Towards Dynamic Spatial-Temporal Graph Learning: A Decoupled Perspective

Binwu Wang [1,2], Pengkun Wang [1,2,*] Yundong Zhang [1,2], Xu Wang [1,2], Zhengyang Zhou [1,2], Lei Bai [3], Wang Yang [1,2,*]

[1]University of Science and Technology of China
[2]Suzhou Institute of Advanced Research, University of Science and Technology of China
[3]Shanghai AI Laboratory
{wbw1995,zyd2020,wx309}@mail.ustc.edu.cn, zzy0929@ustc.edu.cn, baisanshi@gmail.com,
{pengkun,angyan}@ustc.edu.cn

Abstract

With the progress of urban transportation systems, a significant amount of high-quality traffic data is continuously collected through streaming manners, which has propelled the prosperity of the field of spatial-temporal graph prediction. In this paper, rather than solely focusing on designing powerful models for static graphs, we shift our focus to spatial-temporal graph prediction in the dynamic scenario, which involves a continuously expanding and evolving underlying graph. To address inherent challenges, a decoupled learning framework (DLF) is proposed, which consists of a spatial-temporal graph learning network (DSTG) with a specialized decoupling training strategy. Incorporating inductive biases of time-series structures, DSTG can interpret time dependencies into latent trend and seasonal terms. To enable prompt adaptation to the evolving distribution of the dynamic graph, our decoupling training strategy is devised to iteratively update these two types of patterns. Specifically, for learning seasonal patterns, we conduct a thorough training of the model using a long time series (e.g., three months of data). To enhance the learning ability of the model, we also introduce the masked auto-encoding mechanism. During this period, we frequently update trend patterns to expand new information from dynamic graphs. Considering both effectiveness and efficiency, we develop a subnet sampling strategy to select a few representative nodes for fine-tuning the weights of the model. These sampled nodes cover unseen patterns and previously learned patterns. Experiments on dynamic spatial-temporal graph datasets further demonstrate the competitive performance, superior efficiency, and strong scalability of the proposed framework.

Introduction

Spatial-temporal graph prediction has emerged as an essential task in the intelligent transportation systems (Yin, Zhang, and Jing 2023; Wang et al. 2023a; Varga et al. 2023; Jiang et al. 2023; Wang et al. 2023b), with the potential to have a significant impact on our daily routines. Recently, researchers have been devoted to developing deep-learning models due to their remarkable capacity to capture complex relationships. The prevailing approaches (Jin et al. 2023; Lan et al. 2022; Zhang et al. 2022) entail conceptualizing spatial-temporal data as spatial-temporal graphs, where monitoring sensing systems are represented through graph structures. Subsequently, spatial-temporal graph convolution networks (STGs) are as an engine to model the spatial-temporal correlations, typically including two modules: graph convolutional networks (GCNs) for spatial correlation and sequence modules for temporal correlation, such as Long Short-Term Memory (LSTM) (Tian et al. 2018; Chahal et al. 2023) or Transformer (Gu et al. 2023; Yan, Ma, and Pu 2021).

Despite their promising results, most of these models are evaluated using short-term datasets (e.g., two months) and portray the underlying graph as static and unchanging. However, when considering a longer time frame, the distribution of the graph can undergo substantial evolution over time, which is termed as **dynamic spatial-temporal graph**. This concept encompasses two crucial elements. Firstly, the underlying structure of the graph would change over time. For example, new nodes may emerge due to urban development and traffic network expansion. These new nodes would introduce new patterns and updated neighbor information. Secondly, the distribution feature of original nodes in the graph would also evolve over time. These evolved patterns should be incorporated into models to update outdated knowledge. Otherwise, the prediction performance of the model for these nodes would be unpromising. Overall, to accurately capture the evolving nature of a dynamic graph, it is crucial to incorporate these unseen patterns and new neighbor information.

The initial approach is the retraining method, however, once the graph changes, retraining a new model can be computationally burdensome. To mitigate this issue, researchers have shifted their focus towards developing continuous learning strategies (Chen, Wang, and Xie 2021; Wang et al. 2023a) that aim to reduce computational complexity, but there may be a trade-off in the prediction performance. However, both the retraining method and continuous methods assume that there is sufficient data available from the updated graph for at least one month. In fact, this assumption is very ideal and oversimplifies the dynamic graph scenario, especially for new nodes. Consequently, the emergence of few-shot (or even zero-shot) challenges would have

*Yang Wang and Pengkun Wang are corresponding authors.

a catastrophic impact, resulting in unreliable predictions (Li, Tang, and Ma 2022). Such inefficient scalability of these methods under the dynamic graph setting would raise concerns for traffic managers who are actively involved in devising new traffic planning strategies. In summary, developing a method with excellent performance, scalability, and efficiency for dynamic spatial-temporal graph learning is still an open question.

