Recurrent Prompt Learning for Spatio-Temporal Forecasting

Changlu Chen, Yanbin Liu, Chaoxi Niu, Kaize Shi, Ling Chen, *Senior Member, IEEE*, Tianqing Zhu

***Abstract*—Spatio-temporal forecasting holds great significance for various applications in the intelligent transportation system. Foundation models are revolutionizing spatio-temporal forecasting models due to their one-fits-all generalization capabilities. To reprogram the foundation models for the targeted downstream tasks, prompt learning has emerged as an effective approach by optimizing a small set of learnable input tokens while keeping the backbone intact. However, current prompt learning in the spatio-temporal domain typically suffers from two critical limitations: (1) time-agnostic, which cannot capture the temporal evolution during the prompt learning to deal with temporally dependent or sensitive scenarios. (2) input-agnostic, which remain static upon the end of the training, thus failing to fit different distributions at inference. To address these challenges, we propose a recurrent prompt learning framework named RePro to repurpose foundation models to downstream ST forecasting tasks. For the first challenge, we design a recurrent prompt network that is dynamically conditioned on the time-evolving prompts and recurrently optimized based on the historical context. This design injects time-awareness into the prompt to achieve progressive recalibration of the intermediate representations in the foundation model under varying temporal contexts. For the second challenge, we incorporate input data of the current time step into the update of each recurrent prompt state, leading to the input-conditioned prompt learning. This design effectively encapsulates distributional shifts into the prompt dynamics, improving generalization and robustness. Furthermore, two complementary modules are introduced to facilitate the effective application of the recurrent prompt to the foundation model, i.e., cross-prompt aggregation and layer-conditioned prompt adaptation. Specifically, the first module aims to unify the prompt representation and reduce the redundancy, while the second module distributes the recurrent prompts into diverse layers of the foundation model for hierarchical prompting. Extensive experiments on multiple spatio-temporal forecasting benchmarks demonstrate that RePro consistently outperforms strong state-of-the-art baselines across MAE, RMSE, and MAPE, achieving up to 8.3% reduction in MAE, with ablation studies validating the contribution of each proposed component.**

***Index Terms*—Spatio-Temporal Forecasting, Foundation Models, Prompting, Recurrent Prompt Network.**

I. Introduction

Spatio-temporal forecasting is fundamental to many intelligent transportation system applications, including intelligent traffic prediction and urban mobility management [1]–[12]. Pre-trained foundation models have profoundly reshaped modern deep learning by scaling model capacity and the amount of data, achieving remarkable results in natural language processing (NLP) [13], [14] and computer vision (CV) [15], [16]. By learning general-purpose representations from massive and heterogeneous datasets, these models provide a versatile backbone that can be effectively transferred to a wide range of downstream tasks. Motivated by this success, the spatio-temporal (ST) forecasting community has begun exploring large pretrained models capable of capturing intricate temporal and spatial dependencies across diverse scenarios [17]–[20]. These models serve as powerful backbones that can be further fine-tuned to different downstream ST forecasting tasks.

However, fully fine-tuning foundation models is often prohibitively expensive and prone to overfitting when task-specific data are limited, which substantially constrains their versatility in downstream applications. To address these challenges, prompt learning has recently emerged as a particularly compelling direction [21]–[24]. Specifically, prompt learning introduces a set of lightweight, learnable embeddings in the input space to recontextualize the frozen model's knowledge and thus reprogram the foundation models for specific downstream objectives [25]–[29].

Compared with NLP and CV, the application of prompt learning to spatio-temporal forecasting remains underexplored.

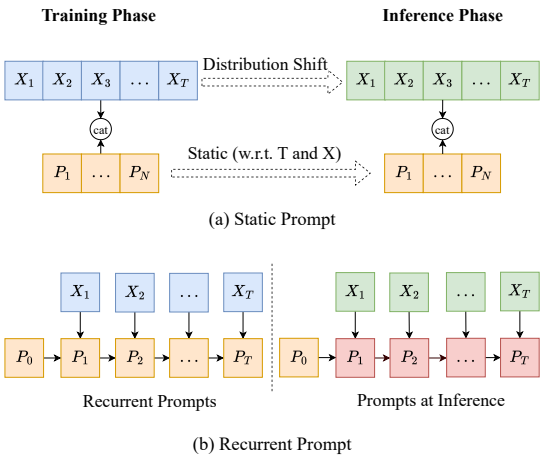


Fig. 1. Statistically designed prompts are time- and input-agnostic, failing to address the temporal dependencies and the potential distribution shifts, while our recurrent prompt can evolve dynamically based on the current input and updating prompt state. Quantitative comparisons are reported in the ablation studies.

Changlu Chen, Chaoxi Niu, and Tianqing Zhu are with the Faculty of Data Science, City University of Macau, Macau, China. (email: clchen@cityu.edu.mo; cxniu@cityu.edu.mo; tqzhu@cityu.edu.mo).

Yanbin Liu is with the Department of Data Science & Artificial Intelligence, Auckland University of Technology, Auckland, New Zealand (email: yanbin.liu@aut.ac.nz).

Kaize Shi is with the School of Science, Engineering and Digital Technologies, University of Southern Queensland, Queensland, Australia. (email: Kaize.Shi@unisq.edu.au).

Ling Chen is with the Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia. (email: Ling.Chen@uts.edu.au).

Corresponding author: Chaoxi Niu.

A representative approach introduces a fixed set of learnable, sensor-specific prompts which are appended to input data along the temporal dimension [30]. These prompts inject sensor-dependent priors and enrich the temporal context, enabling the pretrained model to be more effectively reprogrammed to downstream forecasting tasks. Despite simplicity, such prompts suffer from two fundamental limitations (Figure 1(a)). First, they are time-agnostic, overlooking the inter-temporal dependencies among the evolving temporal sequences. Simply concatenating the learnable prompts along the time dimension limits the model's ability to capture dynamic variations, constraining its capacity to represent evolving patterns. This issue is particularly critical in spatio-temporal forecasting, where data are inherently non-stationary and subject to concept drift across hours, days, or seasons. Without explicitly adapting to these contextual dynamics, these prompts limit the foundation model's adaptability to varying temporal distributions. Second, they are input-agnostic, which means they are learned independently of the current contextual input and remain static at the end of the training. Consequently, these prompts can not effectively instruct foundation models in response to unseen events or underlying distribution shifts during inference or real-world deployment. These limitations underscore the necessity of novel prompting strategies that are both temporally adaptive and input-aware. In this paper, we propose a recurrent prompt (RePro) learning framework, which dynamically evolves prompts conditioned on both temporal context and input signals (Figure 1(b)). In this way, the learned prompts enable the model to capture temporal dependencies while simultaneously adapting to varying input patterns, providing a flexible and expressive interface for spatio-temporal forecasting.

To address the first challenge, a straightforward approach is to design separate prompts for each timestep of input data. However, this naïve approach treats temporal slices independently, neglecting correlations and continuity across timesteps and thus leading to temporally fragmented adaptation. To capture the temporal correlations, we introduce a recurrent prompt network that allows the prompts to dynamically adapt along the temporal sequence to reflect the evolving prompt states. Specifically, rather than optimizing the independently-generated learnable prompt tokens, our prompts are maintained as a latent temporal state that updates regarding the time-evolving context. This recurrent design transforms prompts into a dynamic temporal modulator, carrying forward evolving information across timesteps and guiding the foundation model to capture complex, non-stationary temporal dependencies.

To introduce input-conditioned contextual awareness into prompts, one might attempt to initialize prompts directly from training data. However, such initialization tends to overfit the training distribution and can collapse under distribution shifts or unseen patterns during inference. In this paper, we explicitly incorporate the input signals into the recurrent prompt network to generate input-conditioned prompts. Specifically, the update of each prompt in the recurrent network is influenced by the upcoming observation (current input) while grounded in its evolving internal state (previous prompt), allowing the foundation model to adjust its representations according to varying inputs. This design facilitates the coupling between the prompt and the input features, enabling the foundation model to leverage input-conditioned contextual cues more effectively. Overall, the prompts serve as an adaptive interface that flexibly responds to both long-term temporal trends and short-term variations while embedding distributional shifts into the prompt dynamics from explicit input context, ensuring stable performance across diverse input conditions.

