

# 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 (EVTSTF) 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.

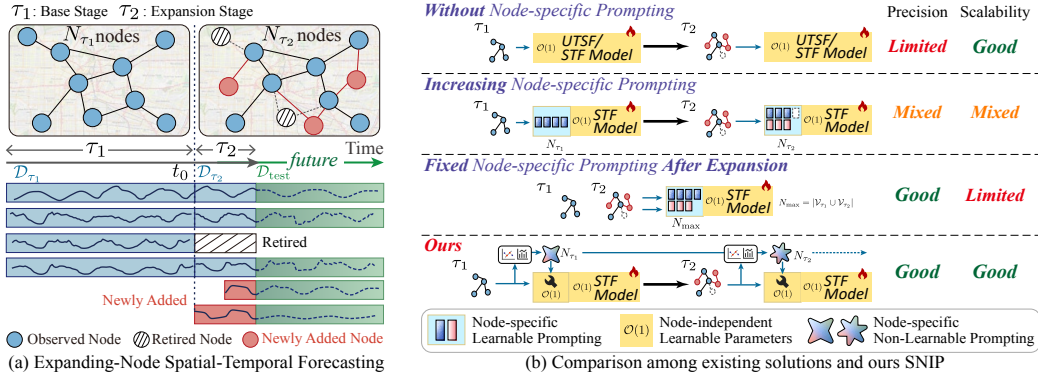


Figure 1: Examples of expanding-node spatial-temporal forecasting and different solutions. (a) Sensor nodes may added, retired or replaced in the expansion stage. (b) Comparison of three existing solution paradigms with our proposed SNIP framework.

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 environments**, 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. **In addition, node-count-agnostic models like literature (Altieri et al., 2024; Li et al., 2018; Zheng et al., 2020) can also handle changing node sets, but they mainly rely on short-window inputs and ignore node-specific inherent heterogeneity.** 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  $\tau$ , denoted as  $\mathcal{G}_\tau = (\mathcal{V}_\tau, \mathcal{E}_\tau)$ , where  $\mathcal{V}_\tau = \{v_1, v_2, \dots, v_{N_\tau}\}$  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  $\mathbf{A}_\tau \in \mathbb{R}^{N_\tau \times N_\tau}$ ,  $N_\tau = |\mathcal{V}_\tau|$ , 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

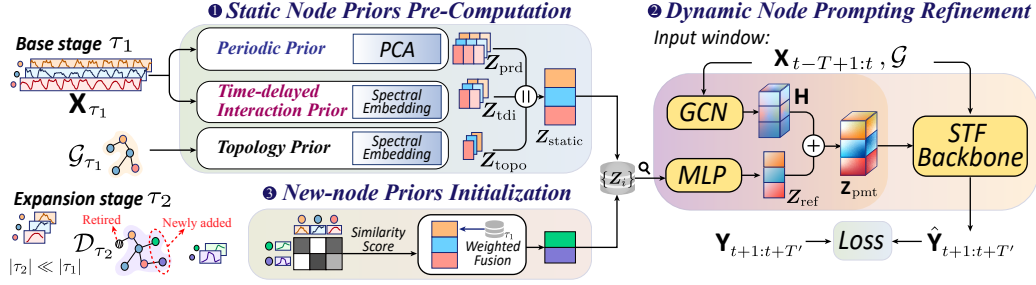


Figure 2: Overall framework of our proposed SNIP.

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 the parameter set  $\Theta$  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.

#### 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  $\tau$  with length  $L$ ,  $\mathbf{X} \in \mathbb{R}^{L \times N \times C}$  denote the historical sequence for  $N$  nodes. For clarity, we describe the single-feature case ( $C = 1$ ) below, which naturally extends to multi-channel inputs by concatenation. We specify a set of cycle lengths  $\{p_1, p_2, \dots, p_n\}$  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  $\bar{\mathbf{Z}}^{(j)} \in \mathbb{R}^{N \times k_{\text{pca}}}$ ,  $k_{\text{pca}}$  represents the value of a low dimensionality. Finally, we concatenate results across all  $n$  cycle

lengths to obtain the periodic priors:

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

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  $\mathbf{A} \in \mathbb{R}^{N \times N}$ , one can construct the normalized Laplacian:  $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ , where  $\mathbf{D}$  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  $k_{\text{topo}}$  eigenvalues of  $\mathbf{L}$ . This can be formulated as:

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

where  $\mathbf{u}_1, \dots, \mathbf{u}_{k_{\text{topo}}}$  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 reflect 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:  $\mathbf{R}(\delta)$ . 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  $\mathbf{Z}_{\text{tdi}}$ :

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

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  $z_{i*}^{(t)}$  for each node  $i$ , which maximizes predictive accuracy. This configuration can be decomposed as:  $z_{i*}^{(t)} = \mathbf{q}_i + \mathbf{r}_i^{(t)}$ ,  $\mathbf{r}_i^{(t)} = g(\mathbf{x}_j^{(t)}, j \in \mathbb{N}_{(i)})$ , where  $\mathbf{q}_i$  represents spatially intrinsic characteristics of node  $i$  (time-invariant reference),  $\mathbf{r}_i^{(t)}$  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.*

This decomposition allows  $\mathbf{q}_i$  to represent slowly varying identity, with  $\mathbf{r}_i^{(t)}$  capturing fast, context-dependent deviations. In our study, long horizon priors built via multi cycle PCA and spectral embeddings preserve between node variance and aim to capture inherent node properties in  $\mathbf{q}_i$ . The refinement  $\mathbf{r}_i^{(t)}$  then adapts these identities to current conditions during training. Through this, the learnable model only needs to fit  $\mathbf{r}_i^{(t)}$ . 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:  $\mathbf{Z}_{\text{ref}} = \text{MLP}([\mathbf{Z}_{\text{prd}}, \mathbf{Z}_{\text{topo}}, \mathbf{Z}_{\text{tdi}}]) \in \mathbb{R}^{N \times d}$ . 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:  $\mathbf{H}_{t,:,:} = \text{DiffGCN}(\mathbf{X}_{t,:,:}^{\text{emb}}, \mathbf{A})$ , where  $\mathbf{X}^{\text{emb}} \in \mathbb{R}^{T \times N \times d}$  is the series embedding and  $\mathbf{A}$  is the adjacency matrix. To simulate potential changes in graph topology during the expansion stage, we further apply edge dropout to  $\mathbf{A}$  during training, enhancing robustness to evolving structures. The final adaptive embedding is obtained by combining static refinement and dynamic aggregation:

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

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. This adjustment block, an MLP and a diffusion graph convolution, is trained jointly with the STF backbone in both base stage and expansion stage and is executed at test stage to produce the final prompts used by the predictor.

**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. 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  $\mathbf{P}$  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  $s_{i,j} = P_{i,j} / \sum_{j \in \mathcal{V}_{\text{remain}}} P_{i,j}$ ,  $v_i \in \mathcal{V}_{\text{new}}$ ;  $v_j \in \mathcal{V}_{\text{remain}}$ . Let  $\mathbb{P}_{\text{all}} = \{\text{prd}, \text{topo}, \text{tdi}\}$  denote the *candidate* prior types, and let  $\mathbb{P}_i \subseteq \mathbb{P}_{\text{all}}$  be the subset actually constructed for new node  $i$ . For any prior type  $\mathbb{P} \in \mathbb{P}_i$  with feature matrix  $\mathbf{Z}_{\mathbb{P}} \in \mathbb{R}^{N \times d_{\mathbb{P}}}$ , the prior of node  $i$  is obtained by mixing the priors of its top similar remain nodes:  $(\mathbf{z}_{\mathbb{P}})_i = \sum_{j \in \mathcal{N}_{k(i)}} s_{i,j} (\mathbf{z}_{\mathbb{P}})_j$ , where  $\mathcal{N}_{k(i)}$  denotes the top- $k$  most similar remain nodes to  $i$ .

Not all prior types in  $\mathbb{P}_{\text{all}}$  require mixing. In practice, a simple similarity threshold determines whether a new node recomputes its priors or mixes them from similar remain nodes. In our experiments, periodic priors are mixed from old nodes due to insufficient cycle history, whereas topological and time-delayed interaction priors are recomputed from short-term observations because they reflect recent and rapidly varying spatial dependencies. Remain nodes simply reuse their base-stage priors. Most prior works do not address nodes removed in the expansion stage. In our framework, discarded nodes require no priors in the new period, and because model parameters are fully decoupled from node identity, no redundant parameters persist. This avoids parameter waste and enhances flexibility for evolving network structures.

### 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. Appendix A.3 details the complete model structure, the implementation of the prior, and the algorithms for base-stage pre-training and expansion-stage fine-tuning.

