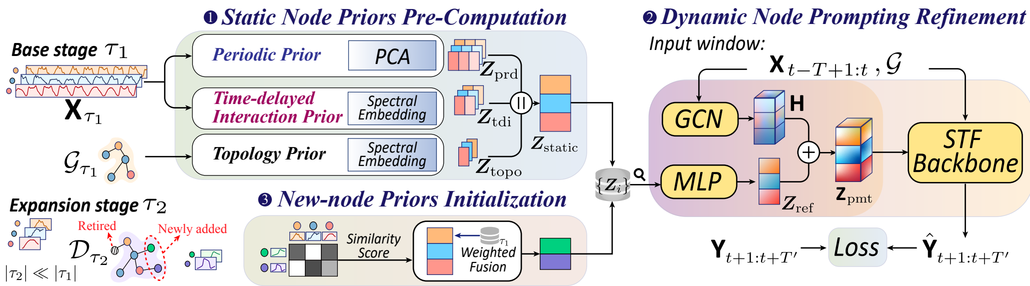


Figure 2: Overall framework of our proposed SNIP.

the parameter set Θ be decoupled from network size, i.e., $|\Theta| = \mathcal{O}(1)$, since parameter scaling with N limits adaptability to evolving networks. This enables the model to generalize across varying node sets. In practice, we evaluate forecasting on the expanded set $\mathcal{V}_{\tau_2}$, while the formulation naturally extends to any node set with $N_\tau \geq N_{\tau_1}$, ensuring applicability to future expansions.

4 Methodology

Figure 2 illustrates the overall structure of the proposed SNIP framework. In this section, we present the construction of structured static priors and their refinement during training and expansion. Then we introduce SNIPformer, an instantiation built on an efficient spatial-temporal encoder.

4.1 Structured Static Priors (Challenge 1)

In recent years, node-specific learnable embeddings have been widely used in spatio-temporal forecasting to provide discriminative identity information (Shao et al., 2022; Liu et al., 2023; Dong et al., 2024; Chen & Liang, 2025), achieving strong performance but conflicting with evolving node sets. To address this, SNIP avoids node-dependent parameters and instead pre-computes node-specific priors from historical data as prompting signals. We derive low-dimensional features that maximize inter-node variance to preserve heterogeneity. For clarity, we describe the single-feature case ($C = 1$), which naturally extends to multi-channel inputs by concatenation.

4.1.1 Periodic Priors

Motivation. Intuitively, a node's long-term sequence itself serves as its unique identifier, but directly using it is impractical due to dimensionality and noise. We instead apply dimensionality reduction to extract informative components. Given the strong periodicity of spatial-temporal data (e.g., daily or weekly cycles), we partition histories into repeated cycles, compress each into low-rank "snapshots," and average them to form a compact representation of node identity.

Periodic Priors. Given a historical period τ with length L, let $\boldsymbol{X} = \mathbf{X}_{:,:,1} \in \mathbb{R}^{L \times N}$ denote the historical sequence for N nodes. We specify a set of cycle lengths $\{p_1, p_2, \ldots, p_n\}$ and partition each node's series into non-overlapping segments accordingly. For instance, when p_j corresponds to one day, the sequence is divided into consecutive daily fragments. Each segment is normalized independently, and then projected into a low-dimensional representation using Principal Component Analysis (PCA) (Abdi & Williams, 2010). For each cycle length p_j, the node representations from all complete segments are averaged to yield a compact descriptor $\bar{\boldsymbol{Z}}^{(j)} \in \mathbb{R}^{N \times k_{\text{pca}}}$, k_{pca} represents the value of a low dimensionality. Finally, we concatenate results across all n cycle lengths to obtain the periodic priors:

$$\boldsymbol{Z}_{\text{prd}} = \text{Concat}\left(\bar{\boldsymbol{Z}}^{(1)}, \bar{\boldsymbol{Z}}^{(2)}, \ldots, \bar{\boldsymbol{Z}}^{(n)}\right) \in \mathbb{R}^{N \times (n \cdot k_{\text{pca}})}. \tag{1}$$

By the Eckart-Young-Mirsky theorem (Eckart & Young, 1936), PCA guarantees the optimal rank-k approximation under the Frobenius norm, thereby preserving the maximum variance. In our context, this ensures that the periodic priors retain the most discriminative directions of node dynamics, effectively encoding node ***heterogeneity*** from a temporal perspective. Implementation details are provided in Appendix A.1.

4.1.2 Topology Priors and Time-delayed Interaction Priors

Motivation. While periodic features separate node-specific temporal patterns, they overlook inter-node correlations, another key factor in spatial-temporal forecasting (Wang et al., 2022). Topology priors, derived from graph adjacency, capture latent positional and structural relations. Meanwhile, many spatial-temporal phenomena propagate with delays (e.g., traffic congestion spreading) (Long et al., 2024; Zheng et al., 2025a), and such delayed or short-term correlations are inherently dynamic. To capture these correlations, we construct two complementary priors: (1) topology features from static adjacency, and (2) time-delayed interaction features from frequency-domain correlations under short windows.

Topology Priors. We adopt spectral embedding (Belkin & Niyogi, 2003) to obtain low-dimensional node representations. In the case of topology, given the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, one can construct the normalized Laplacian: $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$, where $\boldsymbol{D}$ is the diagonal degree matrix with $D_{i,i} = \sum_j A_{i,j}$. Then, the topology embedding is formed by the eigenvectors corresponding to the smallest k_{topo} eigenvalues of $\boldsymbol{L}$. This can be formulated as:

$$\boldsymbol{Z}_{\text{topo}} = \Phi(\boldsymbol{A}, k_{\text{topo}}) = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{k_{\text{topo}}}] \in \mathbb{R}^{N\times k_{\text{topo}}}, \tag{2}$$

where $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{k_{\text{topo}}}$ are the leading eigenvectors of the normalized Laplacian. This embedding captures the static positional structure of nodes in the network, where nearby or strongly connected nodes are embedded closer together. Physically, they reflects both global communities and local connectivity patterns.