While the recurrent prompt network enhances temporal and input adaptivity, its effectiveness still hinges on how the learned prompts interact with input data and model layers to form a coherent conditioning context. To reinforce prompt integration and cross-layer collaboration, we introduce two complementary designs. First, the cross-prompt aggregation module is proposed to consolidate the diverse prompt signals generated across time and inputs into a unified latent representation. Instead of treating each prompt pattern independently, a learnable latent query attends over the entire prompt set, selectively integrating its complementary cues through attention weighting. This aggregation produces a coherent prompt that delivers consistent guidance to the backbone, reducing redundancy and enhancing efficiency for downstream forecasting. Furthermore, a layer-conditioned prompt adaptation is used to transform the unified prompt into depth-aware variants via lightweight linear projections, allowing lower layers to focus on localized and short-term variations while enabling deeper layers to capture more global and long-range dependencies. These two modules bridge prompt-level fusion and layer-level adaptation, achieving a unified yet hierarchical prompting mechanism that enhances both applicability and generalization of the recurrent prompt framework.

The main contributions are summarized as follows:

- We propose a versatile prompting framework for robust, generalizable spatio-temporal forecasting. By reformulating the problem from static feature modeling to input-conditioned dynamic guidance, we provide a new perspective on how foundation models can be reprogrammed for complex, non-stationary spatio-temporal environments.
- An input-conditioned recurrent prompt learning framework is proposed, where prompts evolve as a latent temporal state across time steps. This design captures varying temporal correlations while embedding distributional shifts into the prompt dynamics from explicit input context, improving generalization and robustness.
- We introduce cross-prompt aggregation and layer-conditioned adaptation to facilitate the effective application of prompts to the foundation model. Diverse prompt signals are unified and transformed into depth-aware variants to enable hierarchical modulation of temporal and input-conditioned cues, enhancing overall model adaptability and efficiency.
- Extensive experiments on multiple traffic forecasting benchmarks demonstrate state-of-the-art performance, and ablation studies validate the effectiveness of each proposed component.

II. Related Work

A. Foundation Models for Urban Forecasting

Foundation models have achieved remarkable success in natural language processing and computer vision [13]–[16] due to their powerful representation capability from extensive pretraining on large-scale data. Although large-scale pretraining data in the spatio-temporal forecasting domain is comparatively limited than in CV and NLP, recent efforts have started to explore ST foundation models capable of capturing universal spatio-temporal patterns in a unified framework. Existing approaches can be generally categorized into two classes based on how the foundation model is obtained, i.e., adapting existing large models [8], [18], [20] for ST tasks and pretraining from scratch with ST data [17], [19], [31].

For the first category, UrbanGPT [18] integrates ST learning with an instruction-tuning framework, leveraging the capabilities of LLMs to comprehend complex spatio-temporal correlations. Similarly, CityGPT [20] proposes a comprehensive framework that incorporates urban-related data into LLMs, including the generation of instruction-tuning datasets, fine-tuning methods for various LLMs, and evaluation benchmarks for urban geospatial tasks. For the second category, ST foundation models are pretrained from scratch specifically for spatio-temporal forecasting to generalize across diverse urban scenarios. For instance, UniST [17] uses extensive, heterogeneous spatio-temporal datasets to pretrain a Transformer architecture and employs knowledge-based prompts to enhance generalization. UrbanDiT [31] develops a foundation model to unify multiple ST forecasting tasks by integrating data-driven prompts and task-specific prompts into a diffusion Transformer backbone, and leveraging multi-city and multi-format data. OpenCity [19] pretrains a Transformer-GCN model on large-scale heterogeneous traffic datasets to capture complex spatio-temporal correlations, enabling generalization to a broad range of traffic forecasting tasks.

Although foundation models provide a powerful backbone for spatio-temporal forecasting, task-specific adaptation strategies are essential to fully exploit their potential in downstream forecasting. In this paper, we focus on exploring the prompt-based adaptation approaches.

B. Prompt Learning for Spatio-temporal Forecasting

Prompt learning, which initially emerged in the NLP and CV domains [21]–[24], customizes foundation models for the downstream tasks by introducing a set of learnable tokens while keeping the backbone intact. Unlike traditional parameter-efficient fine-tuning strategies [32]–[36], which modify the model architecture by inserting additional layers to regulate the intermediate representation, prompting directly operates on the input space to inject task-specific patterns into foundation models for downstream tasks. This paradigm has also been extended to the spatio-temporal forecasting domain to exploit spatial and temporal contexts. Specifically, PromptST [30] proposes a lightweight spatio-temporal prompt strategy, designing tokens for each spatial location and combining them along the temporal dimension to capture attribute-specific contextual information. However, these prompts are static and cannot adapt to varying temporal ranges. FlashST [37] employs a spatio-temporal prompt network and adds regularization on prompt embeddings to mitigate distribution shifts. UniST [17] designs prompts based on spatial and temporal memory pools to enhance the generalization of pretrained ST foundation models. ProST [38] targets dynamic graphs, capturing edge structural information via subgraph prompts to guide node interactions. UrbanDiT [31] integrates both task-specific and data-driven prompts to enable generalization across diverse urban applications.

Despite these advances, existing prompting methods lack the flexibility to represent the correlated and evolving temporal patterns of ST data. Moreover, existing static prompts can not deal with distribution shifts between training and inference. This limitation motivates our recurrent prompt learning framework, which updates prompts dynamically over time based on the input, thereby enabling foundation models to better adapt to spatio-temporal forecasting dynamics.

III. Methodology

A. Problem Formulation

In this paper, we focus on the spatio-temporal forecasting on a traffic network, denoted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is the set of nodes representing spatial grids or sensors, $|\mathcal{V}| = N$ is the number of nodes, $\mathcal{E}$ is the set of edges, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix encoding the spatial dependencies between nodes. Each node $v_i \in \mathcal{V}$ continuously records a sequence of observations such as traffic flow, speed, or demand over time. Specifically, for time step t, the node features are represented as $\mathbf{X}_t \in \mathbb{R}^{N \times d}$, where d denotes the feature dimension.

Given the historical observations from the previous H time steps $\mathbf{X}_{t-H+1:t} = [\mathbf{X}_{t-H+1}, \ldots, \mathbf{X}_t]$, spatio-temporal forecasting aims to predict the future states $\mathbf{Y}_{t+1:t+F} = [\mathbf{Y}_{t+1}, \ldots, \mathbf{Y}_{t+F}]$ for the next F time steps, i.e.,

$$\mathbf{Y}_{t+1:t+F} = f_\Theta(\mathbf{X}_{t-H+1:t}, \mathbf{A}), \tag{1}$$

where $f_\Theta(\cdot)$ is the forecasting model parameterized by Θ. In this paper, the historical window H and prediction horizon F are both set to 288 time steps. This corresponds to one-day historical input and one-day-ahead forecasting, with a data aggregation frequency of 5 minutes.

In this work, f_Θ is instantiated as a frozen ST foundation model equipped with a recurrent prompt network. The recurrent prompt network learns to generate a time-evolving and input-aware prompt set $\mathbf{P_t}$ conditioned on the current input $\mathbf{X_t}$ and the previous prompt state $\mathbf{P_{t-1}}$.

B. Overview of the Framework

The framework of the proposed method is presented in Figure 2. Given the spatio-temporal inputs, we introduce the recurrent prompt learning framework to adaptively modulate the ST foundation model according to evolving temporal contexts. Specifically, we begin by initializing a static node-aware prompt $\mathbf{P}_0$. Then, the recurrent prompt network iteratively updates the prompt state over time: at each time step t, the current prompt $\mathbf{P}_t$ is generated as an input-conditioned