## 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. Beyond single-stage expansion, we also test SNIP in multi-stage expanding-node scenarios using the PEMS-Stream (Chen et al., 2021) and Air-Stream (Chen & Liang, 2025) datasets, where the node set evolves over several consecutive expansion periods. Moreover, SNIP is evaluated in multi-horizon forecasting (24/48/96-step) settings. Detailed results and analyses are provided in Appendix B.5 and B.6.

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 B.4 with consistent conclusions.

Table 1: Dataset statistics and characteristics

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

**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), **DUET** (Qiu et al., 2025). 2) STF models without node-specific modules: **DCRNN** (Li et al., 2018), **GWNET**<sup>†</sup> (Wu et al., 2019), **GMAN** (Zheng et al., 2020), **STID**<sup>†</sup> (Shao et al., 2022), **STAEformer**<sup>†</sup> (Liu et al., 2023), **TESTAM**<sup>†</sup> (Lee & Ko, 2024), **STOP** (Ma et al., 2025a), where <sup>†</sup> 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

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

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
	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
DUET	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
GWN <sup>†</sup>	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 <sup>†</sup>	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 <sup>†</sup>	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	<u>22.90</u>	<u>22.35</u>	<u>23.97</u>	<u>20.55</u>	<u>20.42</u>	<u>21.01</u>	<u>3.92</u>	<u>3.95</u>	<u>3.84</u>	<u>1.57</u>	<u>1.58</u>	<u>1.53</u>
	RMSE	<u>34.51</u>	<u>33.95</u>	<u>35.60</u>	<u>32.46</u>	<u>32.13</u>	<u>33.53</u>	<u>6.62</u>	<u>6.66</u>	<u>6.51</u>	<u>2.88</u>	<u>2.93</u>	<u>2.73</u>
SNIPformer (ours)	MAE	<b>22.05</b>	<b>21.39</b>	<b>23.35</b>	<b>19.20</b>	<b>19.22</b>	<b>19.10</b>	<b>3.46</b>	<b>3.47</b>	<b>3.42</b>	<b>1.62</b>	<b>1.65</b>	<b>1.55</b>
	RMSE	<b>33.91</b>	<b>33.16</b>	<b>35.33</b>	<b>31.02</b>	<b>30.87</b>	<b>31.54</b>	<b>6.10</b>	<b>6.14</b>	<b>5.97</b>	<b>2.87</b>	<b>2.92</b>	<b>2.71</b>

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. More results under the multi-stage expansion and multi-horizon settings are provided in Appendix B.5 and B.6.

**Generality across architectures.** To validate SNIP’s model-agnostic design, we integrate it into four categories of backbones: MLP based (DLinear, STID), graph based (DCRNN, GWN<sup>†</sup>, GMAN), attention based (iTransformer, DUET, STAEformer<sup>†</sup>), and a hybrid architecture TESTAM<sup>†</sup>, three experts for temporal modeling, static graph spatio-temporal modeling, and dynamic graph spatio-temporal modeling). Additionally, we employ an attention-based module from STGP (Hu et al., 2024) as a prompting baseline, referring to it as AttP in this experiment. Table 3 reports the MAE results of (i) the **original backbone**, (ii) **backbone + AttP**, and (iii) **backbone + SNIP** from three representative backbones. Full results are provided in Appendix B.4 (Table 10 and Table 11). 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. 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.

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	1.95	1.99	1.84
	+ SNIP	<b>24.67</b>	<b>24.14</b>	<b>1.84</b>	<b>1.88</b>	<b>1.74</b>
GWN <sup>†</sup>	23.73	23.11	24.93	1.79	1.83	1.69
	+ AttP	23.75	23.13	1.79	1.82	1.69
	+ SNIP	<b>23.41</b>	<b>22.79</b>	<b>1.77</b>	<b>1.81</b>	<b>1.66</b>
STID <sup>†</sup>	24.40	24.31	24.56	2.00	2.03	1.89
	+ AttP	24.35	24.26	2.01	2.05	1.90
	+ SNIP	<b>21.84</b>	<b>21.13</b>	<b>1.86</b>	<b>1.89</b>	<b>1.77</b>



### 5.3 ABLATION AND HYPER-PARAMETER STUDIES

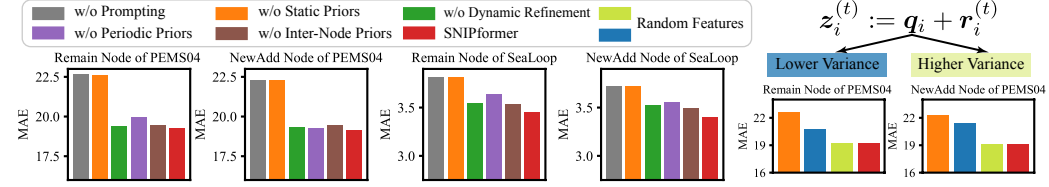


Figure 3: Ablation Results. Left: Comparison of contribution of different components. Right: Performance of using random features with high and low variance as static priors.

**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.

**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.

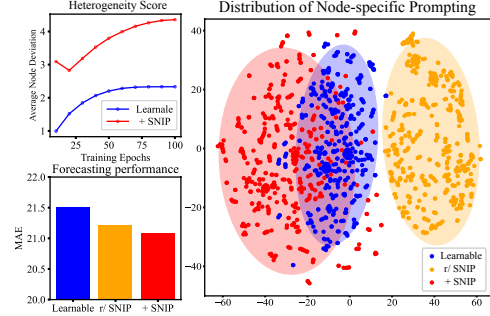


Figure 4: Contribution of SNIP to the STID model on PEMS04 dataset. Right: Distribution of node-specific prompting after dimensionality reduction via t-SNE.

**Hyper-parameter study.** We examine two groups of hyper-parameters that control prior construction:  $k_{pca}$  and the number of periodicities  $n$  for the periodic priors, and  $k_{topo}$ ,  $k_{delay}$ ,  $k_{corr}$  together with the Hann window size of Welch method for the interaction priors. Figure 5 reports RMSE on PEMS04, and full results are given in Appendix B.3. In Figure 5(a), varying  $k_{pca}$  shows that performance is stable across a wide range. We set  $k_{pca} = 24$  to keep preprocessing cost low while retaining prediction accuracy. In addition, using both daily and weekly periods ( $n = 2$ ) consistently outperforms using either alone, which matches the multi seasonal nature of traffic data. In Figure 7(b), increasing  $k_{topo}$ ,  $k_{delay}$ ,  $k_{corr}$  beyond small values brings negligible gains but higher training and preprocessing cost. For the Hann window size, performance varies within a narrow band and stabilizes as the window grows. We set it to  $T$  (12 in our task) in the main experiments as a balanced choice that matches the forecasting horizon and preserves time localization.

Moreover, we further evaluate different ratios of new and retired nodes to demonstrate robustness under diverse real deployment scenarios, with results reported in Appendix B.3. In addition, we

visualize node distributions and forecasting performance under three expansion protocols (following STEV and detailed in Appendix B.1), and present the corresponding results in Appendix B.7.

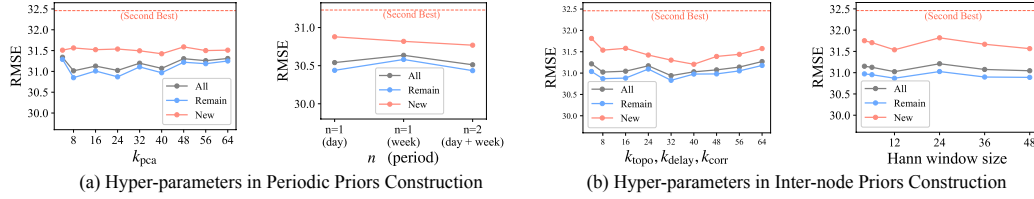


Figure 5: Hyper-parameter study in PEMS04 dataset.