Time-delayed Interaction Priors. Recent studies have shown that correlations between node sequences often emerge more strongly when temporal delays are considered, rather than assuming synchronous dynamics (Long et al., 2024; Zheng et al., 2025a). To capture such effects, we estimate cross power spectral density (CSD) between node pairs using Welch's method with short sliding windows (Welch, 1967). This formulation enables us to measure correlations across all possible lags without pre-specifying a maximum delay in previous STF models. Consequently, we can obtain the cross-correlation matrices under different time delays: $\boldsymbol{R}(\delta)$. From this, we extract (i) the dominant delay $\Delta_{i,j} = \arg\max_\delta |R_{i,j}(\delta)|$, that maximizes correlation between nodes i and j, and (ii) the corresponding correlation strength $P_{i,j} = \max_\delta |R_{i,j}(\delta)|$. These two matrices encode how information propagates with delays and how strongly nodes interact. Similarly, we apply spectral embedding to both matrices, and then concatenate results into $\boldsymbol{Z}_{\text{tdi}}$:

$$\boldsymbol{Z}_{\text{tdi}} = \text{Concat}(\Phi(\Delta, k_{\text{delay}}), \Phi(\boldsymbol{P}, k_{\text{corr}})) \in \mathbb{R}^{N\times(k_{\text{delay}}+k_{\text{corr}})}. \tag{3}$$

In summary, topological priors preserve static, position-driven relationships, while time-delayed embeddings capture dynamic propagation and short-term coupling. In particular, spectral embeddings emphasize the principal eigenvectors, which correspond to directions of maximum structural or interaction variance, this is analogous to PCA but under graph constraints. These priors reflect how nodes interact and differ within the network, boosting promptings from the ***correlation*** angle. Details of CSD method are provided in the Appendix A.2.

4.2 Dynamic Refinement and Adaptation (Challenge 2)

Motivation. The static priors in Section 4.1.2 capture invariant properties but cannot reflect temporal dynamics, such as evolving behaviors of existing nodes or the emergence of new nodes during expansion. To address this, we design a refinement-and-adaptation mechanism that treats above three priors as reference points subject to dynamic correction. To formalize this intuition, we propose the following hypothesis, which conceptualizes how an optimal node prompting should be decomposed into stable and dynamic components.

Hypothesis 1 (Decomposition of Optimal Node Prompting)**.** *At any time t, there exists an optimal prompting configuration $\boldsymbol{z}_{i\star}^{(t)}$ for each node i, which maximizes predictive accuracy. This configuration can be decomposed as: $\boldsymbol{z}_{i\star}^{(t)} = \boldsymbol{q}_i + \boldsymbol{r}_i^{(t)}, \boldsymbol{r}_i^{(t)} = g(\boldsymbol{x}_j^{(t)}, j \in \mathbb{N}_{(i)})$, where $\boldsymbol{q}_i$ represents spatially intrinsic characteristics of node i (time-invariant reference), $\boldsymbol{r}_i^{(t)}$ reflects spatial-temporal interaction effects that vary over time, $\mathbb{N}_{(i)}$ is the set of nodes that have a correlation relationship with node i, and g is a transformation function.*

If the proposed static prior in Section 4.1 approximates $\boldsymbol{q}_i$, the backbone STF model only needs to fit $\boldsymbol{r}_i^{(t)}$, which typically exhibits lower variance. Consequently, the hypothesis space is constrained, yielding reduced sample complexity and improved generalization under limited data (Vapnik, 1999). We will provide empirical ablations in Section 5.3 to support this assumption.

Dynamic refinement via MLP and diffusion graph convolution. We first project the concatenated static priors into the model dimension d using a two-layer MLP: $\boldsymbol{Z}_{\text{ref}} = \text{MLP}([\boldsymbol{Z}_{\text{prd}}, \boldsymbol{Z}_{\text{topo}}, \boldsymbol{Z}_{\text{tdi}}]) \in \mathbb{R}^{N\times d}$. We refine priors by aggregating temporal variations through diffusion graph convolution (Li et al., 2018; Wu et al., 2019). Specifically, for each input time slice, we apply the diffusion convolution operation: $\mathbf{H}_{t,:,:} = \texttt{DiffGCN}(\mathbf{X}^{\text{emb}}_{t,:,:}, \boldsymbol{A})$, where $\mathbf{X}^{\text{emb}} \in \mathbb{R}^{T\times N\times d}$ is the series embedding and $\boldsymbol{A}$ is the adjacency matrix. To simulate potential changes in graph topology during the expansion stage, we further apply edge dropout to $\boldsymbol{A}$ during training, enhancing robustness to evolving structures. The final adaptive embedding is obtained by combining static refinement and dynamic aggregation:

$$(\mathbf{Z}_{\text{pmt}})_{t,:,:} = \boldsymbol{Z}_{\text{ref}} + \mathbf{H}_{t,:,:}, \tag{4}$$

This embedding $\mathbf{Z}_{\text{pmt}}$ not only incorporates static priors but also adapts to temporal variations, serving as the prompting within the forecasting model.

Prompting initialization in expansion stage. In the expansion stage, new nodes often lack sufficient history to compute reliable priors. For these nodes, we adopt a similarity-based initialization. Specifically, their priors can either be recomputed directly from the limited data available in the expansion stage, or constructed by weighted mixing of the priors from a few most similar remain nodes in the base stage. Similarity is measured using the cross-correlation matrix $\boldsymbol{P}$ introduced in Section 4.1.2, recomputed under the current stage. Formally, For a new node i, its similarity weight with remain node j is calculated as $s_{i,j} = P_{i,j}/\sum_{j\in\mathcal{V}_{\text{remain}}} P_{i,j},\ v_i \in \mathcal{V}_{\text{new}}, ; v_j \in \mathcal{V}_{\text{remain}}$, and its prior of type $\P \in \{\text{prd}, \text{topo}, \text{tdi}\}$ is obtained as $(\boldsymbol{z}_\P)_i = \sum_{j\in\mathcal{N}_{k(i)}} s_{i,j}\, (\boldsymbol{z}_\P)_j$, where $\mathcal{N}_{k(i)}$ denotes the top-k most similar remain nodes. When expansion-stage data are too scarce,$(\boldsymbol{z}_{\text{prd}})_i$ is directly mixed from old nodes' periodic priors. For remain nodes, the priors computed in the base stage are reused without modification.

Moreover, most existing works rarely consider nodes that are removed during the expansion stage. In our framework, discarded nodes do not require prior construction in the new period, and since model parameters are decoupled from node identity, no redundant parameters are left unused. This design avoids parameter waste and provides additional flexibility and convenience for evolving networks.

4.3 Integration with Spatial-temporal Forecasting Models

Based on the static priors and dynamic refinement introduced above, SNIP can be seamlessly integrated into existing STF architectures by injecting the prompting into the input features before spatial-temporal feature extraction. To establish a baseline for expanding-node forecasting, we integrate the SNIP framework with a recent efficient spatio-temporal encoder (Zheng et al., 2025b), which provides a general mechanism for learning compact and expressive representations with complexity linear in the number of nodes. The resulting model, SNIPformer, incorporates our prior-guided prompting into the encoder's input embedding and spatial-temporal extraction process, followed by a lightweight regression head for prediction. The complete model structure is presented in the Appendix A.3.