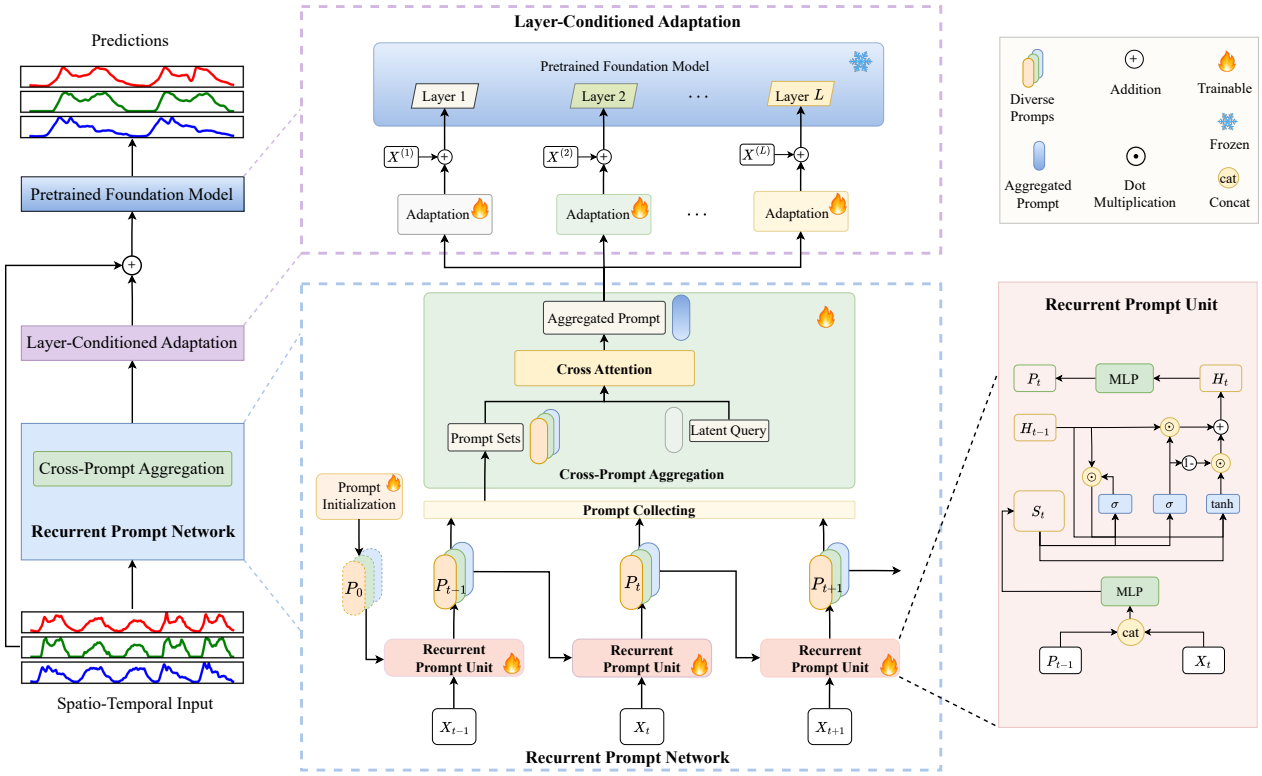


Fig. 2. The overall framework. Our framework equips a spatio-temporal foundation model with a recurrent prompt mechanism. The preprocessed inputs, enriched with spatial and temporal embeddings, are fed into a recurrent prompt network that generates time-evolving, input-conditioned prompts $\{\mathbf{P}_t\}_{t=1}^{T}$, capturing adaptive and input-conditioned temporal context. Then, a cross-prompt aggregation module integrates multiple prompts into a compact one, which interacts with the foundation model via layer-specific adaptations. The layer-specific prompts are added with each layer's input to modulate the hierarchical representations of the foundation model.

evolution of the previous state $\mathbf{P}_{t-1}$ and the current observation $\mathbf{X}_t$, enabling the prompt representations to evolve alongside the temporal progression of the data. Moreover, a cross-prompt aggregation module is further designed to unify diverse prompts into a compact yet expressive representation. This aggregation allows the model to fuse complementary cues from different prompt signals, enhancing representational richness while reducing redundancy. The aggregated prompts are routed through the layer-conditioned adaptations, then fused with the corresponding layer inputs, and finally forwarded to the frozen foundation model.

C. Recurrent Prompt Network

Different from existing static and temporal-agnostic prompting methods, we propose to learn a recurrent prompt network (RPN) to generate temporal-aware and input-conditioned prompts for adapting ST foundation models for downstream tasks.

1) Spatio-temporal Prompt Initialization: In the recurrent prompt learning process, the evolving prompt state is updated across time steps based on its precedence. Thus, it is critical to provide informative spatial priors that anchor the recurrent dynamics before temporal evolution begins. Therefore, we introduce a spatio-temporal prompt initialization that encodes node-wise characteristics before any temporal dynamics are observed. Specifically, we initialize a node-aware prompt state as a learnable tensor:

$$\mathbf{P}_0 \in \mathbb{R}^{P \times N \times D}, \tag{2}$$

where P denotes the number of prompt slots, N is the number of spatial nodes (e.g., sensors or regions). The prompt is broadcast to each sample in the batch and serves as the seed for recurrent updates.

This initialization defines the spatial foundation upon which temporal prompt evolution is built. In other words, $\mathbf{P}_0$ serves as a node-aware prior that anchors the recurrent prompt dynamics before any temporal evolution is observed. The inclusion of the node dimension N allows the model to maintain node-specific priors, while the slot dimension provides multiple latent subspaces that can represent the diverse patterns of input data.

2) Recurrent Prompt Unit: While the initialized spatial prompts provide a structural prior, real-world spatio-temporal systems evolve continually, exhibiting non-stationary patterns, temporal dependencies, and context shifts. A time-agnostic prompt cannot adapt to the inherent inter-dependencies across time steps, which limits its capacity to capture temporal evolution and contextual continuity. To address this issue, we introduce the Recurrent Prompt Unit (RPU) as the core temporal modeling mechanism, which conceptualizes the prompt as a latent temporal state in the recurrent prompt network.

Rather than generating isolated prompts for each timestep, RPU establishes a temporal flow of prompt representations, where each prompt state is updated through a recurrent transition operator. The RPU module is shared across all time steps, and its parameters are consistently reused to evolve the prompt trajectory, ensuring coherent temporal dynamics while enabling the model to accumulate and transform contextual information under the dynamic spatio-temporal environment. Formally, the recurrent evolution of prompts can be expressed as:

$$\mathbf{P}_{1:T} = \Phi_\theta(\mathbf{P}_0)\,, \tag{3}$$

where $\mathbf{P}_0$ is the initialized prompt, Φ_θ is a learnable recurrent transition operator that governs the trajectory of prompt states $\mathbf{P}_t$ over time.

Through this recurrent evolution, the prompt state forms a memory pathway that propagates and refines temporal information, allowing it to adaptively adjust to different sequence lengths and temporal regimes across diverse ST datasets. The RPU therefore transforms the prompt from a static conditioning vector into a dynamic temporal modulation mechanism, which continuously restructures the foundation model's representational space to accommodate non-stationary trends, long-range dependencies, and temporally shifting contextual cues.

Although temporal recurrence enables the prompt to evolve with time, the process remains purely self-driven: the evolution depends solely on the preceding prompt state and lacks explicit awareness of the incoming observations. This limitation prevents the prompt from adapting to instance-specific variations or capturing potential distribution shifts during inference. To overcome this issue, we extend the recurrent transition operator into an input-conditioned state evolution mechanism. It allows each prompt $\mathbf{P}_t$ at time step t to update depending jointly on the current encoded feature $\mathbf{X}_t \in \mathbb{R}^d$, and the previous prompt state $\mathbf{P}_{t-1} \in \mathbb{R}^{d_p}$. To achieve this, we first concatenate the two signals and align their dimensions through a fusion network:

$$\mathbf{S}_t = \mathrm{Fuse}_\phi([\mathbf{P}_{t-1}; \mathbf{X}_t]) \in \mathbb{R}^D, \tag{4}$$

where $\mathrm{Fuse}_\phi(\cdot)$ is a light-weight linear layer that aligns the two modalities into a unified prompt–input representation with D features.

To capture the evolution of the latent prompt state, we instantiate Φ_θ using a gated transition formulation that modulates how current contextual input updates the internal prompt trajectory:

$$\mathbf{Z}_t = \sigma(\mathbf{W}_z\mathbf{S}_t + \mathbf{U}_z\mathbf{H}_{t-1}), \tag{5}$$

$$\mathbf{R}_t = \sigma(\mathbf{W}_r\mathbf{S}_t + \mathbf{U}_r\mathbf{H}_{t-1}), \tag{6}$$

$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{W}_h\mathbf{S}_t + \mathbf{U}_h(\mathbf{R}_t \odot \mathbf{H}_{t-1})), \tag{7}$$

where $\sigma(\cdot)$ denotes the sigmoid activation and $\odot$ represents element-wise multiplication. Here, the gating terms $\mathbf{Z}_t$ and $\mathbf{R}_t$ regulate how the prompt-state memory interacts with the new observation: the update gate $\mathbf{Z}_t$ adaptively determines the proportion of new information admitted into the prompt state, the reset gate $\mathbf{R}_t$ modulates the influence of preceding prompt state during the update, and the candidate state $\tilde{\mathbf{H}}_t$ captures the input-driven refinement of the latent prompt state. $\mathbf{W}_z$, $\mathbf{W}_r$, and $\mathbf{W}_h$ transform the fused feature $\mathbf{S}_t$, while $\mathbf{U}_z$, $\mathbf{U}_r$, and $\mathbf{U}_h$ encode recurrent interactions with the previous hidden state. All these transition parameters are implemented as mappings in $\mathbb{R}^{P\times D}$, ensuring that the gating mechanism naturally integrates both historical prompt dynamics and current input context.

Furthermore, the new hidden state is updated as a convex interpolation between the previous and candidate states:

$$\mathbf{H}_t = (1 - \mathbf{Z}_t) \odot \mathbf{H}_{t-1} + \mathbf{Z}_t \odot \tilde{\mathbf{H}}_t. \tag{8}$$

Finally, we obtain the updated hidden representation by projecting it to a new prompt embedding:

$$\mathbf{P}_t = \mathrm{MLP}_\psi(\mathbf{H}_t), \tag{9}$$

where $\mathrm{MLP}_\psi(\cdot)$ is a lightweight feed-forward projection that refines the prompt signal.