#### 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 <sup>†</sup> +SNIP	↓%
<b>Pre-computation</b>	Augmentation	Static Priors	-	-	Static Priors	-
Time Cost (min)	0.21	2.61	-	-	2.61	-
<b>Training</b>	$(\tau_1, \tau_2)$	$(\tau_1 \rightarrow \tau_2)$		$(\tau_1)$	$(\tau_1)$	
Time (s/epoch)	<u>325.42</u>	<b>28.26</b> → <b>1.42</b>	↓91.3%	<b>132.03</b>	<u>134.77</u>	↑2.0%
Footprint (MB)	<u>31430</u>	<b>1466</b> → <b>2358</b>	↓92.5%	<b>8130</b>	<u>8296</u>	↑2.0%
<b>Inference</b>	Time(s)					
$(\tau_2)$	<u>20.46</u>	<b>1.05</b>	↓94.9%	Invalid	1.37	↑ Feasibility
MAE	<u>22.90</u>	<b>22.05</b>	↓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.
- Massimiliano Altieri, Roberto Corizzo, and Michelangelo Ceci. GAP-LSTM: Graph-Based Auto-correlation Preserving Networks for Geo-Distributed Forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11773–11787, September 2024. ISSN 2162-2388. doi: 10.1109/TNNLS.2024.3398441. URL <https://ieeexplore.ieee.org/document/10533686>.
- 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=FRzCilK7I&noteId=RDxGMROaMj>.
- Xu Chen, Junshan Wang, and Kunqing Xie. TrafficStream: A Streaming Traffic Flow Forecasting Framework Based on Graph Neural Networks and Continual Learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 3620–3626, Montreal, Canada, August 2021. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-9-6. doi: 10.24963/ijcai.2021/498. URL <https://www.ijcai.org/proceedings/2021/498>.
- Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yin Hai 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, Qunjun 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 V2, 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>.
- Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. GMAN: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1234–1241, 2020.
- 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  $\tau$  with a length of  $L$ . Let  $\mathbf{X} \in \mathbb{R}^{L \times N}$  denote the historical sequence of for  $N$  nodes. We specify a set of cycle lengths  $\{p_1, p_2, \dots, p_n\}$ . For node  $i$  and a given cycle length  $p_j$ , we partition its sequence  $\mathbf{X}_{:,i} \in \mathbb{R}^L$  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:

$$\mathbb{S}_i^{(j)} = \text{Partition}(\mathbf{X}_{:,i}, p_j) = \left\{ \mathbf{X}_{:,i}[(m-1)p_j + 1 : mp_j] \mid m = 1, \dots, M_j \right\}, \quad (5)$$

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:  $\tilde{\mathbf{x}} = (\mathbf{x} - \mu_i^{(j)})/\sigma_i^{(j)}$ ,  $\mathbf{x} \in \mathbb{S}_i^{(j)}$ , 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  $\tilde{\mathbf{X}}_m^{(j)}$  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  $\mathbf{U}^{(j)} \in \mathbb{R}^{p_j \times k_{\text{pca}}}$  from the top  $k_{\text{pca}}$  eigenvectors of the covariance matrix of  $\tilde{\mathbf{X}}_m^{(j)}$ . Then the segment-level low-dimensional representation is computed, and the  $M_j$  representations are the averaged across segments:

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

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

$$\mathbf{Z}_{\text{prd}} = \text{Concat}(\bar{\mathbf{Z}}^{(1)}, \bar{\mathbf{Z}}^{(2)}, \dots, \bar{\mathbf{Z}}^{(n)}) \in \mathbb{R}^{N \times (n \cdot k_{\text{pca}})}. \quad (7)$$

Figure 6 illustrates this construction.

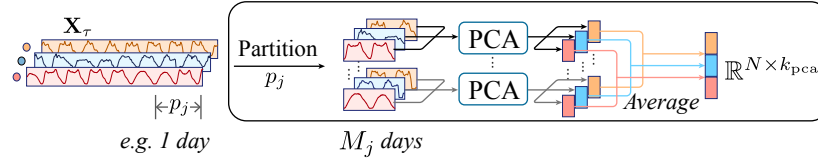


Figure 6: An illustration for the periodic prior construction under a cycle length  $p_j$ .

## 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  $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^L$  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  $T$ :

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

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  $i$ ,  $\nu$  is the frequency variable, and  $*$  denotes complex conjugation. The cross-correlation function is obtained by inverse FFT:

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

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

$$\Delta_{ij} = \arg \max_{\delta} |R_{ij}(\delta)|, \quad P_{ij} = \max_{\delta} |R_{ij}(\delta)|. \quad (10)$$

where  $\Delta$  is the delay matrix recording absolute dominant lags, and  $P$  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  $Z_{\text{tdi}}$ :

$$Z_{\text{tdi}} = \text{Concat}(\Phi(\Delta, k_{\text{delay}}), \Phi(P, k_{\text{corr}})). \quad (11)$$

### A.3 ARCHITECTURE OF SNIPFORMER AND ALGORITHM OF PIPELINES

Figure 7 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.

To facilitate reproducibility and implementation, we provide detailed algorithmic descriptions of the proposed SNIP framework, covering prior construction, base stage training, and expansion stage adaptation. These procedures are summarized in Algorithm 1, Algorithm 2, Algorithm 3, and Algorithm 4, respectively.

---

#### Algorithm 1 OfflinePriorConstruction

---

**Input:**  $\mathbf{X}_{\text{hist}} \in \mathbb{R}^{N \times L \times C}$ , graph matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$ , config  $\text{cfg}$

**Output:**  $Z_{\text{prd}}, Z_{\text{topo}}, Z_{\text{tdi}}$

1: **Periodic prior via PCA**

2: **for**  $j = 1$  to  $n$  **do**

3:  $Z_{\text{prd}}^{(j)} \leftarrow \text{SEGMENTPCA}(\mathbf{X}_{\text{hist}}, p_j, k_{\text{pca}}) \triangleright Z_{\text{prd}}^{(j)} \in \mathbb{R}^{N \times k_{\text{pca}}} \triangleright p_j$  is the  $j$ -th cycle length

4: **end for**

5:  $Z_{\text{prd}} \leftarrow \text{concat}(Z_{\text{prd}}^{(1)}, \dots, Z_{\text{prd}}^{(n)})$

6:  $Z_{\text{prd}} \in \mathbb{R}^{N \times (n \cdot k_{\text{pca}})}$

7: **Topology prior via spectral embedding**

8:  $Z_{\text{topo}} \leftarrow \text{SPECTRALEMBED}(\mathbf{A}, k_{\text{topo}})$

9: **Time delayed interaction prior via spectral embedding**

10: Initialize matrices  $\Delta, P \in \mathbb{R}^{N \times N}$  with zeros

11: **for** each pair  $(i, j)$  **do**

12:  $Q_{ij} \leftarrow \text{WELCHCSD}(\mathbf{X}_{\text{hist}}[i, :], \mathbf{X}_{\text{hist}}[j, :], \text{Hann}, \text{cfg.nperseg}, \text{cfg.noverlap})$

13:  $R_{ij} \leftarrow \text{ifft}(Q_{ij})$ , real part, fftshift to centered lags

14:  $\Delta[i, j] \leftarrow \arg \max_{\delta} |R_{ij}(\delta)|$

15:  $P[i, j] \leftarrow \max_{\delta} |R_{ij}(\delta)|$

16: **end for**

17:  $Z_{\text{tdi}} \leftarrow \text{concat}(\text{SPECTRALEMBED}(\Delta, k_{\text{delay}}), \text{SPECTRALEMBED}(P, k_{\text{corr}}))$

18: **return**  $Z_{\text{prd}}, Z_{\text{topo}}, Z_{\text{tdi}}$

---

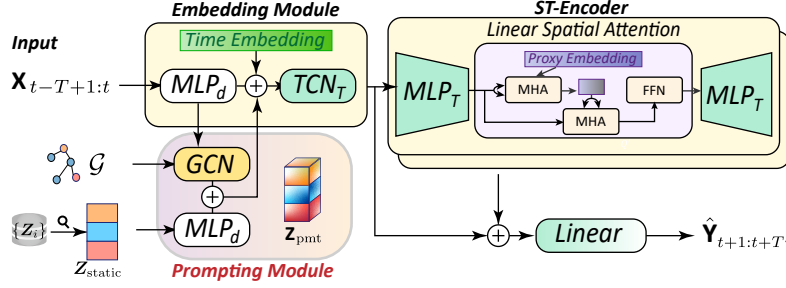


Figure 7: The architecture of SNIPformer.