5 Experiments

In this section, we evaluate and analysis the effectiveness, generality, and flexibility of our proposed SNIP framework under node expansion scenarios using four real-world datasets.

5.1 Experiment Setting

Datasets and Evaluation Setting. We use the following spatial-temporal datasets across traffic and energy domain for evaluation: **EPeMS** (Ma et al., 2025b), **PEMS04** (Song et al., 2020), **SeaLoop** (Cui et al., 2019), and **NREL-AL** (Xu et al., 2025). For EPeMS, we follow the node expansion setup introduced in STEV (Ma et al., 2025b). For the other datasets, we simulate node expansion by randomly partitioning the node set into remain, deleted, and newadd groups. The detailed implementation procedure is provided in the Appendix B.1. Table 1 summarizes the stage and node partitions. We use a 12-step history to predict the next 12 steps, correspond to 1 hour ahead prediction.

Table 1: Dataset statistics and characteristics

Dataset	Stage Split τ_1 / τ_2 / test	Node Expansion ($\tau_1 \rightarrow \tau_2$)
EPeMS	63d / 3d + 2d / 22d	296 → 447
PEMS04	35d / 6d + 1d / 17d	241 → 290
SeaLoop	18d / 6d + 1d / 3d	255 → 303
NERL-AL	122d / 6d + 1d / 53.5d	103 → 130

We compute prior features in the ***base stage*** using full historical data and train models with sliding-window samples. In the ***expansion stage***, priors are recomputed from short-term history and priors transferred from the base stage, followed by fine-tuning. Final evaluation is conducted in the ***test stage***. We report Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) in the main tables, while Mean Absolute Percentage Error (MAPE) and Mean Relative Error (MRE) are provided in the Appendix with consistent conclusions.

Table 2: Comparison of the expanding-node forecasting results of different methods and SNIPformer.

Model	Metric	EPeMS All	EPeMS Remain	EPeMS New	PEMS04 All	PEMS04 Remain	PEMS04 New	SeaLoop All	SeaLoop Remain	SeaLoop New	NREL-AL All	NREL-AL Remain	NREL-AL New
DLinear	MAE	32.70	32.26	33.56	28.97	28.91	29.17	4.59	4.62	4.49	2.54	2.59	2.41
	RMSE	48.32	48.14	48.66	44.55	44.29	45.42	7.99	8.02	7.92	3.91	4.00	3.63
iTransformer	MAE	26.83	26.65	27.16	24.76	24.74	24.83	4.29	4.31	4.21	1.94	1.97	1.83
	RMSE	41.40	41.22	41.73	39.62	39.34	40.52	7.54	7.56	7.46	3.36	3.44	3.12
DUETformer	MAE	25.25	25.17	25.39	23.21	23.21	23.24	4.02	4.04	3.93	1.82	1.85	1.72
	RMSE	38.05	38.18	37.77	36.54	36.32	37.26	7.02	7.04	6.93	3.10	3.17	2.88
GWNET†	MAE	23.73	23.11	24.93	22.99	23.05	22.79	3.94	3.97	3.84	1.79	1.83	1.69
	RMSE	35.81	35.27	36.84	36.70	36.57	37.16	6.82	6.86	6.68	3.16	3.24	2.92
STID†	MAE	24.40	24.31	24.56	22.49	22.56	22.25	4.10	4.12	4.03	2.00	2.03	1.89
	RMSE	37.38	37.44	37.23	35.92	35.77	36.42	7.26	7.28	7.20	3.25	3.33	3.01
STAEformer†	MAE	24.86	24.66	25.27	22.95	23.03	22.67	4.15	4.17	4.09	1.91	1.95	1.81
	RMSE	38.34	38.26	38.50	36.75	36.63	37.14	7.38	7.39	7.37	3.29	3.37	3.05
STOP	MAE	24.45	24.47	24.41	22.54	22.56	22.46	4.12	4.13	4.08	2.01	2.05	1.89
	RMSE	37.24	37.41	36.89	35.74	35.52	36.45	7.32	7.32	7.35	3.25	3.32	3.02
STKEC	MAE	29.99	29.78	30.40	25.64	25.84	24.87	5.00	5.01	4.98	2.33	2.34	2.28
	RMSE	42.91	43.05	42.64	39.55	39.74	38.73	8.14	8.13	8.16	3.62	3.66	3.51
EAC	MAE	28.74	28.23	29.75	24.05	24.27	23.21	4.72	4.73	4.72	2.16	2.17	2.14
	RMSE	40.33	39.80	41.35	36.51	36.79	35.41	7.82	7.81	7.86	3.37	3.39	3.32
STEV	MAE	22.90	22.35	23.97	20.55	20.42	21.01	3.92	3.95	3.84	**1.57**	**1.58**	**1.53**
	RMSE	34.51	33.95	35.60	32.46	32.13	33.53	6.62	6.66	6.51	2.88	2.93	2.73
SNIPformer (ours)	MAE	**22.05**	**21.39**	**23.35**	**19.20**	**19.22**	**19.10**	**3.46**	**3.47**	**3.42**	1.62	1.65	1.55
	RMSE	**33.91**	**33.16**	**35.33**	**31.02**	**30.87**	**31.54**	**6.10**	**6.14**	**5.97**	**2.87**	**2.92**	**2.71**

Baselines and Hyperparameter Settings. We compare SNIPformer (introduced in Section 4.3) with four categories of existing solutions for expanding-node STF: 1) Models without node-specific prompting: **DLinear** (Zeng et al., 2023), **iTransformer** (Liu et al., 2024), **DUETformer** (Qiu et al., 2025). 2) STF models without node-specific modules: **GWNET**† (Wu et al., 2019), **STID**† (Shao et al., 2022), **STAEformer**† (Liu et al., 2023), **STOP** (Ma et al., 2025a), where † indicates removal of learnable node embeddings. 3) Continual learning methods: **STKEC** (Wang et al., 2023), **EAC** (Chen & Liang, 2025). 4) Fixed-node models after expansion: **STEV** (Ma et al., 2025b). For SNIPformer, we set the PCA feature dimension to 24 (each for daily and weekly periods) and the spectral embedding dimension to 8. The model dimension is 64 (32 for NREL-AL). Other implementation details are provided in the Appendix. Average results are reported after repeating the experiments no less than five times.