Iterating this process, each time-evolving prompt $\mathbf{P}_t$ produced by the RPU is appended to a prompt queue $\{\mathbf{P}_1, \ldots, \mathbf{P}_T\}$ via the Prompt Collecting module, yielding a set of prompts that jointly encode both temporal continuity and input-conditioned variations. The gating structure allows the model to selectively integrate new information while preserving relevant temporal memory, effectively embedding distributional shifts into the recurrent dynamics. With the input-conditioned design, the RPU transforms static prompts into adaptive interfaces that bridge dynamic temporal input streams and the frozen foundation model.

The gating mechanism in the recurrent prompt unit controls the balance between preserving historical prompt context and incorporating new input-driven information. Under stable temporal regimes, the prompt state evolves smoothly, while the gates facilitate stronger updates during distribution shifts, allowing the prompt trajectory to adapt rapidly to changing patterns.

3) Cross-Prompt Aggregation: Conventional prompt-based architectures typically concatenate prompts along a specific dimension, such as the temporal axis, to condition the model on auxiliary context. However, in our recurrent prompt framework, temporal dynamics are already explicitly modeled through the time-evolving prompt sequence. Directly flattening and concatenating all prompt slots (P) across space (S) and time (T) would disrupt this temporal structure and incur substantial computational overhead for the subsequent foundation model. Instead, we design a principled aggregation strategy that distills the information from multiple prompt slots into a unified latent representation at each time step. This design preserves the temporal granularity established by recurrence while forming a compact, contextually consistent conditioning signal that can interact efficiently with the downstream backbone.

Specifically, we employ a cross-attention mechanism to aggregate P prompt slots into a single latent representation for each time step. Given the set of prompt embeddings $\mathbf{P}_t = \{\mathbf{P}_t^{(1)}, \ldots, \mathbf{P}_t^{(P)}\} \in \mathbb{R}^{P\times D}$, we first introduce a learnable latent query $\mathbf{z} \in \mathbb{R}^{1\times D}$ that serves as the global

summary token. Then, we obtain the keys and values with the slot-wise prompts:

$$\mathbf{q} = \mathbf{W}_q \mathbf{z}, \quad \mathbf{k} = \mathbf{W}_k \mathbf{P}_t, \quad \mathbf{v} = \mathbf{W}_v \mathbf{P}_t, \tag{10}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{D \times D}$ are learnable projection matrices. To get the aggregation weights, a multi-head attention is utilized:

$$\alpha_t = \mathrm{softmax}\left(\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{d_p}}\right). \tag{11}$$

With the obtained aggregation weights, we obtain the final prompt:

$$\hat{\mathbf{P}}_t = \alpha_t \mathbf{v}. \tag{12}$$

The aggregation operation is repeated for all $t \in [1, T]$ to yield the aggregated prompt stream $\hat{\mathbf{P}} = \{\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_T\} \in \mathbb{R}^{T \times D}$. Instead of treating all prompt slots equally, the cross-attention mechanism adaptively weights informative subspaces, functioning as a latent selection layer that emphasizes salient prompting cues while suppressing redundant signals.

D. Layer-Conditioned Adaptation

Transformers in large foundation models naturally exhibit a hierarchical structure, where lower layers capture local spatial correlations and higher layers encode global or long-range dependencies. Therefore, a single shared prompt would impose uniform conditioning across all layers, failing to align with the depth-dependent semantics of the backbone. To hierarchically prompt the frozen foundation model, we introduce a layer-conditioned prompt adaptation mechanism that tailors the conditioning signal to each Transformer layer.

After obtaining the temporally aggregated prompt stream $\hat{\mathbf{P}} = \{\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \ldots, \hat{\mathbf{P}}_T\}$ from the previous stage, we derive layer-adaptive prompts through lightweight projection modules. Specifically, for each Transformer layer $l \in \{1, \ldots, L\}$, a dedicated $\mathrm{MLP}^{(l)}(\cdot)$ transforms the shared temporal prompt into a depth-specific representation:

$$\mathbf{P}^{(l)} = \mathrm{MLP}^{(l)}(\hat{\mathbf{P}}). \tag{13}$$

Each $\mathrm{MLP}^{(l)}$ produces a prompt that reflects the abstraction level of its target layer, i.e., shallow layers focus on fine-grained spatial structures, while deeper layers emphasize high-level temporal semantics. These layer-specific prompts are then integrated into the corresponding Transformer layers of the foundation model to guide feature learning. In this paper, we add each prompt $\mathbf{P}^{(l)}$ to the input token sequence of the corresponding layer l:

$$\mathbf{H}^{(l)} = \mathrm{Transformer}^{(l)}([\mathbf{P}^{(l)} + \mathbf{H}^{(l-1)}]), \tag{14}$$

where $\mathbf{H}^{(l)}$ represents the output of the l-th Transformer layer. This token-level injection enables direct contextual modulation of the foundation model without modifying or retraining its parameters.

E. Training Objective

At the training stage, the model is optimized to forecast future states via a standard regression objective:

$$\mathcal{L}_{\text{pred}} = \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} \left| Y_{n,t} - \hat{Y}_{n,t} \right|, \tag{15}$$

where N and T are the number of spatial nodes and temporal steps, $\hat{\mathbf{Y}}$ is the prediction output of the forecasting head and $\mathbf{Y}$ denotes the ground truth. The learnable prompts and recurrent prompt network are jointly optimized with the prediction head.

During inference, the recurrent prompt network autonomously propagates through time and updates prompt states based on the observed input sequence, thereby generating temporally adaptive prompts in a single forward pass. These evolving prompts allow the model to dynamically respond to temporal shifts or unseen input distributions without retraining.

IV. Experiments

A. Datasets

To ensure comprehensive and fair evaluation, we adopt the same benchmark datasets as used in the latest ST foundation model OpenCity [19], which covers a wide range of real-world urban traffic scenarios across different regions and modalities. These datasets span multiple cities in the United States and China, including New York City, Chicago, Los Angeles, Shenzhen, and Chengdu, and jointly cover traffic flow, demand, and speed prediction tasks. Specifically, the dataset collection includes sensor-based datasets (*e.g.*, CAD-series and PEMS07M) and grid-based datasets (*e.g.*, CHI-NYC, NYC-TAXI, and TrafficSH). The sensor-based datasets represent road networks using node-level traffic sensors, while the grid-based datasets divide urban regions into regular grids to capture aggregated mobility demand or flow intensities. The diversity of datasets ensures the evaluation of both spatially structured and unstructured urban traffic patterns.

Overall, these datasets collectively provide a broad and challenging benchmark for evaluating the generalization capability and robustness of spatio-temporal forecasting models under diverse spatial structures and temporal dynamics.

B. Evaluation Metrics

To evaluate forecasting performance, three widely used metrics are employed, i.e., Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). Specifically, MAE measures the average magnitude of absolute prediction errors, RMSE captures the square root of the mean squared deviation between predictions and ground truths, and MAPE quantifies the relative percentage

TABLE I
COMPARISON WITH STATE-OF-THE-ART METHODS ACROSS DIVERSE DATASETS, EVALUATED BY MAE, RMSE, AND MAPE (LOWER VALUES INDICATE BETTER PERFORMANCE). THE BEST PERFORMANCE IS HIGHLIGHTED IN BOLD. OUR METHOD OUTPERFORMS ALL BASELINES ON NEARLY ALL DATASETS AND METRICS.