**Algorithm 2 BaseStagePretrain**

**Input:** base windows  $\{(X_{t-L+1:t}, Y_t)\}$ , priors  $Z_{\text{prd}}, Z_{\text{topo}}, Z_{\text{tdi}}$ , backbone  $f_\theta$ , adjuster  $g_\phi$   
**Output:** trained parameters  $\theta^*, \phi^*$

- 1: Initialize  $\theta, \phi$
- 2:  $Z_{\text{static}} = \text{Concat}(Z_{\text{prd}}, Z_{\text{topo}}, Z_{\text{tdi}})$
- 3: **for** each mini batch  $\mathcal{B}$  **do**
- 4:    $Z_{\text{pmt}} \leftarrow g_\phi(Z_{\text{static}}, X_{t-L+1:t})$   $\triangleright g_\phi$  includes MLP and GCN layers
- 5:    $\hat{Y}_{t+1:t+T'} \leftarrow f_\theta(X_{t-L+1:t}, Z_{\text{pmt}})$
- 6:    $\mathcal{L}_{\text{pred}} \leftarrow \ell(\hat{Y}_{t+1:t+T'}, Y_{t+1:t+T'})$
- 7:   Update  $\theta, \phi$  by minimizing  $\mathcal{L}_{\text{pred}}$
- 8: **end for**
- 9: **return**  $\theta^* \leftarrow \theta, \phi^* \leftarrow \phi$

**Algorithm 3 ExpansionStageAdapt and Inference**

**Input:** expansion data, graph matrix  $A_{\tau_2}$ , remain set  $\mathcal{V}_{\text{rem}}$ , new set  $\mathcal{V}_{\text{new}}$ ,  
 cached priors  $Z_{\text{prd}}, Z_{\text{topo}}, Z_{\text{tdi}}$ , trained params  $\theta^*, \phi^*$ ,  
 mixing flags  $(b_{\text{prd}}, b_{\text{topo}}, b_{\text{delay}}) \in \{0, 1\}^3$ , hyperparams  $k_{\text{pca}}, k_{\text{topo}}, k_{\text{delay}}$ ,  
 periodic set  $\{p_j\}$ , top  $k$ , similarity function  $s(\cdot, \cdot)$   
**Output:** adapted params  $\theta^\dagger, \phi^\dagger$ , updated priors  $Z_{\text{prd}}^\dagger, Z_{\text{topo}}^\dagger, Z_{\text{tdi}}^\dagger$

- 1: **Update static priors at expansion stage**
- 2:  $Z_{\text{prd}}^\dagger, Z_{\text{topo}}^\dagger, Z_{\text{tdi}}^\dagger \leftarrow \text{EXPANSIONPRIORSUPDATE}(\text{expansion data}, A_{\tau_2}, \mathcal{V}_{\text{rem}}, \mathcal{V}_{\text{new}},$   
 $Z_{\text{prd}}, Z_{\text{topo}}, Z_{\text{tdi}}, b_{\text{prd}}, b_{\text{topo}}, b_{\text{delay}},$   
 $\{p_j\}, k_{\text{pca}}, k_{\text{topo}}, k_{\text{delay}}, k, s)$   $\triangleright$  in Algorithm 4
- 3: **Concatenate priors and initialize parameters**
- 4:  $Z_{\text{static}}^\dagger \leftarrow \text{concat}[Z_{\text{prd}}^\dagger, Z_{\text{topo}}^\dagger, Z_{\text{tdi}}^\dagger]$
- 5:  $\theta^\dagger \leftarrow \theta^*, \phi^\dagger \leftarrow \phi^*$
- 6: **Expansion stage finetuning**
- 7: **for** each mini batch  $\mathcal{B}$  from expansion windows **do**
- 8:    $Z_{\text{pmt}}^\dagger \leftarrow g_{\phi^\dagger}(Z_{\text{static}}^\dagger, X_{t-L+1:t})$
- 9:    $\hat{Y}_t \leftarrow f_{\theta^\dagger}(X_{t-L+1:t}, Z_{\text{pmt}}^\dagger)$
- 10:   Update  $\phi^\dagger$  and  $\theta^\dagger$  by loss  $\mathcal{L}_{\text{pred}}$
- 11: **end for**
- 12: **Inference**
- 13: **for** each test window **do**
- 14:    $Z_{\text{pmt}}^\dagger \leftarrow g_{\phi^\dagger}(Z_{\text{static}}^\dagger, X_{t-L+1:t})$
- 15:   Output  $f_{\theta^\dagger}(X_{t-L+1:t}, Z_{\text{pmt}}^\dagger)$
- 16: **end for**
- 17: **return**  $\theta^\dagger, \phi^\dagger, Z_{\text{prd}}^\dagger, Z_{\text{topo}}^\dagger, Z_{\text{tdi}}^\dagger$

**Algorithm 4** ExpansionPriorsUpdate

---

**Input:** expansion data, graph matrix  $\mathbf{A}_{\tau_2}$ , remain set  $\mathcal{V}_{\text{rem}}$ , new set  $\mathcal{V}_{\text{new}}$ , cached priors  $\mathbf{Z}_{\text{prd}}, \mathbf{Z}_{\text{topo}}, \mathbf{Z}_{\text{tdi}}$ ,  
 mixing flags  $(b_{\text{prd}}, b_{\text{topo}}, b_{\text{delay}}) \in \{0, 1\}^3$ , periodic set  $\{p_j\}$ , dims  $k_{\text{pca}}, k_{\text{topo}}, k_{\text{delay}}$ ,  
 top  $k$ , similarity function  $s(\cdot, \cdot)$

**Output:** updated priors  $\mathbf{Z}_{\text{prd}}^\dagger, \mathbf{Z}_{\text{topo}}^\dagger, \mathbf{Z}_{\text{tdi}}^\dagger$

---

- 1: **Partition expansion data**
- 2:  $\mathbf{X}_{\text{rem}} \leftarrow \text{gather}(\text{expansion data}, \mathcal{V}_{\text{rem}}), \mathbf{X}_{\text{new}} \leftarrow \text{gather}(\text{expansion data}, \mathcal{V}_{\text{new}})$
- 3: **Similarity matrix and top- $k$  weights**
- 4:  $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}_{\text{new}}| \times |\mathcal{V}_{\text{rem}}|} \leftarrow \text{SIMILARITYMATRIX}(\mathbf{X}_{\text{new}}, \mathbf{X}_{\text{rem}}, s)$
- 5:  $\mathbf{M} \leftarrow \text{TOPKMASK}(\mathbf{S}, k)$   $\triangleright$  row wise top  $k$  indicator
- 6:  $\alpha \leftarrow \text{ROWNORMALIZE}(\mathbf{S} \odot \mathbf{M})$   $\triangleright \sum_j \alpha_{ij} = 1$  where  $M_{ij} = 1$
- 7: **Periodic prior update**
- 8: **if**  $b_{\text{prd}} = 0$  **then**
- 9:    $\mathbf{Z}_{\text{prd}}^\dagger \leftarrow \text{SEGMENTPCA}(\text{expansion data}, \{p_j\}, k_{\text{pca}})$
- 10: **else**
- 11:    $\mathbf{Z}_{\text{prd}}^\dagger[\mathcal{V}_{\text{rem}}, :] \leftarrow \mathbf{Z}_{\text{prd}}[\mathcal{V}_{\text{rem}}, :], \mathbf{Z}_{\text{prd}}^\dagger[\mathcal{V}_{\text{new}}, :] \leftarrow \alpha \cdot \mathbf{Z}_{\text{prd}}[\mathcal{V}_{\text{rem}}, :]$
- 12: **end if**
- 13: **Topology prior update**
- 14: **if**  $b_{\text{topo}} = 0$  **then**
- 15:    $\mathbf{Z}_{\text{topo}}^\dagger \leftarrow \text{SPECTRALEMBED}(\mathbf{A}_{\tau_2}, k_{\text{topo}})$
- 16: **else**
- 17:    $\mathbf{Z}_{\text{topo}}^\dagger[\mathcal{V}_{\text{rem}}, :] \leftarrow \mathbf{Z}_{\text{topo}}[\mathcal{V}_{\text{rem}}, :],$
- 18:    $\mathbf{Z}_{\text{topo}}^\dagger[\mathcal{V}_{\text{new}}, :] \leftarrow \alpha \cdot \mathbf{Z}_{\text{topo}}[\mathcal{V}_{\text{rem}}, :]$
- 19: **end if**
- 20: **Time delayed interaction prior update**
- 21: **if**  $b_{\text{delay}} = 0$  **then**
- 22:    $\tilde{\Delta}, \tilde{\mathbf{P}} \leftarrow \text{BUILDDELAYCORRMATRIX}(\text{expansion data})$   $\triangleright$  Welch CSD method
- 23:    $\mathbf{Z}_{\text{tdi}}^\dagger \leftarrow \text{concat}(\text{SPECTRALEMBED}(\tilde{\Delta}, k_{\text{delay}}), \text{SPECTRALEMBED}(\tilde{\mathbf{P}}, k_{\text{delay}}))$
- 24: **else**
- 25:    $\mathbf{Z}_{\text{tdi}}^\dagger[\mathcal{V}_{\text{rem}}, :] \leftarrow \mathbf{Z}_{\text{tdi}}[\mathcal{V}_{\text{rem}}, :], \mathbf{Z}_{\text{tdi}}^\dagger[\mathcal{V}_{\text{new}}, :] \leftarrow \alpha \cdot \mathbf{Z}_{\text{tdi}}[\mathcal{V}_{\text{rem}}, :]$
- 26: **end if**
- 27: **return**  $\mathbf{Z}_{\text{prd}}^\dagger, \mathbf{Z}_{\text{topo}}^\dagger, \mathbf{Z}_{\text{tdi}}^\dagger$

---

**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.,  $C = 1$ .

**Stage and Node Division.** 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.

Although these real-world spatial-temporal datasets provide sensor locations or physical adjacency, none of them include authoritative deployment or decommission timestamps, and, to the best of our knowledge, there is currently no publicly available benchmark that reflects truly incremental sensor deployments. This limitation has also been emphasized in STEV, which constructs expanding-node settings through spatial or internal partitioning due to the absence of real EVTS logs.