5.2 Effectiveness and Generality

Expanding-node forecasting results. Table 2 summarizes the results across all nodes, *Remain* nodes, and *New* nodes, where the best results are highlighted in **bold red** and the second-best results in underlined blue. SNIP achieves the best performance on the three traffic datasets, with relative averaged improvements up to 7.61% / 5.61% in MAE and RMSE over the strongest baselines.

On NREL-AL, SNIP ranks second on MAE, slightly below STEV. We attribute this gap to domain-specific characteristics, such as stronger trend strength (Qiu et al., 2024)(in Table 5) and more severe distribution shifts, which are more effectively captured by the contrastive learning strategy in STEV. Nevertheless, compared to node-agnostic models and continual learning approaches, SNIP consistently delivers superior accuracy, confirming the effectiveness of structured priors in encoding node heterogeneity under expansion scenarios.

Generality across architectures. To validate SNIP's model-agnostic design, we integrate it into three representative backbones: iTransformer, GWNET, and STID. Table 3 reports the results of (i) the **original backbone**, (ii) **backbone + AttP**, an attention-based prompting module proposed in STGP (Hu et al., 2024), and (iii) **backbone + SNIP**. Across all cases, AttP does not yield noticeable improvements, whereas SNIP consistently and significantly enhances forecasting performance under dynamic node changes. This confirms that prior-guided prompting provides a more effective way to capture node heterogeneity and adapt to evolving networks.

Table 3: Forecasting MAE of different backbones with and without prompting modules.

Model	EPeMS			NREL-AL		
	All	Remain	New	All	Remain	New
iTransformer	26.83	26.65	27.16	1.94	1.97	1.83
+ AttP	26.81	26.64	27.14	1.95	1.99	1.84
+ SNIP	**24.67**	**24.14**	**25.71**	**1.84**	**1.88**	**1.74**
GWNET†	23.73	23.11	24.93	1.79	1.83	1.69
+ AttP	23.75	23.13	24.96	1.79	1.82	1.69
+ SNIP	**23.41**	**22.79**	**24.62**	**1.77**	**1.81**	**1.66**
STID†	24.40	24.31	24.56	2.00	2.03	1.89
+ AttP	24.35	24.26	24.53	2.01	2.05	1.90
+ SNIP	**21.84**	**21.13**	**23.23**	**1.86**	**1.89**	**1.77**

More importantly, these results highlight SNIP's generality: as a model-agnostic framework, it can be seamlessly combined with diverse forecasting architectures, enabling them to remain effective in expanding-node scenarios while preserving strong accuracy. This suggests that prompting frameworks and spatio-temporal feature extractors can evolve in parallel as complementary directions.

5.3 Ablation Studies

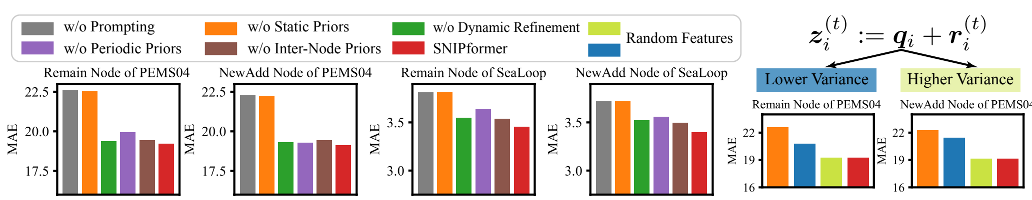


Figure 3: Ablation Results. Left: Comparison of contribution of different components. Right: Performance of using random features with high and low variance as static priors.

Component-wise analysis We first assess the contribution of different components in SNIP by progressively removing them: (i) w/o Prompting, (ii) w/o Static Priors, (iii) w/o Dynamic Refinement, (iv) w/o Periodic Priors, and (v) w/o Inter-node Priors (removing both topology and time-delayed interaction priors). Figure 3 reports results on PEMS04 and SeaLoop, evaluated on *remain* nodes and *newadd* nodes. The results yield several key insights. Removing prompting causes a substantial accuracy drop; relying solely on dynamic refinement to learn full embeddings also performs poorly, suggesting that directly fitting optimal embeddings without helpful priors is highly challenging. In contrast, using only static priors without refinement underscores the necessity of modeling temporal variations. Finally, eliminating periodic or inter-node priors consistently degrades performance, validating that the constructed priors effectively encode node heterogeneity and structural dependencies.

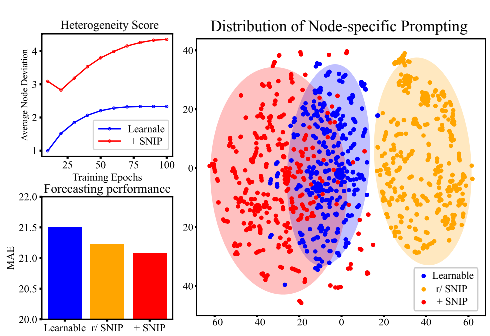


Figure 4: Contribution of SNIP to the STID model on PEMS04 dataset. Right: Distribution of node-specific prompting after dimensionality reduction via t-SNE.

Empirical analysis of decomposition and heterogeneity We further validate Hypothesis 1 by replacing static priors with alternative designs: (a) random priors with high variance, (b) random priors with low variance, and (c) no static priors. Figure 4 shows that under the decomposition framework of Hypothesis 1, even randomly initialized features can achieve competitive results. Moreover, larger initialization variance improves performance, underscoring the importance of heterogeneity.

To intuitively demonstrate the heterogeneity introduced by SNIP prompting, we visualize results on the PEMS04 dataset under a fixed-node forecasting setup with STID in Figure 4. When the learnable embeddings in STID are either replaced by SNIP or augmented with SNIP, both heterogeneity score (Chen & Liang, 2025) and predictive performance improve. As shown in the t-SNE visualization under a unified embedding space, the combination of learnable embeddings and SNIP yields a wider spread and more distinct clusters, indicating that SNIP effectively enhances heterogeneity.

5.4 Efficiency and Flexibility

Computational efficiency is an important consideration for expanding-node forecasting. The additional cost of SNIP mainly comes from three preprocessing operations: multi-cycle PCA for periodic features, cross-correlation estimation between node pairs, and spectral embedding of the resulting matrices. Crucially, all of these steps are performed once in the base stage, and the priors are reused throughout training and expansion. As shown in Table 4, the one-off preprocessing overhead is minor compared with training time.