Dataset	Metric	RePro	$\text{Opencity}_{\text{mini}}$	TGCN	STGCN	ASTGCN	GWN	STSGCN	MTGNN	AGCRN	MSDR	STWA	PDFormer
CAD8-1	MAE	**19.21**	20.14	27.68	31.26	29.38	29.03	32.38	31.60	34.98	28.99	32.09	29.64
	RMSE	**32.79**	34.25	45.09	49.91	46.84	49.26	53.28	53.01	58.37	46.91	53.26	49.79
	MAPE	**12.60%**	13.17%	20.18%	24.26%	22.51%	23.23%	25.63%	25.82%	26.82%	23.60%	25.62%	20.65%
CAD8-2	MAE	**15.62**	16.24	23.43	24.30	24.24	25.17	24.60	21.85	24.40	26.67	26.57	24.32
	RMSE	**26.46**	27.49	37.03	39.59	38.14	41.47	41.23	34.02	39.21	41.50	44.01	38.95
	MAPE	**9.23%**	9.59%	15.55%	18.61%	18.36%	18.50%	18.54%	14.20%	16.49%	18.96%	20.21%	14.84%
CAD12-2	MAE	**20.53**	21.53	36.53	34.60	35.19	38.05	37.00	36.20	39.91	40.88	41.21	34.23
	RMSE	**34.24**	36.06	60.01	62.47	57.87	69.89	63.19	65.55	74.47	72.87	77.78	59.70
	MAPE	**25.40%**	26.99%	61.60%	52.49%	59.73%	64.77%	58.29%	56.86%	63.36%	69.47%	67.72%	50.92%
CAD3	MAE	**13.11**	14.29	19.56	20.24	23.60	16.94	21.88	17.59	18.72	22.74	21.65	20.28
	RMSE	**22.13**	23.66	30.82	34.34	39.35	28.81	34.52	28.92	31.93	37.15	37.55	36.43
	MAPE	**18.61%**	21.12%	28.15%	25.33%	41.22%	22.98%	30.20%	25.22%	25.45%	32.37%	26.85%	25.19%
CAD5	MAE	**9.16**	9.78	13.07	13.76	12.58	10.69	13.87	11.70	12.88	15.44	14.43	12.89
	RMSE	**16.17**	17.23	21.56	27.49	21.23	19.75	22.32	20.30	23.78	29.52	24.14	21.18
	MAPE	**23.26%**	25.26%	32.65%	32.89%	30.56%	25.98%	32.84%	26.75%	26.29%	32.49%	29.34%	28.82%
PEMS07M	MAE	**4.02**	4.11	4.88	4.44	4.39	4.17	4.56	4.52	4.61	4.81	4.54	4.62
	RMSE	7.68	**7.57**	8.38	8.37	8.21	7.84	8.05	8.06	8.63	8.61	8.57	8.36
	MAPE	**10.87%**	11.11%	13.94%	12.59%	12.58%	11.46%	12.70%	13.11%	12.81%	13.90%	12.91%	13.74%
TrafficSH	MAE	**0.49**	0.53	1.79	1.60	0.69	0.76	1.66	0.81	1.33	1.84	1.42	0.77
	RMSE	**0.76**	0.83	2.65	3.14	1.09	1.39	3.33	1.26	2.45	3.55	2.49	1.23
	MAPE	**7.28%**	7.76%	17.75%	8.04%	8.05%	9.23%	9.33%	8.29%	8.51%	9.91%	9.92%	8.34%
CHI-TAXI	MAE	**1.86**	1.93	4.02	3.09	3.28	3.56	4.87	3.27	3.60	3.55	3.70	4.03
	RMSE	**4.18**	4.29	11.70	9.54	10.32	11.27	14.40	9.87	11.31	10.39	11.49	12.82
	MAPE	**38.82%**	41.26%	60.25%	42.47%	42.82%	41.31%	104.64%	39.38%	46.48%	52.88%	42.52%	44.42%
NYC-TAXI	MAE	**3.10**	3.14	6.10	4.17	5.45	4.28	5.03	3.72	4.55	4.22	6.05	3.64
	RMSE	**6.97**	7.18	12.70	9.19	13.44	10.59	10.96	7.94	10.11	9.08	15.22	8.24
	MAPE	**36.80%**	36.83%	80.39%	45.54%	59.05%	41.82%	65.17%	43.35%	52.45%	55.50%	54.81%	37.40%
p-value	RMSE	-	0.0117	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039	0.0039

deviation, reflecting the model's proportional accuracy. These metrics are formally defined as follows:

$$\text{MAE}(Y, \hat{Y}) = \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} \left| Y_{n,t} - \hat{Y}_{n,t} \right|, \tag{16}$$

$$\text{RMSE}(Y, \hat{Y}) = \sqrt{\frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} \left(Y_{n,t} - \hat{Y}_{n,t} \right)^2}, \tag{17}$$

$$\text{MAPE}(Y, \hat{Y}) = \frac{100}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} \frac{\left| Y_{n,t} - \hat{Y}_{n,t} \right|}{Y_{n,t}}, \tag{18}$$

where $\hat{Y}$ and Y denote the prediction and ground truth values respectively. For all the metrics, a lower value indicates better forecasting performance.

C. Baselines

We compare our approach with eleven state-of-the-art spatio-temporal forecasting methods. To ensure a fair comparison, all baselines follow the same preprocessing, data splitting, and training configurations as in our framework. Moreover, these methods can be generally divided into the following categories according to the backbone they employ.

(1) RNN-Based Methods. AGCRN [1] introduces adaptive graph learning within a recurrent framework, enabling each node to maintain its own spatio-temporal dynamics. MSDR [39] improves long-term dependency modeling in RNNs by preserving multiple historical hidden states at each time step, thereby mitigating the forgetting problem commonly observed in standard recurrent networks.

(2) Attention-Based Methods. ASTGCN [40] employs attention mechanisms across spatial and temporal dimensions to capture dynamic multi-scale dependencies. STWA [41] incorporates time-varying and region-specific attention parameters to improve the representation of temporal fluctuations and spatial heterogeneity. PDFormer [42] separates spatial and temporal attention modules and introduces delay-aware modeling to handle mobility lag and spatial diffusion effects.

(3) Graph Convolutional and Hybrid Models. TGCN [43] integrates graph convolutional networks with gated recurrent units to jointly capture spatial and temporal dependencies. STGCN [40] models both dimensions in a unified convolutional structure with temporal gating and graph convolutions. GWN [4] employs adaptive graph learning combined with dilated temporal convolutions to extract multi-range dependencies. MTGNN [44] introduces multi-scale temporal graph learning with diffusion-based convolution, improving generalization across complex spatial networks. Finally, STSGCN [45] constructs dynamic spatio-temporal subgraphs to jointly learn localized correlations in both space and time.

(4) Foundation-Model-Based Baseline. OpenCity [19] represents a recent foundation model for spatio-temporal forecasting. OpenCity designs a Transformer-Graph architecture and is pretrained on large-scale heterogeneous traffic datasets. In our experiments, we adopt the OpenCity version that tunes only the prediction head as a competitive baseline. This setup

TABLE II
ABLATION STUDIES OF VARIOUS PROMPT MECHANISMS. OPENCITY WITHOUT PROMPT SERVERS (**#1**)AS THE BASELINE TO VERIFY THE EFFECTIVENESS OF DIFFERENT PROMPT DESIGNS. WITH THE RECURRENT PROMPT NETWORK, THE PROPOSED METHOD (**#6**) SHOWS SUBSTANTIAL IMPROVEMENTS.

ID	Prompt Mechnisms				CAD3			CAD5			PEMS07M			CAD8-1			CAD12-2		
	Static	Node-aware	Layer-wise	Temporal-ware	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
#1	✗	✗	✗	✗	14.29	23.66	21.12%	9.78	17.23	25.26%	4.11	7.57	11.11%	20.14	34.25	13.17%	21.53	36.06	26.99%
#2	✓	✗	✗	✗	14.21	23.75	20.26%	9.72	17.20	25.35%	4.14	7.61	11.20%	20.09	34.17	13.13%	21.56	36.15	26.95%
#3	✓	✓	✗	✗	14.18	23.72	19.87%	10.14	17.40	25.53%	4.11	**7.53**	10.99%	19.50	33.10	12.94%	21.05	35.13	25.92%
#4	✓	✓	✗	✗	13.90	23.26	19.69%	9.62	16.93	24.45%	4.13	7.83	11.19%	19.37	33.01	12.74%	20.85	34.78	25.59%
#5	✓	✓	✓	✗	13.58	22.74	19.25%	9.56	16.79	24.14%	4.07	7.74	10.88%	19.23	32.90	12.64%	20.62	34.42	25.44%
#6	✗	✓	✓	✓	**13.11**	**22.13**	**18.61%**	**9.16**	**16.17**	**23.26%**	**4.02**	7.68	**10.87%**	**19.21**	**32.79**	**12.60%**	**20.53**	**34.24**	**25.40%**

allows us to directly evaluate the effectiveness of the proposed recurrent prompt mechanism.

D. Implementation Details

Three versions of the OpenCity are used in our experiments, with varying model sizes in terms of the number of layers and the hidden dimension. Specifically, the $\text{OpenCity}_{\text{mini}}$ (2M parameters) has 3 layers of encoder with the embedding dimension as 128, and $\text{OpenCity}_{\text{base}}$ (5M parameters) is also a 3-layer model with feature dimension as 256, while the largest version $\text{OpenCity}_{\text{plus}}$ (26M) has 6 layers with 512 of the embedding size. Following OpenCity, both the future and historical time steps are set to 288, representing one-day-forward forecasting, with the data aggregation frequency to be 5 minutes. For datasets with an extensive spatial coverage, we follow the same segmentation and preprocessing protocols introduced in OpenCity to reduce memory overhead and maintain consistent spatial scales across experiments. Moreover, the patch length and stride are set to 12 to cover an hour range.