Following this practice, for EPeMS we adopt the same *area-expansion* protocol in STEV (Ma et al., 2025b), where new nodes correspond to sensors hypothetically deployed in newly covered regions. For the remaining datasets, despite available spatial coordinates or topological links, installation and retirement records are not provided. Therefore, we simulate the *internal expansions* in STEV, assigning a subset of nodes as remain, deleted, and newadd, and restricting the observation horizon of new nodes to emulate short post-deployment histories. This protocol follows the commonly adopted assumption that adding or removing sensors does not change the underlying physical process being monitored, making such synthetic expansions a reasonable proxy for evolving networks.

Looking forward, we believe that constructing benchmarks with true deployment logs or continuous sensor rollouts represents an important direction for the community. Such datasets would provide more realistic evaluation settings and further facilitate research on expanding-node spatial-temporal forecasting.

Table 5: Dataset statistics and characteristics

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

We additionally adopt two streaming benchmarks, Air-Stream and PEMS-Stream, which are commonly used in continual spatio temporal forecasting, to assess the performance of SNIP under multiple expansion phases:

- Air-Stream (Chen & Liang, 2025): an air quality index dataset introduced in the EAC (Chen & Liang, 2025) work, constructed from observations recorded at environmental monitoring stations across China, and it models a growing sensor network without node deletions.
- PEMS-Stream (Chen et al., 2021): a traffic flow dataset from the Traffic-Stream (Chen et al., 2021) work, built from District 3 in California, which assumes no deleted nodes.

Both datasets contain multiple node expansion stages and Air-Stream involves more than one thousand sensors, which allows evaluation at larger spatial scales. Detailed statistics of these streaming datasets are summarized in Table 6.

Table 6: Details of stream datasets with multi-stage node expansion

Dataset	Sample Rate	Stage	Node Expansion	Average Growth Rate	Frames
Air-Stream	1 hour	4	1087 $\rightarrow$ 1154 $\rightarrow$ 1193 $\rightarrow$ 1202	3.43%	8578 $\rightarrow$ 8619 $\rightarrow$ 8378 $\rightarrow$ 8490
PEMS-Stream	5 min	7	655 $\rightarrow$ 715 $\rightarrow$ 786 $\rightarrow$ 822 $\rightarrow$ 834 $\rightarrow$ 850 $\rightarrow$ 871	4.92%	8928 $\rightarrow$ 8928 $\rightarrow$ 8928 $\rightarrow$ 8928 $\rightarrow$ 8928 $\rightarrow$ 8928 $\rightarrow$ 8928



## 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), **DUET** (Qiu et al., 2025).
2. STF models without node-specific modules: **DCRNN** (Li et al., 2018), **GWNET**<sup>†</sup> (Wu et al., 2019), **GMAN** (Zheng et al., 2020), **STID**<sup>†</sup> (Shao et al., 2022), **STAEformer**<sup>†</sup> (Liu et al., 2023), **TESTAM**<sup>†</sup> (Lee & Ko, 2024), **STOP** (Ma et al., 2025a), where <sup>†</sup> 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  $\mathbf{Z}_{\text{static}}$  is 72. The hyper-parameters study is in Appendix B.3. 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 means  $\mathbb{P}_i = \{\text{prd}\}$  for EPeMS, PEMS04 and SeaLoop datasets, while  $\mathbb{P}_i = \{\text{prd}, \text{tdi}\}$  for NREL-AL dataset. 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 ON HYPER-PARAMETERS STUDY

We report a hyperparameter sensitivity study. The analysis is organized into two groups of design choices that correspond to the construction of the periodic prior and the node interaction priors. For each group, a grid search is carried out on PEMS04 and the MAE and RMSE curves are visualized in the sensitivity plots, while all other settings are fixed to their default values.

First, the hyperparameters for the periodic prior are examined, namely the number of periodic PCA components  $k_{\text{pca}}$  and the number of explicit periods  $n$ . Results are in Figure 8(a). The parameter  $k_{\text{pca}}$  controls the dimensionality of the periodic PCA subspace that summarizes day level and week level seasonal patterns. The sensitivity plots show that both MAE and RMSE remain in a narrow band when  $k_{\text{pca}}$  varies over a wide range. There is a relatively flat region around  $k_{\text{pca}} \in [16, 40]$ , where accuracy is slightly better and the variance across runs is small, which indicates that the model does not critically depend on a specific choice as long as enough variance is retained by the components. To balance accuracy with preprocessing cost,  $k_{\text{pca}} = 24$  is used in the main experiments. For the number of periods  $n$ , three settings are compared, one daily cycle, one weekly cycle, and the combination of daily and weekly cycles. The plots confirm that using a single period loses useful structure, while combining day and week consistently yields the lowest MAE and RMSE, in line with the multi seasonal behavior observed in datasets.

Second, the hyperparameters for the node interaction priors are studied, namely the embedding dimensions  $k_{\text{topo}}$ ,  $k_{\text{delay}}$ ,  $k_{\text{corr}}$  and the Hann window size in the Welch estimator used for time delayed interactions. Results are in Figure 8(b). The three interaction dimensions control the latent spaces for the topology prior, the time delay prior, and the correlation prior. To keep these subspaces balanced when concatenated, they are tied to a common value and swept jointly. The sensitivity

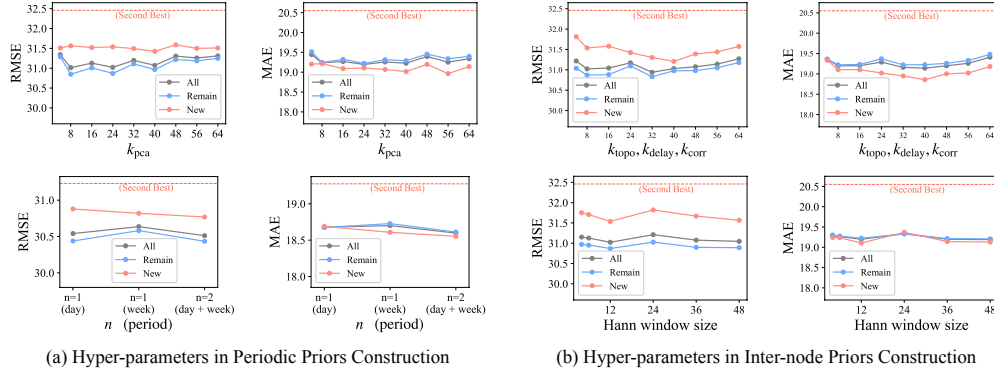


Figure 8: Full results of the hyper-parameter study on PEMS04 dataset.

plots show only mild fluctuations of MAE and RMSE over the full range of embedding sizes. Very small dimensions slightly hurt performance, while very large dimensions provide no visible gain but increase both prior construction cost and refinement cost. A moderate setting with  $k_{\text{topo}} = k_{\text{delay}} = k_{\text{corr}} = 8$  achieves a good trade off and is adopted as the default. For the Hann window size, the curves exhibit the expected bias variance behavior. Very short windows suffer from higher variance and weaker frequency resolution, which slightly degrades accuracy. As the window size increases, performance quickly enters a stable band and then saturates. We set the Hann window size to  $T = 12$ , which matches the forecasting horizon, providing a balanced compromise between variance reduction and time localization.

Finally, robustness with respect to network evolution is evaluated by varying the proportion of newly added nodes and the retirement rate of existing nodes. Figure 9 reports results for different new node ratios and retirement ratios. We observe that MAE and RMSE only change within a narrow range across all tested configurations. This indicates that the periodic prior and the interaction priors act as effective regularizers when the composition of the node set shifts, and that the model can gracefully handle a wide range of expansion profiles.