When comparing training and inference efficiency, SNIPformer shows clear advantages over the strongest baseline, STEV. While STEV incurs heavy retraining whenever nodes are expanded, SNIPformer requires only lightweight fine-tuning with precomputed priors. This results in substantial reductions in both training time and memory consumption, while maintaining competitive accuracy. In addition, applying SNIP to classical backbones such as STAEformer introduces only minimal extra cost, yet enables these models to operate effectively in expansion scenarios where their original designs fail. Overall, SNIP achieves high efficiency, flexibility, and scalability, offering a model-agnostic prompting framework that can be seamlessly incorporated into existing or future STF architectures.

Table 4: Training and inference efficiency comparison on EPeMS (batch size = 32).

Metric	**STEV**	**SNIPformer**	⇕%	**STAEformer**	**STAEformer**† **+SNIP**	⇕%
Pre-computation Time Cost (min)	Augmentation 0.21	Static Priors 2.61	-	-	Static Priors 2.61	-
Training	(τ_1, τ_2)	$(\tau_1 \rightarrow \tau_2)$		(τ_1)	(τ_1)	
Time (s/epoch)	<u>325.42</u>	**28.26** → **1.42**	↓91.3%	**132.03**	<u>134.77</u>	↑2.0%
Footprint (MB)	<u>31430</u>	**1466** → **2358**	↓92.5%	**8130**	<u>8296</u>	↑2.0%
Inference Time(s)	<u>20.46</u>	**1.05**	↓94.9%	Invalid	1.37	↑ Feasibility
(τ_2) MAE	<u>22.90</u>	**22.05**	↓3.7%		23.75	

6 Conclusion

In this paper, we proposed SNIP, a model-agnostic prompting framework for expanding-node spatial-temporal forecasting. It constructs structured static priors from heterogeneity and correlation angles and performing learnable dynamic refinement. A similarity-weighted initialization further enables few-shot adaptation for new nodes. SNIP allows existing spatio-temporal forecasting models to be easily adapted to expanding-node scenarios. Experiments across multiple datasets and backbones show that SNIP achieves strong accuracy, generality, and efficiency. Ablations show that variance-preserving, correlation-aware priors and dynamic refinement are all indispensable. Future work will study the optimal composition of prompting, extend SNIP to cross-domain settings, and integrate it as a prompting layer in large spatial-temporal models.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317.

Wei Chen and Yuxuan Liang. Expand and compress: Exploring tuning principles for continual spatio-temporal graph forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FRzCIlkM7I&noteId=RDXGMROaMj.

Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

Zulong Diao, Xin Wang, Dafang Zhang, Gaogang Xie, Jianguo Chen, Changhua Pei, Xuying Meng, Kun Xie, and Guangxing Zhang. DMSTG: Dynamic Multiview Spatio-Temporal Networks for Traffic Forecasting. *IEEE Transactions on Mobile Computing*, 23(6):6865–6880, June 2024. ISSN 1558-0660. doi: 10.1109/TMC.2023.3328038.

Zheng Dong, Renhe Jiang, Haotian Gao, Hangchen Liu, Jinliang Deng, Qingsong Wen, and Xuan Song. Heterogeneity-informed meta-parameter learning for spatiotemporal time series forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 631–641, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671961. URL https://doi.org/10.1145/3637528.3671961.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

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, 2022. doi: 10.1109/TKDE.2021.3056502.

Junfeng Hu, Xu Liu, Zhencheng Fan, Yifang Yin, Shili Xiang, Savitha Ramasamy, and Roger Zimmermann. Prompt-based spatio-temporal graph transfer learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, pp. 890–899, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704369. doi: 10.1145/3627673.3679554. URL https://doi.org/10.1145/3627673.3679554.

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, Jun. 2023. doi: 10.1609/aaai.v37i4.25556.

Hyunwook Lee and Sungahn Ko. TESTAM: A Time-Enhanced Spatio-Temporal Attention Model with Mixture of Experts. In *International Conference on Learning Representations*, pp. 1–12, October 2024.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, pp. 1–10. OpenReview.net, 2018.

Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Quanjun Chen, and Xuan Song. Spatio-Temporal Adaptive Embedding Makes Vanilla Transformer SOTA for Traffic Forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pp. 4125–4129, New York, NY, USA, October 2023. Association for Computing Machinery. ISBN 979-8-4007-0124-5.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=JePfAI8fah`.

Qingqing Long, Zheng Fang, Chen Fang, Chong Chen, Pengfei Wang, and Yuanchun Zhou. Unveiling delay effects in traffic forecasting: A perspective from spatial-temporal delay differential equations. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 1035–1044, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701719. doi: 10.1145/3589334.3645688. URL `https://doi.org/10.1145/3589334.3645688`.

Jiaming Ma, Bingwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. Robust spatio-temporal centralized interaction for ood learning. In *Proceedings of the Forty-Second International Conference on Machine Learning (ICML)*, 2025a.

Minbo Ma, Kai Tang, Huan Li, Fei Teng, Dalin Zhang, and Tianrui Li. Beyond fixed variables: Expanding-variate time series forecasting via flat scheme and spatio-temporal focal learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, pp. 2054–2065, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400714542. doi: 10.1145/3711896.3736854. URL `https://doi.org/10.1145/3711896.3736854`.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.*, 17(9): 2363–2377, May 2024. ISSN 2150-8097. doi: 10.14778/3665844.3665863. URL `https://doi.org/10.14778/3665844.3665863`.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, pp. 1185–1196, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712456. doi: 10.1145/3690624.3709325. URL `https://doi.org/10.1145/3690624.3709325`.

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*, CIKM '22, pp. 4454–4458, New York, NY, USA, 2022. ISBN 9781450392365.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 914–921, 2020. doi: 10.1609/aaai.v34i01.5438.

Ke Sun, Pei Liu, Pengfei Li, and Zhifang Liao. ModWaveMLP: MLP-Based Mode Decomposition and Wavelet Denoising Model to Defeat Complex Structures in Traffic Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(8):9035–9043, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i8.28753.

V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10 (5):988–999, 1999. doi: 10.1109/72.788640.

Binwu Wang, Yudong Zhang, Jiahao Shi, Pengkun Wang, Xu Wang, Lei Bai, and Yang Wang. Knowledge expansion and consolidation for continual traffic prediction with expanding graphs. *Trans. Intell. Transport. Sys.*, 24(7):7190–7201, July 2023. ISSN 1524-9050. doi: 10.1109/TITS.2023.3263904. URL `https://doi.org/10.1109/TITS.2023.3263904`.

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*, KDD '24, pp. 2948–2959, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671680. URL `https://doi.org/10.1145/3637528.3671680`.