For the recurrent prompt unit, the number of prompts is set to 2 by default, except for 1, 4 and 8 for CAD3, CAD12 and CAD8-1, respectively. For the recurrent network optimization, the learning rate is set to 1e-3 for all datasets. The batch size is set to 16, 8, and 4 for the mini, base, and plus variants, respectively. The hyperparameters are selected based on the validation data. Note that we use $\text{OpenCity}_{\text{mini}}$ for the majority of experiments and OpenCity refers to $\text{OpenCity}_{\text{mini}}$ unless clearly specified otherwise.

Although the proposed recurrent prompt network focuses on adapting temporal dynamics, spatial correlations are explicitly modeled by the spatio-temporal foundation backbone. In particular, the graph convolutional network is integrated with the attention module to capture spatial dependencies among nodes based on the underlying graph structure. These spatial representations are learned by the backbone and remain frozen during prompt-based adaptation. In this way, the recurrent prompt mechanism operates on spatially-aware representations and modulates both spatial and temporal information encoded in the backbone.

E. Overall Performance

Table I reports the overall performance of all the methods across all the spatio-temporal datasets. As shown in the table, we can see that the proposed method, RePro, consistently achieves the best results under all evaluation metrics and datasets (except for a slightly higher RMSE on PEMS07M). More specifically, from the table, we can draw the following key observations. (1) Traditional spatio-temporal architectures (e.g., STGCN, ASTGCN, and GWN) show considerably inferior performance compared with foundation-model-based methods, demonstrating the advantage of large-scale pretraining for capturing general spatio-temporal priors. (2) Compared with the latest ST foundation method, OpenCity, RePro obtains an improvement of 8% in MAE and 12% in RMSE, highlighting the effectiveness of the proposed recurrent prompt learning in capturing temporal evolution through input-conditioned updates. (3) RePro maintains stable superiority across datasets with varying spatial granularity, temporal density, and non-stationarity. The robustness demonstrates that the recurrent prompt mechanism generalizes effectively across different temporal frequencies and urban scales, dynamically aligning the model with the varying distributions. The adaptability is crucial for real-world deployments, where traffic flow or mobility data often exhibit abrupt regime changes, diverse patterns, or seasonal transitions.

We also perform a paired Wilcoxon signed rank test [46] to verify the statistical significance of RePro against the baselines across all datasets in terms of RMSE. The results are shown in the bottom line of Table I. We can see that our method surpasses all baseline approaches with a confidence level greater than 98%. Overall, these results demonstrate that the recurrent prompt network provides a powerful mechanism for generating temporal-evolving and input-conditioned prompts, enhancing the adaptation of the ST foundation model to various datasets and input distributions.

F. Ablation Studies

To demonstrate the effectiveness of each component, we conduct comprehensive ablation studies by progressively enabling different prompt mechanisms. Without loss of generality, five representative datasets are used, and the results across all three metrics are summarized in Table II. We detail each variant as follows.

#1. Baseline ($\mathbf{P} = \emptyset$). The first variant directly employs OpenCity for ST forecasting without any prompt, which serves as the baseline to encapsulate various prompts on top of it and verify the effectiveness of different prompt designs.

#2. Static Prompt ($\mathbf{P} \in \mathbb{R}^{P \times D}$). Based on the OpenCity backbone, we establish a basic static prompt variant following conventional prompt-as-token formulations widely used in prior NLP and CV prompt learning literature [30]. Specifically, a set of learnable prompt tokens with dimensions (P, D) are concatenated with the temporal dimension T of the spatio-temporal input before being fed into the backbone foundation

TABLE III
PERFORMANCE OF DIFFERENT SIZES OF OPENCITY WITH AND WITHOUT THE RECURRENT PROMPT NETWORK.

Model	CAD8-1			CAD8-2			CAD12-2			NYC-TAXI		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
$\text{OpenCity}_{\text{mini}}$	20.14	34.25	13.17%	16.24	27.49	9.59%	21.53	36.06	26.99%	3.14	7.18	36.83%
+ RPN	**19.21**	**32.79**	**12.60%**	**15.62**	**26.46**	**9.23%**	**20.53**	**34.24**	**25.40%**	**3.10**	**6.97**	**36.80%**
$\text{OpenCity}_{\text{base}}$	19.43	33.41	12.65%	15.65	26.56	9.23%	20.69	34.61	25.91%	3.08	6.99	**36.61%**
+ RPN	**18.94**	**32.49**	**12.34%**	**15.28**	**26.03**	**9.01%**	**20.27**	**33.70**	**25.23%**	**3.07**	**6.85**	37.98%
$\text{OpenCity}_{\text{plus}}$	20.85	34.99	13.62%	17.01	28.52	10.01%	22.72	37.63	30.21%	3.32	7.99	38.15%
+ RPN	**20.47**	**34.61**	**13.25%**	**16.55**	**27.86**	**9.74%**	**22.14**	**36.70**	**28.82%**	**3.21**	**7.51**	**36.82%**

model during training. The prompt is agnostic to both the temporal dynamics and the specific inputs. Thus, once trained, it is fixed for inference. As shown in Table II, this variant yields only limited improvement over the baseline on most datasets, suggesting that static prompts lack the adaptability to capture dynamic and evolving patterns.

#3. Node-Aware Static Prompt ($\mathbf{P} \in \mathbb{R}^{P\times N\times D}$). We then improve the above prompt by introducing an additional spatial dimension N corresponding to the number of spatial regions into it. This modification enables each node to have its own learnable prompts, effectively allowing the model to encode region-specific contextual priors. However, the prompt remains temporally and input agnostic. The prompt tokens are still concatenated along the temporal axis. We can see that introducing node-aware prompts yields moderate performance gains across most datasets, indicating that spatially specific prompts facilitate the encoding of localized dynamics and heterogeneous spatial dependencies.

#4. Layer-Specific Static Prompt ($\mathbf{P}^{(l)} \in \mathbb{R}^{P\times N\times D}$). The previous variants only apply the prompt to the model input. To investigate the effect of layer-wise prompting, we further inject the node-aware prompts into each layer of the Transformer within the foundation model. Different MLP adapters are used to project the prompts for different layers, enabling layer-wise modulation of the latent representations. As shown in the table, this design further improves performance by continuously refining the representations through hierarchical conditioning.

#5. Temporally-Aware Static Prompt ($\mathbf{P} \in \mathbb{R}^{P\times T\times N\times D}$). To incorporate temporal cues, we expand the prompt with an additional temporal dimension T, enabling the prompt to align with the full ST structure of the input. Since it is infeasible to concatenate the prompt with the input directly, we add the prompt to the ST input instead. This temporal-aware design captures more dynamic dependencies and improves performance across several datasets. However, the prompt is still time-independent and static. Thus, they cannot capture the temporal correlations among evolving time steps and effectively adapt to distribution shifts during inference.

#6. Recurrent Prompt (Ours). Finally, we introduce the proposed recurrent prompt learning framework, which dynamically updates the prompt across time steps based on the evolving input context. This design naturally encapsulates both the node-aware and layer-specific mechanisms, and further employs a prompt-aggregation module to consolidate diverse prompts. As shown in the table, the proposed recurrent prompt achieves the best overall performance, confirming its effectiveness in modeling temporal evolution conditioned on specific input for general-purpose spatio-temporal forecasting.

TABLE IV
ABLATION STUDY OF DIFFERENT RECURRENT MECHANISMS USED IN THE RECURRENT PROMPT UNIT (RPU). LOWER VALUES INDICATE BETTER PERFORMANCE.

Dataset	Recurrent Mechanism	MAE	RMSE	MAPE (%)
CAD8-1	LRU	19.21	32.75	12.66
	LSTM	19.21	32.78	12.61
	Ours	19.21	32.79	12.60
CAD8-2	LRU	15.57	26.46	9.19
	LSTM	15.73	26.85	9.24
	Ours	15.62	26.46	9.23
CAD12-2	LRU	20.53	34.24	25.40
	LSTM	20.79	34.66	25.92
	Ours	20.53	34.24	25.40
CHI-TAXI	LRU	1.85	4.18	37.92
	LSTM	1.87	4.22	38.18
	Ours	1.86	4.18	38.82

G. Generalization to Different Recurrent Mechanisms

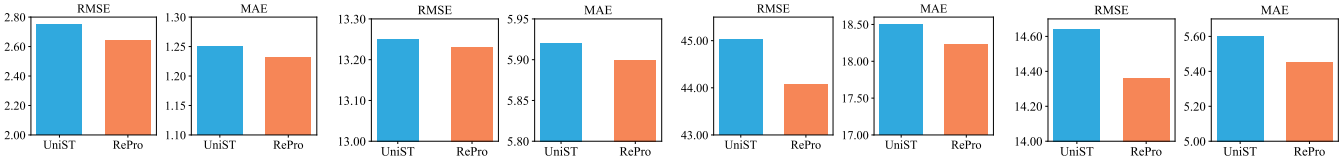
To examine the impact of different recurrent designs in the RPU, we replace the original gated formulation with two alternative recurrent architectures, namely LSTM and LRU [47], while keeping all other components unchanged. Table IV reports the results. We observe that different recurrent mechanisms yield comparable performance across datasets and metrics, suggesting that the effectiveness mainly arises from the input-conditioned recurrent prompt learning framework rather than a specific gating formulation.