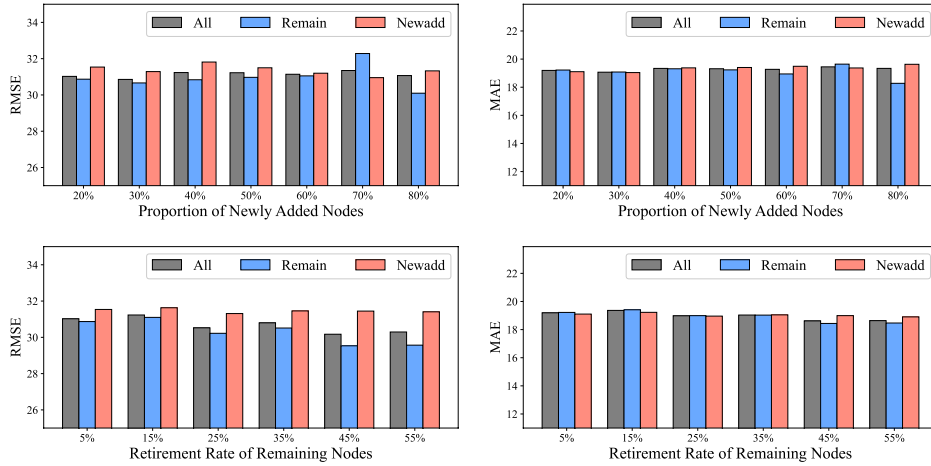


Figure 9: Full results under different node-division-ratio settings on PEMS04 dataset.

#### B.4 FULL RESULTS ON EFFECTIVENESS AND GENERALITY

Table 9 reports the full forecasting results of the effectiveness study. SNIPformer achieves consistently the best accuracy on EPeMS, PEMS04, SeaLoop datasets and has a second-best performance on NREL-AL dataset. Compared with the strongest baseline STEV, the gains of SNIP arise from complementary representation and adaptation. STEV learns node specific parameters from short prediction windows, so each sample covers only a limited temporal context and struggles to encode persistent node heterogeneity. SNIP instead builds node identities as priors from the full historical record using multi period PCA, yielding low dimensional, stable, and discriminative prompts that preserve inter node differences while remaining independent of training window length. On top of these priors, SNIP employs a lightweight refinement module that learns corrections conditioned on the current short window and the diffusion context, which shifts the objective from fitting an all purpose embedding to adjusting residuals under changing conditions. When nodes are long term distinct but become temporarily similar due to events such as congestion, the priors anchor identity and the refinement adapts to the current regime. This combination translates into stronger generalization for new nodes with scarce history and more stable accuracy for remain nodes across horizons, aligning with the observed improvements over STEV.

Table 10 and Table 11 report the full forecasting results of the generality study. Taken together, these results indicate that, across MLP, graph, attention, and hybrid backbones, inserting SNIP consistently improves performance on All, Remain, and New nodes, which supports the interpretation of SNIP as a model agnostic prompting layer rather than a backbone specific trick. In comparison with the attention based prompting strategy (AttP), SNIP leverages explicitly constructed priors and flexible learned fine tuning to produce more informative node discriminative prompts and superior predictive accuracy, while remaining a plug and play component for architectures whose key layers are not hard tied to the cardinality of the node set.

#### B.5 EVALUATION ON MULTIPLE EXPANSION STAGES

In addition to the fixed four stage settings, we further evaluate scalability and adaptability on two standard streaming benchmarks from EAC (Chen & Liang, 2025) with larger and more dynamic node sets. PEMS-Stream contains traffic data from 2011 to 2017, where the number of sensors increases from 655 to 871 over multiple expansion phases, and Air-Stream contains air quality measurements from 2016 to 2019 with more than one thousand sensors. Both benchmarks are designed for continual phase wise evaluation. Following the EAC protocol, we conduct 12 step ahead forecasting and report MAE and RMSE for SNIPformer, the lightweight backbone with SNIP (STID<sup>†</sup>+SNIP), and strong continual baselines EAC and STKEC, as summarized in Table 7 and Table 8. Across all phases on both datasets, SNIPformer and STID<sup>†</sup>+SNIP consistently outperform the continual baselines, which provides additional evidence that the proposed priors and prompting mechanism scale to longer and larger streams and remain effective under multiple successive expansion phases.

Table 7: Forecasting performance on PEMS-Stream dataset.

Model	Metric	2011	2012	2013	2014	2015	2016	2017
STKEC	MAE	15.80	15.77	15.86	16.77	16.27	15.64	17.16
	RMSE	24.63	25.00	25.96	27.60	26.85	27.91	28.17
EAC	MAE	14.51	14.23	14.37	15.20	14.87	14.28	15.91
	RMSE	22.22	22.14	23.13	24.31	24.26	25.65	25.87
SNIPformer	MAE	<b>11.79</b>	<b>11.20</b>	<b>11.20</b>	<u>11.94</u>	<u>11.58</u>	<u>11.07</u>	<u>12.76</u>
	RMSE	<b>18.02</b>	<b>17.73</b>	<u>18.47</u>	<b>19.24</b>	<b>19.37</b>	<u>21.51</u>	<u>21.55</u>
STID <sup>†</sup> +SNIP	MAE	<u>12.30</u>	<u>11.29</u>	<u>11.23</u>	<b>11.59</b>	<b>11.51</b>	<b>10.86</b>	<b>12.62</b>
	RMSE	<u>19.20</u>	<u>18.34</u>	<b>18.18</b>	<u>19.40</u>	<u>19.58</u>	<b>21.23</b>	<b>21.18</b>

Table 8: Forecasting performance on Air-Stream dataset.

Model	Metric	2016	2017	2018	2019
STKEC	MAE	31.04	27.04	<u>20.16</u>	21.46
	RMSE	49.91	41.75	<u>34.08</u>	33.28
EAC	MAE	31.39	25.75	20.71	21.25
	RMSE	49.99	39.34	34.20	32.94
SNIPformer	MAE	<u>25.55</u>	<u>23.38</u>	20.56	<u>19.19</u>
	RMSE	<u>41.31</u>	<u>36.80</u>	35.35	<u>30.50</u>
STID <sup>†</sup> +SNIP	MAE	<b>24.55</b>	<b>21.92</b>	<b>19.00</b>	<b>18.84</b>
	RMSE	<b>40.19</b>	<b>35.17</b>	<b>32.47</b>	<b>30.26</b>

## B.6 EVALUATION ON MULTIPLE HORIZON SETTINGS

To further assess robustness under long range forecasting, we conduct multi horizon experiments on EPeMS using a lightweight backbone, STID<sup>†</sup>, with and without SNIP. Specifically, we consider four multi horizon configurations, where the most recent 12, 24, 48, and 96 time steps are used to predict the subsequent 12, 24, 48, and 96 time steps, respectively, and report MAE and RMSE for All, Remain, and New nodes, as summarized in Figure 10. Across all horizons and node groups, inserting SNIP consistently improves performance, with relative gains that generally increase from short to medium horizons and exhibit only a modest attenuation at 96 step forecasting. These results indicate that SNIP remains effective under multi step forecasting and does not degrade at longer horizons. This behavior is consistent with the design of the priors, where the periodic prior captures stable long cycle statistics to provide node specific identity prompts, while the dynamic refinement adjusts these prompts using short window time delayed interaction features, which helps mitigate potential drift of purely structural priors as the prediction horizon grows.

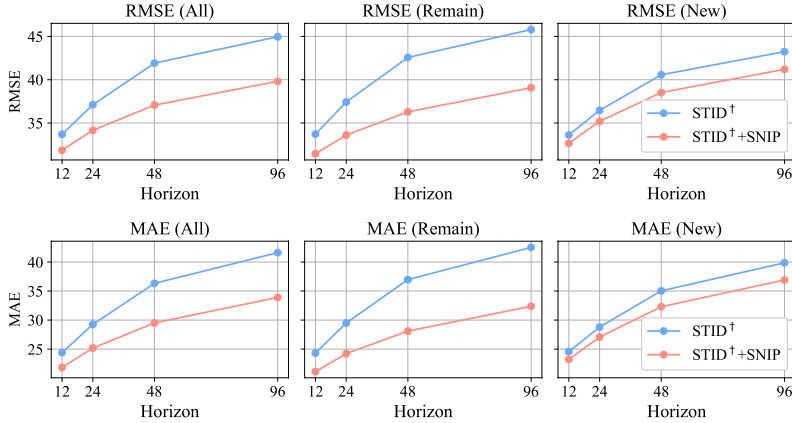


Figure 10: Evaluation on multi-horizon prediction.

## B.7 VISUALIZATION STUDY

Finally, we conduct a qualitative case study that links the geographic layout of node expansion to predictive performance. Figure 11 visualizes three representative scenarios: (a) area expansion on the EPeMS dataset, where new sensors appear in previously uncovered regions of the network, (b) spatial expansion on the EPeMS dataset, where the network is extended along existing corridors, and (c) internal expansion on the SeaLoop dataset, where additional sensors are inserted inside an already monitored region. In each map, existing sensors and newly added sensors are marked separately, which illustrates the different spatial patterns underlying the area, spatial, and internal expansion regimes.