Senzhang Wang, Jiannong Cao, and Philip S. Yu. Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3681–3700, 2022. doi: 10.1109/TKDE.2020.3025580.

P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967. doi: 10.1109/TAU.1967.1161901.

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, IJCAI-19*, pp. 1907–1913, July 2019. doi: 10.24963/ijcai.2019/264.

Qianxiong Xu, Cheng Long, Ziyue Li, Sijie Ruan, Rui Zhao, and Zhishuai Li. Kits: inductive spatio-temporal kriging with increment training strategy. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i12.33412. URL `https://doi.org/10.1609/aaai.v39i12.33412`.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i9.26317. URL `https://doi.org/10.1609/aaai.v37i9.26317`.

Qi Zheng, Minhua Shao, and Yaying Zhang. Tlast: A time-lag aware spatial-temporal transformer for traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 26(9): 13144–13158, 2025a. doi: 10.1109/TITS.2025.3583391.

Qi Zheng, Zihao Yao, and Yaying Zhang. St-rep: Learning predictive representations efficiently for spatial-temporal forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (12):13419–13427, Apr. 2025b. doi: 10.1609/aaai.v39i12.33465. URL `https://ojs.aaai.org/index.php/AAAI/article/view/33465`.

A Appendix: Methodology Details

A.1 Periodic Priors Construction

Given a historical period τ with a length of L. Let $\boldsymbol{X} \in \mathbb{R}^{L \times N}$ denote the historical sequence of for N nodes. We specify a set of cycle lengths $\{p_1, p_2, \ldots, p_n\}$. For node i and a given cycle length p_j, we partition its sequence $\boldsymbol{X}_{:,i} \in \mathbb{R}^L$ into non-overlapping cycle segments. For example, when p_j corresponds to one day, the sequence is divided into consecutive daily fragments, each treated as an individual segment. Formally, the set of segments is defined as:

$$\mathbb{S}_i^{(j)} = \text{Partition}(\boldsymbol{X}_{:,i},\, p_j) = \Big\{ \boldsymbol{X}_{:,i}[(m-1)p_j + 1 : mp_j] \;\Big|\; m = 1, \ldots, M_j \Big\}, \tag{5}$$

where $M_j = \lfloor L/p_j \rfloor$ is the number of complete cycles. Each element of $\mathbb{S}_i^{(j)}$ is a vector in $\mathbb{R}^{p_j}$. Before dimensionality reduction, each segment of node i is normalized independently: $\tilde{\boldsymbol{x}} = (\boldsymbol{x} - \mu_i^{(j)})/\sigma_i^{(j)}, \quad \boldsymbol{x} \in \mathbb{S}_i^{(j)}$, where $\mu_i^{(j)}$ and $\sigma_i^{(j)}$ are the mean and standard deviation of node i's segments under cycle length p_j.

Each normalized segment $\tilde{\boldsymbol{X}}_m^{(j)}$ is treated as an $N \times p_j$ data matrix, which is the full "snapshot" of all nodes in cycle j. We then apply Principal Component Analysis (PCA) to reduce these segments to their low-rank components and obtain compact representations. Specifically, PCA yields a projection matrix $\boldsymbol{U}^{(j)} \in \mathbb{R}^{p_j \times k_{\text{pca}}}$ from the top k_{pca} eigenvectors of the covariance matrix of $\tilde{\boldsymbol{X}}_m^{(j)}$. Then the segment-level low-dimensional representation is computed, and the M_j representations are the averaged across segments:

$$\bar{\boldsymbol{Z}}^{(j)} = \frac{1}{M_j} \sum_{m=1}^{M_j} \tilde{\boldsymbol{X}}_m^{(j)} \boldsymbol{U}^{(j)} \in \mathbb{R}^{N \times k_{\text{pca}}}, j \in [1, ..., n]. \tag{6}$$

Finally, the representations from all n cycle lengths are concatenated, yielding the periodic prior feature matrix:

$$\boldsymbol{Z}_{\text{prd}} = \text{Concat}\left(\bar{\boldsymbol{Z}}^{(1)}, \bar{\boldsymbol{Z}}^{(2)}, \ldots, \bar{\boldsymbol{Z}}^{(n)}\right) \in \mathbb{R}^{N \times (n \cdot k_{\text{pca}})}. \tag{7}$$

Figure 5 illustrates this construction.

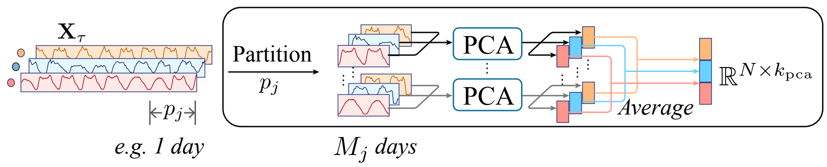


Figure 5: An illustration for the periodic prior construction under a cycle length p_j.

A.2 Time-delayed Interaction Priors Construction

As a increasing trend investigated by recent studies(Long et al., 2024; Zheng et al., 2025a), the correlation between two node sequences is often more pronounced when a temporal delay is considered rather than assuming synchronous dynamics. To capture this, we quantify their association through the cross power spectral density, which avoids the limitation of manually specifying a maximum delay as required in previous research. This formulation allows us to directly compute the delay step that maximizes their correlation, along with the corresponding strength. Intuitively, these two quantities characterize both the temporal span and the spatial extent of the interaction between nodes.

Formally, let $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^L$ denote the historical sequences of nodes i and j. Each sequence is normalized in the same manner as in periodic features. Their cross-spectral density (CSD) is estimated using Welch's method with a Hann window (Welch, 1967) of length T:

$$Q_{ij}(\nu) = \frac{1}{K} \sum_{k=1}^{K} X_i^{(k)}(\nu) X_j^{(k)}(\nu)^*, \tag{8}$$

where $K = \lfloor L/T \rfloor$ is the number of windows, $X_i^{(k)}(\nu)$ is the Fourier transform of the k-th windowed segment of node i, ν is the frequency variable, and * denotes complex conjugation. The cross-correlation function is obtained by inverse FFT:

$$R_{ij}(\delta) = \mathcal{F}^{-1}\big(Q_{ij}(\nu)\big), \tag{9}$$

which is then shifted to align both positive and negative delays. We extract the most significant delay and its corresponding correlation strength as

$$\Delta_{ij} = \arg\max_{\delta} |R_{ij}(\delta)|, \qquad \boldsymbol{P}_{ij} = \max_{\delta} |R_{ij}(\delta)|. \tag{10}$$

where Δ is the delay matrix recording absolute dominant lags, and $\boldsymbol{P}$ is the correlation matrix recording absolute correlation strengths. Following the same procedure, we apply spectral embedding to the delay and correlation matrices, and then concatenate them into $\boldsymbol{Z}_{\text{tdi}}$:

$$\boldsymbol{Z}_{\text{tdi}} = \text{Concat}(\Phi(\Delta, k_{\text{delay}}), \Phi(\boldsymbol{P}, k_{\text{corr}})). \tag{11}$$