H. Performance with Various Sizes of OpenCity

In addition to the results in Table I by using $\text{OpenCity}_{\text{mini}}$ (2M), we also incorporate the proposed framework into $\text{OpenCity}_{\text{base}}$ (5M) and $\text{OpenCity}_{\text{plus}}$ (26M) to demonstrate the adaptability and generalization of the recurrent prompt network. The results are presented in Table III. We can see that the proposed recurrent prompt network consistently improves the forecasting performance over all models across all datasets except for a slightly higher MAPE on NYC-TAXI with $\text{OpenCity}_{\text{base}}$. The performance gains demonstrate the generality of the recurrent prompt mechanism to foundation models of different scales and complexities.

I. Generalization to other ST Foundation Model

To demonstrate the generality of RPN to other ST foundation models, we further apply it to UniST [17], a universal ST model pretrained on diverse spatio-temporal data for various scenarios. For a fair comparison, we incorporate the recurrent



(a) Crowd (b) Cellur (c) TDrive (d) TaxiNYC

Fig. 3. Performance of incorporating the RPN into another ST foundation model, UniST. RePro consistently achieves superior performance to UniST.

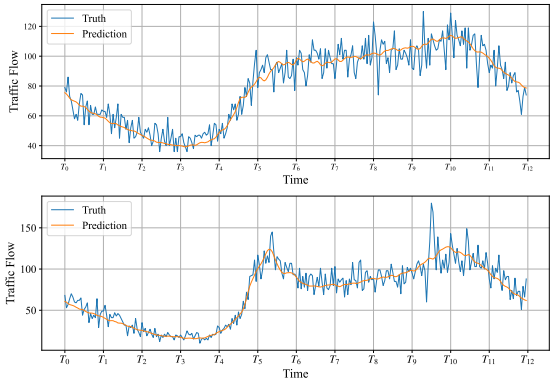


Fig. 4. Traffic flow prediction visualizations on CAD3 dataset.

prompt framework into UniST while keeping other experimental setups fixed. The results on four representative UniST datasets, including two non-traffic spatiotemporal datasets (Crowd and Cellular), are shown in Figure 3. The *Crowd* dataset records crowd flow in Nanjing, China, and the *Cellular* dataset records cellular usage, both with a 30-minute temporal resolution and a 16×20 spatial grid, reflecting regional human activity patterns.

As shown in the figure, the proposed method consistently outperforms UniST across datasets and metrics, further indicating the generalization capability of the proposed method to different foundational backbones.

J. Visualization Results

To qualitatively examine the forecasting performance of our method, we visualize the predicted spatio-temporal sequences together with the ground truth values in Figure 4. As we can see, the proposed method produces predictions that closely follow the real temporal dynamics, capturing both peak and transition patterns across time. This demonstrates that the recurrent prompt enables the foundation model to generate temporally consistent and spatially coherent predictions.

K. Parameter Analysis

In this subsection, we analyze the effect of the number of prompts P on the forecasting performance. Specifically, we vary P in the range of $\{1, 2, 4, 8, 10\}$ while keeping all other hyperparameters and experimental setup fixed. The results are shown in Figure 5. Overall, the model maintains stable performance across a wide range of P values across most datasets. Although increasing P can provide richer contextual representations for temporal adaptation, it also incurs higher computational costs. To balance performance and efficiency, we set the number of prompts to 2 for most datasets.

TABLE V
EFFICIENCY COMPARISON ON CAD3 DATASET.

	Training (min)	Inference (min)	MAE	RMSE	MAPE
Opencity	0.92	0.70	14.29	23.66	21.12%
RePro	2.68	0.82	**13.11**	**22.13**	**18.61%**

L. Efficiency Analysis

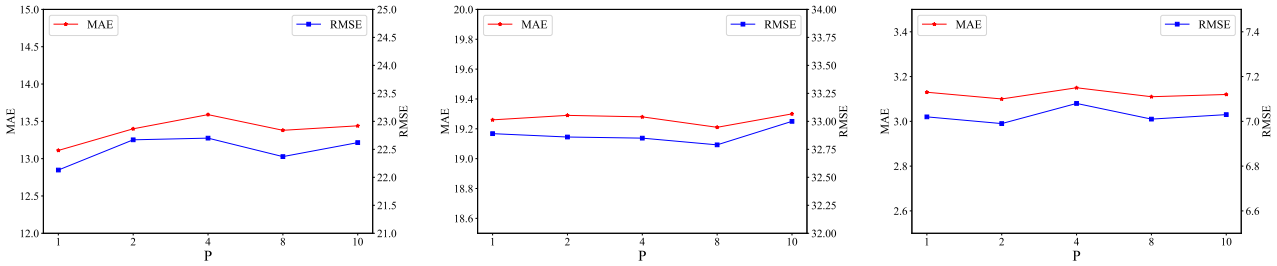
The recurrent prompt updates compact prompt states using a lightweight module, with computation scaling linearly with the temporal horizon and number of spatial nodes. Since it operates only on small prompt representations rather than the full backbone features, the additional overhead is marginal and the backbone computation remains dominant.

We further compare the training and inference time for both the original OpenCity model and our recurrent prompt-equipped method. The results are provided in Table V. Despite the proposed method requiring more training time than OpenCity due to the additional architecture of the recurrent prompt network, it maintains comparable inference time and achieves substantial performance gain, indicating the proposed recurrent prompt maintains a good balance between efficiency and effectiveness for ST forecasting.

V. Conclusion and Future Work

In this paper, we present a recurrent prompt learning framework for adapting foundation models for spatio-temporal forecasting. By reformulating prompts as dynamic latent states that evolve conditioned on input observations and historical context, the proposed method effectively captures time-varying dependencies and mitigates distribution shifts between training and inference. Furthermore, the proposed cross-prompt aggregation and layer-conditioned prompting mechanisms enable hierarchical and depth-aware modulation of spatio-temporal features, further enhancing model adaptability and prompt coherence across layers. Experimental results on multiple urban traffic benchmarks demonstrate that the proposed method outperforms existing methods, providing robust, generalizable forecasting in complex, non-stationary environments. Overall, our work highlights recurrent prompting as a versatile tool to adapt foundation models to various downstream spatio-temporal applications, offering a new paradigm for robust and generalizable forecasting.

While this work primarily focuses on spatio-temporal foundation models for traffic forecasting, the core idea of input-conditioned recurrent prompt evolution is not domain-specific.



(a) CAD3 (b) CAD8-1 (c) NYC-TAXI

Fig. 5. Parameter sensitivity analysis regarding to the number of prompts (P) on different datasets.

Applying the framework to other domains such as NLP, CV, or standard time series forecasting would require appropriate backbone architectures and prompt injection strategies, which we leave for future research.

REFERENCES

[1] L. BAI, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[2] C. Chen, Y. Liu, L. Chen, and C. Zhang, "Test-time training for spatial-temporal forecasting," in *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM, 2024, pp. 463–471.

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

[4] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *The 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[5] T. Luo, Z. Fang, K. Duan, L. Chen, P. Feng, and M. Lu, "Towards online spatio-temporal prediction: A knowledge distillation driven continual learning approach," in *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE, 2025, pp. 2642–2655.

[6] B. An, X. Zhou, Z. Zhou, R. Ragodos, Z. Xu, and J. Luo, "Geopro-net: Learning interpretable spatiotemporal prediction models through statistically-guided geo-prototyping," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 11, 2025, pp. 11 427–11 435.

[7] C. Chen, Y. Liu, L. Chen, and C. Zhang, "Multivariate traffic demand prediction via 2d spectral learning and global spatial optimization," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2024, pp. 72–88.

[8] C. Chen, Y. Liu, C. Niu, L. Chen, and T. Zhu, "Reprogramming vision foundation models for spatio-temporal forecasting," *arXiv preprint arXiv:2507.11558*, 2025.