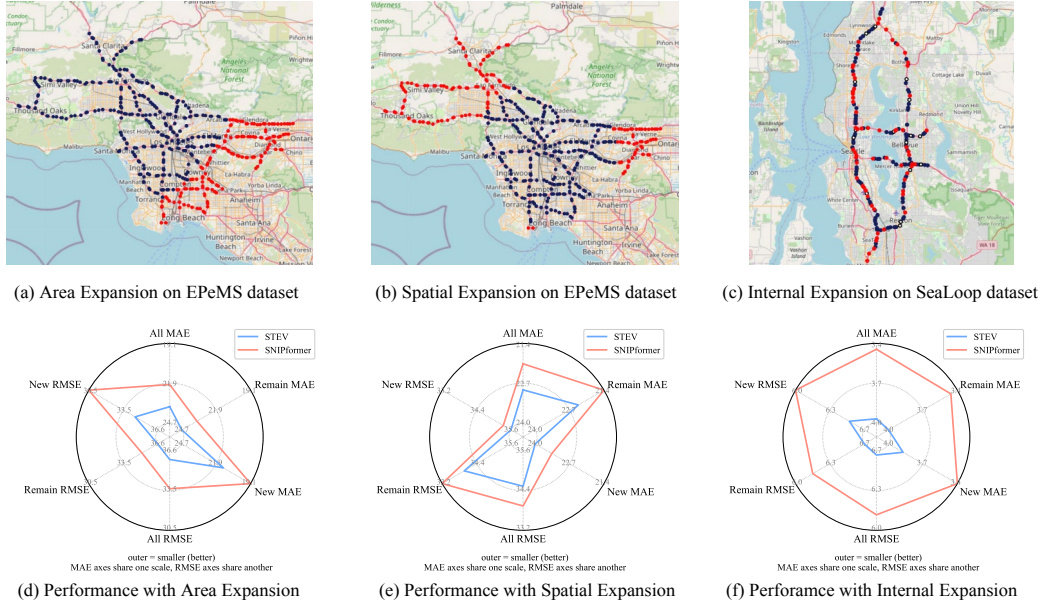


Figure 11: Visualization of geographic distribution and performance comparison with three different expansion scenarios. Red nodes denote the newly added nodes, dark-blue nodes indicate the remain nodes, and white nodes represent the retirements of remain nodes.

For each scenario, panels (d) to (f) report radar plots that summarize MAE and RMSE on All, Remain, and New nodes for the strongest baseline STEV and SNIPformer under the same forecasting setup. MAE related axes share a common scale, RMSE related axes share another, and smaller errors correspond to larger radii on the radar plots. Across the three expansion patterns, SNIPformer typically encloses a larger polygon than STEV, which indicates lower aggregate errors, while the relative gains on Remain and New nodes vary with the specific geographic expansion pattern. In some cases the improvements are more pronounced for newly added sensors in previously uncovered regions, whereas in others they are more balanced between Remain and New nodes. This suggests that the proposed priors and prompting mechanism can adapt to different spatial expansion regimes without introducing systematic degradation on either group of nodes. These gains hold under area, spatial, and internal expansion show that SNIPformer is robust to different geographic expansion modes.

## 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 9: Full comparison of the expanding-node forecasting results of different methods and SNIP-former. 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
DUET	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	<u>9.72</u>	8.98	14.79	14.77	14.88	<u>11.86</u>	<u>12.01</u>	<u>11.37</u>	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	<u>14.67</u>	14.67	<u>14.67</u>	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	<u>87.52</u>	<u>87.75</u>	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	<u>22.90</u>	<u>22.35</u>	<u>23.97</u>	<u>20.55</u>	<u>20.42</u>	<u>21.01</u>	<u>3.92</u>	<u>3.95</u>	<u>3.84</u>	<u>1.57</u>	<u>1.58</u>	<u>1.53</u>
	MAPE	<u>9.45</u>	9.81	<u>8.75</u>	14.77	<u>14.65</u>	15.18	12.56	12.71	12.10	<u>67.52</u>	<u>68.33</u>	<u>65.21</u>
	RMSE	<u>34.51</u>	<u>33.95</u>	<u>35.60</u>	<u>32.46</u>	<u>32.13</u>	<u>33.53</u>	<u>6.62</u>	<u>6.66</u>	<u>6.51</u>	<u>2.88</u>	<u>2.93</u>	<u>2.73</u>
	MRE	<u>0.07</u>	<u>0.07</u>	<u>0.07</u>	<u>0.09</u>	<u>0.09</u>	<u>0.09</u>	<u>0.07</u>	<u>0.07</u>	<u>0.07</u>	<u>0.15</u>	<u>0.14</u>	<u>0.15</u>
SNIPformer (ours)	MAE	<u>22.05</u>	<u>21.39</u>	<u>23.35</u>	<u>19.20</u>	<u>19.22</u>	<u>19.10</u>	<u>3.46</u>	<u>3.47</u>	<u>3.42</u>	<u>1.62</u>	<u>1.65</u>	<u>1.55</u>
	MAPE	<u>8.95</u>	<u>9.20</u>	<u>8.46</u>	<u>12.68</u>	<u>12.64</u>	<u>12.81</u>	<u>10.50</u>	<u>10.62</u>	<u>10.10</u>	88.75	90.38	<u>84.13</u>
	RMSE	<u>33.91</u>	<u>33.16</u>	<u>35.33</u>	<u>31.02</u>	<u>30.87</u>	<u>31.54</u>	<u>6.10</u>	<u>6.14</u>	<u>5.97</u>	<u>2.87</u>	<u>2.92</u>	<u>2.71</u>
	MRE	<u>0.07</u>	<u>0.07</u>	<u>0.07</u>	<u>0.09</u>	<u>0.09</u>	<u>0.09</u>	<u>0.06</u>	<u>0.06</u>	<u>0.06</u>	<u>0.15</u>	<u>0.15</u>	<u>0.15</u>

Table 10: Full forecasting MAE results of different backbones with and without prompting modules. '-' denotes the method is invalid on the expansion scenario.