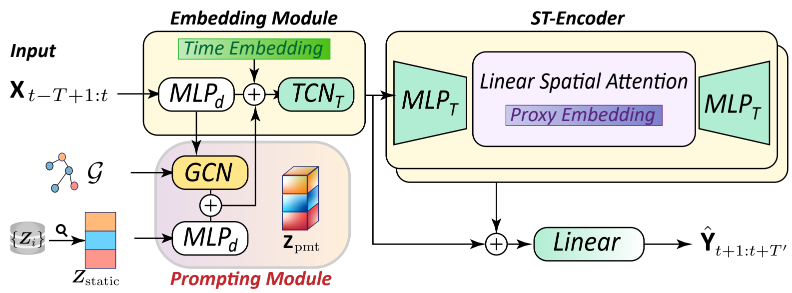


Figure 6: The architecture of SNIPformer.

A.3 Architecture of SNIPformer

Figure 6 represents the entire architecture of SNIPformer. We use the data embedding module and spatial-temporal extractor proposed by ST-ReP (Zheng et al., 2025b) as the main architecture. Differently, we remove the learnable spatial embeddings in the original model and use our dynamic refinement module and pre-computed static priors to build a new node embedding for input series. Moreover, we use a linear head to transform the flattened spatial-temporal hidden features into prediction.

B Appendix: Experiment Details

B.1 Datasets and Evaluation Setting.

We use the following spatial-temporal datasets across traffic and energy domain for evaluation:

- **EPeMS**(Ma et al., 2025b): an expansion-node dataset constructed in STEV (Ma et al., 2025b) from District 7 of California, which assumes no deleted nodes.
- **PEMS04** (Song et al., 2020): traffic flow data collected from the Caltrans Performance Measurement System in California.
- **SeaLoop** (Cui et al., 2019): Seattle traffic loop detector data, recording speed measurements.
- **NREL-AL** (Xu et al., 2025): renewable energy data, recording solar power generation from photovoltaic plants in Alabama in 2016.

The number of feature values for all dataset records is 1, i.e., C=1. Each dataset is divided into three stages: a *base stage*, an *expansion stage*, and a *test stage*. Within the expansion stage, we further split the last portion (e.g., 1 day) as the validation set, while the earlier portion (e.g., 6 days) is used for expansion-stage training. For EPeMS, we strictly follow the experimental setup in Ma et al. (2025b) for consistency. For the other datasets, 80% of nodes are randomly selected as observed nodes in the base stage, providing sufficient history (L_1). The remaining 20% are treated as ***newadd*** nodes, appearing only in the expansion stage with short history ($L_2 \ll L_1$). Additionally, 5% of base nodes are randomly designated as deleted, while the rest remain as ***remain*** nodes. Table 5 summarizes detail statistics of datasets.

Table 5: Dataset statistics and characteristics

Dataset	Sample Rate	Stage Split	Node Expansion ($\tau_1 \to \tau_2$)	Trend Strength
EPeMS	5min	63d / 3d / 2d / 22d	296 $\to$ 447 (296 -0 + 151)	0.12
PEMS04	5min	35d / 6d / 1d / 17d	241 $\to$ 290 (241 -17 + 66)	0.08
SeaLoop	5min	18d / 6d / 1d / 3d	255 $\to$ 303 (255 -20 + 68)	0.11
NERL-AL	5min	122d / 6d / 1d / 53.5d	103 $\to$ 130 (103 -7 + 34)	0.71

B.2 BASELINE AND HYPER-PARAMETERS

We compare SNIPformer with four categories of existing solutions for expanding-node STF:

1. Models without node-specific prompting: **DLinear** (Zeng et al., 2023), **iTransformer** (Liu et al., 2024), **DUETformer** (Qiu et al., 2025).
2. STF models without node-specific modules: **GWNET**† (Wu et al., 2019), **STID**† (Shao et al., 2022), **STAEformer**† (Liu et al., 2023), **STOP** (Ma et al., 2025a), where † indicates removal of learnable node embeddings.
3. Continual learning methods: **STKEC** (Wang et al., 2023), **EAC** (Chen & Liang, 2025).
4. Fixed-node models after expansion: **STEV** (Ma et al., 2025b).

For SNIPformer, we set the PCA feature dimension to 24 (each for daily and weekly periods) and the spectral embedding dimension to 8. This leads to $k_{\text{pca}} = 24$, $n = 2$, $k_{\text{topo}} = k_{\text{delay}} = k_{\text{corr}} = 8$. Collectively, the dimension of $\boldsymbol{Z}_{\text{static}}$ is 72. During the expansion stage, the periodic priors of new nodes are constructed by mixing those of their three most similar remain nodes. Other priors are recomputed directly from the available expansion-stage data, except for the NREL-AL dataset, where the time-delayed interaction priors of new nodes are also obtained via mixing from old nodes. These design choices are made in accordance with the degree of temporal distribution shift observed in each dataset.

We use a 12-step history to predict the next 12 steps, correspond to 1 hour ahead prediction, which denotes $T = T' = 12$. The model dimension is 64 (32 for NREL-AL). Average results are reported after repeating the experiments no less than five times. **Code and data source are provided in the Supplementary Material**. Our experiments is under the PyTorch framework on a Linux server with one Intel(R) Xeon(R) Gold 5220 CPU and one 32GB NVIDIA Tesla V100-SXM2 GPU card.

For methods where embeddings increase with expansion (i.e., continual learning approaches) or rely on fixed node-specific learnable parameters (e.g., STEV), the case of deleted nodes is not explicitly considered. In our implementation on datasets with node removals, we carefully align the learnable embeddings across stages. This means that the parameters corresponding to deleted nodes are also discarded during the expansion stage, ensuring fair and consistent evaluation.

B.3 FULL RESULTS

Table 6 reports the full forecasting results. SNIPformer achieves consistently the best accuracy on EPeMS, PEMS04, SeaLoop datasets and has a second-best performance on NREL-AL dataset.

C USE OF LLMS

In this work, we used large language models solely for polishing grammar and improving clarity. All research ideas, methodologies, experiments, analyses, and conclusions were independently conceived and conducted by the authors. The LLM was not used for generating research content, experiments, results, or references.