[9] Y. Zhao, Z. Zhong, A. Wang, H. Wen, M. Jin, Y. Liang, H. Wan, and H. Wu, "Fast: Efficient and effective long-horizon forecasting for large-scale spatial-temporal graphs via mixture-of-experts," *arXiv preprint arXiv:2601.05174*, 2026.

[10] L. Chen, "Pgsformer: traffic flow prediction based on joint optimization of progressive graph convolutional networks with subseries transformer," *Scientific Reports*, 2026.

[11] C. Zhou, Z. Zhang, C. Zhang, H. Miao, Y. Zhang, K. Lyu, and J. Hu, "Feddis: A causal disentanglement framework for federated traffic prediction," *arXiv preprint arXiv:2601.22578*, 2026.

[12] J. Chen, H. Miao, C. Liu, Y. Zhao, and K. Zheng, "Visionst: Coordinating cross-modal traffic prediction with interactive geo-image encoding," in *Proceedings of the ACM Web Conference (WWW)*, 2026.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in NeurIPS*, vol. 33, pp. 1877–1901, 2020.

[14] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[15] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the international conference on computer vision*, 2021, pp. 9650–9660.

[16] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[17] Y. Yuan, J. Ding, J. Feng, D. Jin, and Y. Li, "Unist: A prompt-empowered universal model for urban spatio-temporal prediction," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4095–4106.

[18] Z. Li, L. Xia, J. Tang, Y. Xu, L. Shi, L. Xia, D. Yin, and C. Huang, "Urbangpt: Spatio-temporal large language models," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5351–5362.

[19] Z. Li, L. Xia, L. Shi, Y. Xu, D. Yin, and C. Huang, "Opencity: Open spatio-temporal foundation models for traffic prediction," *arXiv preprint arXiv:2408.10269*, 2024.

[20] J. Feng, T. Liu, Y. Du, S. Guo, Y. Lin, and Y. Li, "Citygpt: Empowering urban spatial cognition of large language models," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 591–602.

[21] C. Han, Q. Wang, Y. Cui, Z. Cao, W. Wang, S. Qi, and D. Liu, "Eˆ 2vpt: An effective and efficient approach for visual prompt tuning," *arXiv preprint arXiv:2307.13770*, 2023.

[22] T. Wang, Y. Liu, J. C. Liang, Y. Cui, Y. Mao, S. Nie, J. Liu, F. Feng, Z. Xu, C. Han *et al.*, "M2 pt: Multimodal prompt tuning for zero-shot instruction learning," *arXiv preprint arXiv:2409.15657*, 2024.

[23] R. Zeng, C. Han, Q. Wang, C. Wu, T. Geng, L. Huangg, Y. N. Wu, and D. Liu, "Visual fourier prompt tuning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 5552–5585, 2024.

[24] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[25] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," *arXiv preprint arXiv:2309.16588*, 2023.

[26] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.

[27] A. Xiao, W. Xuan, H. Qi, Y. Xing, R. Ren, X. Zhang, L. Shao, and S. Lu, "Cat-sam: Conditional tuning for few-shot adaptation of segment anything model," in *European Conference on Computer Vision*. Springer, 2024, pp. 189–206.

[28] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan, "Maple: Multi-modal prompt learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 19 113–19 122.

[29] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Conditional prompt learning for vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 816–16 825.

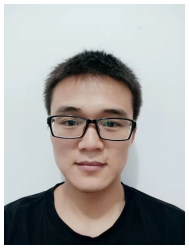
[30] Z. Zhang, X. Zhao, Q. Liu, C. Zhang, Q. Ma, W. Wang, H. Zhao, Y. Wang, and Z. Liu, "Promptst: Prompt-enhanced spatio-temporal multi-attribute prediction," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3195–3205.

[31] Y. Yuan, C. Han, J. Ding, D. Jin, and Y. Li, "Urbandit: A foundation

model for open-world urban spatio-temporal learning," *arXiv preprint arXiv:2411.12164*, 2024.
[32] P. Lan, E. Yang, Y. Liu, G. Guo, J. Zhao, and X. Wang, "Ept: Efficient prompt tuning by multi-space projection and prompt fusion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 23, 2025, pp. 24 366–24 374.
[33] R. Zeng, G. Sun, Q. Wang, T. Geng, S. Dianat, X. Han, R. Rao, X. Zhang, C. Han, L. Huang *et al.*, "Mept: Mixture of expert prompt tuning as a manifold mapper," *arXiv preprint arXiv:2509.00996*, 2025.
[34] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.
[35] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.
[36] Z. Hu, L. Wang, Y. Lan, W. Xu, E.-P. Lim, L. Bing, X. Xu, S. Poria, and R. Lee, "Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models," in *Proceedings of the 2023 conference on empirical methods in natural language processing*, 2023, pp. 5254–5276.
[37] Z. Li, L. Xia, Y. Xu, and C. Huang, "Flashst: A simple and universal prompt-tuning framework for traffic prediction," *arXiv preprint arXiv:2405.17898*, 2024.
[38] K. Xia, L. Lin, S. Wang, Q. Zhang, S. Wang, and T. He, "Prost: Prompt future snapshot on dynamic graphs for spatio-temporal prediction," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 2025, pp. 1645–1656.
[39] D. Liu, J. Wang, S. Shang, and P. Han, "Msdr: Multi-step dependency relation networks for spatial temporal forecasting," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 1042–1050.
[40] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.
[41] R.-G. Cirstea, B. Yang, C. Guo, T. Kieu, and S. Pan, "Towards spatio-temporal aware traffic time series forecasting," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 2900–2913.
[42] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proceedings of the AAAI*, vol. 37, no. 4, 2023, pp. 4365–4373.
[43] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.
[44] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.
[45] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 914–921.
[46] R. F. Woolson, "Wilcoxon signed-rank test," *Wiley encyclopedia of clinical trials*, pp. 1–3, 2007.
[47] V. Patraucean, X. O. He, J. Heyward, C. Zhang, M. S. Sajjadi, G.-C. Muraru, A. Zholus, M. Karami, R. Goroshin, Y. Chen *et al.*, "Trecvit: A recurrent video transformer," *Transactions on Machine Learning Research*, 2024.

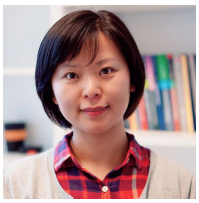
Changlu Chen received the B.S. and M.S. degrees in Electronic Information Science and Technology from Lanzhou University, China. She completed her doctor's degree at the University of Technology Sydney in 2024. Her current research interests include spatial-temporal learning and traffic forecasting.

Yanbin Liu received the B.E. degree and M.S. degrees from Tianjin University, China, in 2013 and 2015, respectively. He received the Ph.D. degree from the University of Technology Sydney, Australia, in 2021. He is currently a lecturer at the Auckland University of Technology. His research interests lie in machine learning and deep learning for computer vision problems, especially learning with limited labelled data.



Chaoxi Niu received the B.S. and M.S. degrees in electronic information science and technology from Lanzhou University, Lanzhou, China. He completed the Ph.D. degree at the University of Technology Sydney, Sydney, NSW, Australia. His research interests primarily focus on deep learning on graphs and spatio-temporal data.

Kaize Shi received the Ph.D. degrees in computer science and computer systems from Beijing Institute of Technology, China, in 2022, and the University of Technology Sydney, Australia, in 2023. He is currently a Lecturer with the School of Mathematics, Physics and Computing, University of Southern Queensland, Australia. His research interests include computational social intelligence, natural language generation, and urban computing. Dr. Shi is an Associate Editor of IEEE Transactions on Computational Social Systems and a Guest Editor for journals such as Information Fusion. He served as a Program Committee Member for conferences such as NeurIPS, ICLR, ICML, ACL, EMNLP, COLING, SIGKDD, etc.



Ling Chen (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore, and undertook postdoctoral training at Leibniz University Hannover (L3S Research Centre), Germany. She is a Professor in the School of Computer Science at the University of Technology of Sydney, Sydney, Australia. She leads the Data Science and Knowledge Discovery Laboratory (The DSKD Lab) within the Australian Artificial Intelligence Institute (AAII) at UTS. Her papers appear in major conferences and journals, including SIGKDD, AAAI, ICLR, ICDM, NeurIPS, and TNNLS. Her research interests mainly include discovering regularities and irregularities from various types of data, data representation learning, social media and social network mining, and dialogue and interactive systems.

Tianqing Zhu received her BEng and MEng degrees from Wuhan University, China, in 2000 and 2004, respectively, and a PhD degree from Deakin University in Computer Science, Australia, in 2014. Dr. Tianqing Zhu is currently a professor in the faculty of data science at the City University of Macau. Before that, she was a lecturer at the School of Information Technology, Deakin University, Australia, and an associate professor at the University of Technology Sydney, Australia. Her research interests include privacy-preserving and AI security.