Family	Model	EPeMS			PEMS04			SeaLoop			NREL-AL		
		All	Remain	New	All	Remain	New	All	Remain	New	All	Remain	New
MLP based	DLinear	32.70	32.26	33.56	28.97	28.91	29.17	4.59	4.62	4.49	2.54	2.59	2.41
	+AttP	32.37	31.91	33.25	28.58	28.52	28.79	4.58	4.61	4.48	2.50	2.55	2.37
	+SNIP	<b>29.13</b>	<b>28.95</b>	<b>29.47</b>	<b>26.45</b>	<b>26.35</b>	<b>26.78</b>	<b>4.52</b>	<b>4.55</b>	<b>4.43</b>	<b>2.19</b>	<b>2.23</b>	<b>2.08</b>
	STID <sup>†</sup>	24.40	24.31	24.56	22.49	22.56	22.25	4.10	4.12	4.03	2.00	2.03	1.89
	+AttP	24.35	24.26	24.53	22.48	22.55	22.23	4.11	4.13	4.04	2.01	2.05	1.90
	+SNIP	<b>21.84</b>	<b>21.13</b>	<b>23.23</b>	<b>19.19</b>	<b>19.21</b>	<b>19.10</b>	<b>3.74</b>	<b>3.74</b>	<b>3.75</b>	<b>1.86</b>	<b>1.89</b>	<b>1.77</b>
Graph based	DCRNN	25.09	24.55	26.14	23.17	23.21	23.04	4.08	4.11	4.01	1.93	1.96	1.83
	+AttP	25.05	24.45	26.23	22.83	22.89	22.64	4.10	4.14	3.99	1.94	1.97	1.84
	+SNIP	<b>23.88</b>	<b>23.07</b>	<b>25.47</b>	<b>19.73</b>	<b>19.71</b>	<b>19.79</b>	<b>3.82</b>	<b>3.84</b>	<b>3.75</b>	<b>1.89</b>	<b>1.93</b>	<b>1.78</b>
	GWN <sup>†</sup>	23.73	23.11	24.93	22.99	23.05	22.79	3.94	3.97	3.84	1.79	1.83	1.69
	+AttP	23.75	23.13	24.96	23.01	23.07	22.80	3.94	3.97	3.84	1.79	1.82	1.69
	+SNIP	<b>23.41</b>	<b>22.79</b>	<b>24.62</b>	<b>19.93</b>	<b>20.11</b>	<b>19.31</b>	<b>3.80</b>	<b>3.83</b>	<b>3.71</b>	<b>1.77</b>	<b>1.81</b>	<b>1.66</b>
	GMAN	26.06	25.75	26.68	21.77	22.07	20.78	4.20	4.24	4.08	2.91	2.96	2.78
	+AttP	31.27	30.92	31.94	21.90	22.10	21.23	4.20	4.24	4.09	2.68	2.74	2.51
	+SNIP	<b>25.71</b>	<b>25.42</b>	<b>26.28</b>	<b>21.25</b>	<b>21.42</b>	<b>20.65</b>	<b>4.18</b>	<b>4.22</b>	<b>4.07</b>	<b>2.53</b>	<b>2.56</b>	<b>2.43</b>
Attention based	iTransformer	26.83	26.65	27.16	24.76	24.74	24.83	4.29	4.31	4.21	1.94	1.97	1.83
	+AttP	26.81	26.64	27.14	24.76	24.74	24.84	4.29	4.31	4.21	1.95	1.99	1.84
	+SNIP	<b>24.67</b>	<b>24.14</b>	<b>25.71</b>	<b>21.55</b>	<b>21.57</b>	<b>21.50</b>	<b>4.02</b>	<b>4.02</b>	<b>4.01</b>	<b>1.84</b>	<b>1.88</b>	<b>1.74</b>
	DUET	25.25	25.17	25.39	23.21	23.21	23.24	4.02	4.04	3.93	1.82	1.85	1.72
	+AttP	25.18	25.10	25.32	23.08	23.08	23.08	4.02	4.04	3.93	1.82	1.86	1.72
	+SNIP	<b>23.16</b>	<b>22.56</b>	<b>24.34</b>	<b>20.05</b>	<b>19.92</b>	<b>20.48</b>	<b>3.60</b>	<b>3.58</b>	<b>3.67</b>	<b>1.74</b>	<b>1.77</b>	<b>1.66</b>
	STAEformer <sup>†</sup>	24.86	24.66	25.27	22.95	23.03	22.67	4.15	4.17	4.09	1.95	1.99	1.84
	+AttP	24.80	24.62	25.15	22.92	23.00	22.64	4.19	4.21	4.12	1.93	1.96	1.81
	+SNIP	<b>23.75</b>	<b>23.05</b>	<b>25.13</b>	<b>21.43</b>	<b>21.55</b>	<b>21.01</b>	<b>3.83</b>	<b>3.86</b>	<b>3.73</b>	<b>1.90</b>	<b>1.94</b>	<b>1.80</b>
Hybrid Architecture	TESTAM <sup>†</sup>	-	-	-	-	-	-	-	-	-	-	-	-
	+AttP	29.51	29.59	29.36	26.75	26.74	26.75	4.04	4.06	3.95	2.42	2.46	2.30
	+SNIP	<b>26.22</b>	<b>25.70</b>	<b>27.26</b>	<b>24.45</b>	<b>24.16</b>	<b>25.42</b>	<b>3.68</b>	<b>3.68</b>	<b>3.71</b>	<b>1.88</b>	<b>1.92</b>	<b>1.78</b>

Table 11: Full forecasting RMSE results of different backbones with and without prompting modules. '-' denotes the method is invalid on the expansion scenario.

Family	Model	EPeMS			PEMS04			SeaLoop			NREL-AL		
		All	Remain	New	All	Remain	New	All	Remain	New	All	Remain	New
MLP based	DLinear	48.32	48.14	48.66	44.55	44.29	45.42	7.99	8.02	7.92	3.91	4.00	3.63
	+AttP	47.49	47.22	48.00	43.61	43.32	44.57	7.94	7.96	7.87	3.88	3.97	3.60
	+SNIP	<b>43.30</b>	<b>43.41</b>	<b>43.08</b>	<b>41.09</b>	<b>40.78</b>	<b>42.12</b>	<b>7.82</b>	<b>7.84</b>	<b>7.74</b>	<b>3.59</b>	<b>3.68</b>	<b>3.34</b>
	STID <sup>†</sup>	37.38	37.44	37.23	35.92	35.77	36.42	7.26	7.28	7.20	3.25	3.33	3.01
	+AttP	37.34	37.39	37.22	35.94	35.79	36.43	7.28	7.29	7.22	3.27	3.36	3.03
	+SNIP	<b>33.73</b>	<b>32.90</b>	<b>35.31</b>	<b>31.03</b>	<b>30.86</b>	<b>31.63</b>	<b>6.76</b>	<b>6.76</b>	<b>6.77</b>	<b>3.07</b>	<b>3.13</b>	<b>2.88</b>
Graph based	DCRNN	37.45	37.04	38.23	36.75	36.60	37.26	7.13	7.15	7.06	3.40	3.48	3.15
	+AttP	37.25	36.76	38.20	36.25	36.14	36.60	7.23	7.27	7.07	3.41	3.49	3.16
	+SNIP	<b>35.70</b>	<b>34.86</b>	<b>37.31</b>	<b>31.53</b>	<b>31.34</b>	<b>32.17</b>	<b>6.78</b>	<b>6.81</b>	<b>6.66</b>	<b>3.31</b>	<b>3.40</b>	<b>3.06</b>
	GWN <sup>†</sup>	35.81	35.27	36.84	36.70	36.57	37.16	6.82	6.86	6.68	3.16	3.24	2.92
	+AttP	35.82	35.26	36.88	36.73	36.59	37.19	6.80	6.83	6.68	3.16	3.24	2.92
	+SNIP	<b>35.54</b>	<b>34.97</b>	<b>36.63</b>	<b>32.13</b>	<b>32.25</b>	<b>31.71</b>	<b>6.69</b>	<b>6.73</b>	<b>6.55</b>	<b>3.13</b>	<b>3.20</b>	<b>2.90</b>
	GMAN	38.87	38.76	39.08	35.21	35.46	34.36	7.64	7.69	7.49	4.27	4.36	4.01
	+AttP	45.46	45.58	45.22	35.58	35.67	35.27	7.64	7.69	7.47	4.09	4.19	3.78
	+SNIP	<b>38.52</b>	<b>38.45</b>	<b>38.65</b>	<b>34.49</b>	<b>34.57</b>	<b>34.20</b>	<b>7.58</b>	<b>7.62</b>	<b>7.41</b>	<b>3.91</b>	<b>3.99</b>	<b>3.67</b>
Attention based	iTransformer	41.40	41.22	41.73	39.62	39.34	40.52	7.54	7.56	7.46	3.36	3.44	3.12
	+AttP	41.38	41.21	41.71	39.62	39.35	40.53	7.54	7.57	7.46	3.36	3.45	3.11
	+SNIP	<b>38.14</b>	<b>37.68</b>	<b>39.02</b>	<b>34.55</b>	<b>34.42</b>	<b>34.97</b>	<b>7.15</b>	<b>7.18</b>	<b>7.05</b>	<b>3.23</b>	<b>3.31</b>	<b>2.99</b>
	DUET	38.05	38.18	37.77	36.54	36.32	37.26	7.02	7.04	6.93	3.10	3.17	2.88
	+AttP	37.97	38.11	37.68	36.35	36.15	37.02	7.03	7.05	6.94	3.09	3.17	2.86
	+SNIP	<b>35.00</b>	<b>34.33</b>	<b>36.28</b>	<b>31.72</b>	<b>31.40</b>	<b>32.79</b>	<b>6.17</b>	<b>6.12</b>	<b>6.33</b>	<b>2.98</b>	<b>3.05</b>	<b>2.77</b>
	STAEformer <sup>†</sup>	38.34	38.26	38.50	36.75	36.63	37.14	7.38	7.39	7.37	3.33	3.42	3.07
	+AttP	38.19	38.15	38.25	36.72	36.60	37.11	7.44	7.45	7.40	3.30	3.39	3.04
	+SNIP	<b>36.55</b>	<b>35.74</b>	<b>38.08</b>	<b>34.67</b>	<b>34.82</b>	<b>34.18</b>	<b>6.99</b>	<b>7.06</b>	<b>6.78</b>	<b>3.21</b>	<b>3.29</b>	<b>2.97</b>
Hybrid Architecture	TESTAM <sup>†</sup>	-	-	-	-	-	-	-	-	-	-	-	-
	+AttP	41.74	41.87	41.48	38.83	38.71	39.22	7.08	7.11	6.99	3.63	3.72	3.38
	+SNIP	<b>38.25</b>	<b>37.54</b>	<b>39.61</b>	<b>35.03</b>	<b>34.56</b>	<b>36.56</b>	<b>6.42</b>	<b>6.41</b>	<b>6.44</b>	<b>3.10</b>	<b>3.17</b>	<b>2.91</b>