Table 6: Full comparison of the expanding-node forecasting results of different methods and SNIPformer. MAPE values are scaled by 100 for presentation.

Model	Metric	EPeMS			PEMS04			SeaLoop			NREL-AL		
		All	Remain	New	All	Remain	New	All	Remain	New	All	Remain	New
DLinear	MAE	32.70	32.26	33.56	28.97	28.91	29.17	4.59	4.62	4.49	2.54	2.59	2.41
	MAPE	14.15	15.00	12.47	19.48	19.35	19.90	14.17	14.33	13.61	110.29	110.64	109.29
	RMSE	48.32	48.14	48.66	44.55	44.29	45.42	7.99	8.02	7.92	3.91	4.00	3.63
	MRE	0.10	0.10	0.10	0.13	0.13	0.13	0.08	0.09	0.08	0.21	0.21	0.21
iTransformer	MAE	26.83	26.65	27.16	24.76	24.74	24.83	4.29	4.31	4.21	1.94	1.97	1.83
	MAPE	10.91	11.39	9.96	16.08	16.02	16.29	12.71	12.80	12.40	100.71	101.10	99.60
	RMSE	41.40	41.22	41.73	39.62	39.34	40.52	7.54	7.56	7.46	3.36	3.44	3.12
	MRE	0.08	0.09	0.08	0.11	0.11	0.11	0.08	0.08	0.08	0.22	0.22	0.22
DUETformer	MAE	25.25	25.17	25.39	23.21	23.21	23.24	4.02	4.04	3.93	1.82	1.85	1.72
	MAPE	10.28	10.78	9.31	15.27	15.23	15.42	12.32	12.43	11.96	93.49	93.80	92.63
	RMSE	38.05	38.18	37.77	36.54	36.32	37.26	7.02	7.04	6.93	3.10	3.17	2.88
	MRE	0.08	0.08	0.07	0.10	0.10	0.10	0.07	0.07	0.07	0.21	0.21	0.21
GWNET	MAE	23.73	23.11	24.93	22.99	23.05	22.79	3.94	3.97	3.84	1.79	1.83	1.69
	MAPE	9.47	9.72	8.98	14.79	14.77	14.88	11.86	12.01	11.37	92.04	93.47	88.02
	RMSE	35.81	35.27	36.84	36.70	36.57	37.16	6.82	6.86	6.68	3.16	3.24	2.92
	MRE	0.07	0.07	0.07	0.10	0.10	0.10	0.07	0.07	0.07	0.23	0.23	0.23
STID	MAE	24.40	24.31	24.56	22.49	22.56	22.25	4.10	4.12	4.03	2.00	2.03	1.89
	MAPE	9.92	10.43	8.92	14.67	14.67	14.67	13.66	13.77	13.29	104.42	104.96	102.91
	RMSE	37.38	37.44	37.23	35.92	35.77	36.42	7.26	7.28	7.20	3.25	3.33	3.01
	MRE	0.08	0.08	0.07	0.10	0.10	0.10	0.08	0.08	0.07	0.17	0.17	0.17
STAEformer	MAE	24.86	24.66	25.27	22.95	23.03	22.67	4.15	4.17	4.09	1.91	1.95	1.81
	MAPE	9.94	10.38	9.07	14.76	14.71	14.91	13.08	13.15	12.83	87.52	87.75	86.86
	RMSE	38.34	38.26	38.50	36.75	36.63	37.14	7.38	7.39	7.37	3.29	3.37	3.05
	MRE	0.08	0.08	0.07	0.10	0.10	0.10	0.08	0.08	0.07	0.17	0.17	0.17
STOP	MAE	24.45	24.47	24.41	22.54	22.56	22.46	4.12	4.13	4.08	2.01	2.05	1.89
	MAPE	10.00	10.57	8.89	14.81	14.78	14.90	13.37	13.37	13.35	89.90	90.82	87.30
	RMSE	37.24	37.41	36.89	35.74	35.52	36.45	7.32	7.32	7.35	3.25	3.32	3.02
	MRE	0.08	0.08	0.07	0.10	0.10	0.10	0.08	0.08	0.07	0.16	0.16	0.16
STKEC	MAE	29.99	29.78	30.40	25.64	25.84	24.87	5.00	5.01	4.98	2.33	2.34	2.28
	MAPE	14.37	15.83	11.52	17.61	17.39	18.42	17.76	17.59	18.44	121.15	121.70	119.56
	RMSE	42.91	43.05	42.64	39.55	39.74	38.73	8.14	8.13	8.16	3.62	3.66	3.51
	MRE	0.09	0.10	0.09	0.12	0.12	0.12	0.09	0.09	0.09	0.21	0.21	0.21
EAC	MAE	28.74	28.23	29.75	24.05	24.27	23.21	4.72	4.73	4.72	2.16	2.17	2.14
	MAPE	12.24	12.83	11.06	18.14	17.74	19.58	17.00	16.62	18.43	114.49	115.24	112.29
	RMSE	40.33	39.80	41.35	36.51	36.79	35.41	7.82	7.81	7.86	3.37	3.39	3.32
	MRE	0.09	0.09	0.09	0.11	0.11	0.11	0.09	0.09	0.09	0.20	0.20	0.20
STEV	MAE	22.90	22.35	23.97	20.55	20.42	21.01	3.92	3.95	3.84	**1.57**	**1.58**	**1.53**
	MAPE	9.45	9.81	8.75	14.77	14.65	15.18	12.56	12.71	12.10	**67.52**	**68.33**	**65.21**
	RMSE	34.51	33.95	35.60	32.46	32.13	33.53	6.62	6.66	6.51	2.88	2.93	2.73
	MRE	0.07	0.07	0.07	0.09	0.09	0.09	0.07	0.07	0.07	**0.15**	**0.14**	0.15
SNIPformer (ours)	MAE	**22.05**	**21.39**	**23.35**	**19.20**	**19.22**	**19.10**	**3.46**	**3.47**	**3.42**	1.62	1.65	1.55
	MAPE	**8.95**	**9.20**	**8.46**	**12.68**	**12.64**	**12.81**	**10.50**	**10.62**	**10.10**	88.75	90.38	84.13
	RMSE	**33.91**	**33.16**	**35.33**	**31.02**	**30.87**	**31.54**	**6.10**	**6.14**	**5.97**	**2.87**	**2.92**	**2.71**
	MRE	**0.07**	**0.07**	**0.07**	**0.09**	**0.09**	**0.09**	**0.06**	**0.06**	**0.06**	0.15	0.15	**0.15**