One novel insight in the time series community is the structured inductive bias of time series, which refers to interpreting complex temporal correlations as seasonal and trend patterns. The former refers to recurring patterns derived from long-term time series, and these patterns tend to be stable. Adjacent nodes within the graph may exhibit similar seasonal patterns. In contrast, trend patterns exhibit strong correlations with short-term time steps. In light of the characteristics of these two patterns, we propose a dynamic spatial-temporal graph learning method by updating two patterns alternately with different frequencies. The inert update of seasonal patterns enhances the scalability capabilities of the model by providing historical information from the entire graph. On the other hand, by constantly learning trend patterns from short-term data, the model can promptly respond to the evolution of the dynamic graph.

Specifically, we propose a decoupled learning framework (DLF), which can be summarized two-fold. First, we design a disentangled spatial-temporal graph convolutional network (DSTG) to effectively separate temporal associations into seasonal patterns and trend patterns. This allows the model to focus on independently capturing the long-term seasonal variations and the short-term trends. Second, we design a decoupled training strategy for DSTG to achieve efficient and effective dynamic spatial-temporal graph learning, which involves alternately updating two distinct patterns using unequal-length data. Specifically, given that seasonal patterns are believed stable, we only conduct comprehensive training of the model every three months to prevent the assimilation of incorrect long-term patterns from short time series. Furthermore, we implement an additional masked autoencoding strategy to boost the model's learning ability. During this period, we sequentially fine-tune the trend module to promptly learn new knowledge from the dynamic graph (e.g., every half month), the core idea is to extend unseen patterns into the model while consolidating learned ones. Considering performance and efficiency, we develop a subset sampling strategy to sample a representative subset, which can encompass new patterns from the updated road network and previously learned patterns. During the fine-tuning process, we freeze the weights of the seasonal patterns to avoid capturing spurious seasonal patterns from short-period time series. Our contributions are three-fold:

- We first reframe the spatial-temporal prediction problem in a data streaming context and address it with dynamic spatial-temporal graph learning. Our method involves a decoupled learning framework, which consists of two key components: a disentangled spatial-temporal graph network (DSTG) and a decoupled training strategy.
- DSTG decouples temporal correlation into seasonal and trend patterns, then the decoupled training strategy involves alternately updating two patterns to achieve efficient and effective dynamic graph learning, leveraging their separate characteristics and dynamics.
- We evaluate our framework on real-world datasets, and experimental results demonstrate the superiority of our framework in prediction performance, training efficiency, and scalability for new knowledge.

Related Work

Spatial-temporal graph prediction Recently, the accuracy of spatial-temporal graph prediction has significantly improved with the emergence of deep learning (Zhou et al. 2022, 2023b). Among these advancements, the most advanced approach is the spatial-temporal Graph Neural Convolutional Networks (STG) (Wang et al. 2023c; Bai et al. 2020; Wang et al. 2023d,a; Jiang et al. 2023; Zhou et al. 2023a; Xia et al. 2023). For example, ST-GDN (Zhang et al. 2021) and D2STGNN (Shao et al. 2022) employ diffusion graph convolutional networks with RNNs to capture temporal patterns. ST-LSTM (Bi et al. 2022) employs TCN for efficient capture of time dependencies. Additionally, researchers have begun integrating the Transformer architecture into spatial-temporal graph prediction models (Yan, Ma, and Pu 2021; Ren, Li, and Liu 2023), which is renowned for its proficiency in modeling long-time series. However, existing models are primarily designed for static graphs and neglect the dynamic evolution of distribution and underlying structure in the streaming scenario.

Dynamic spatial-temporal graph learning We are witnessing a growing interest in dynamic spatial-temporal graph learning with the availability of large-scale spatial-temporal datasets (Liu et al. 2023; Wang et al. 2020). In this field, developing continuous learning methods is an emerging step. Compared to the retraining method with high performance, their efficiency is impressive, whose primary objective is to consolidate the learned knowledge of the model. For example, TrafficStream (Chen, Wang, and Xie 2021) and ER-GNN (Zhou and Cao 2021) design different experience-replay strategies, and STKEC (Wang et al. 2023a) uses the memory mechanism to explicitly preserve important knowledge. Nevertheless, these methods are limited in their ability to handle dynamic graphs since they necessitate a specific quantity of data from the updated graph.

Problem Formulation

In this section, we first define the spatial-temporal graph prediction problem in a dynamic context, where data is collected in the streaming manner and the graph evolves dynamically over time. To aid our analysis, we assume that the graph structure remains relatively stable over a one-month period (30 days).

We use $\mathbb{G} = (G_1, G_2, \cdots, G_{\mathcal{T}})$ to denote a dynamic spatial-temporal graph, where $G_\tau = \{\mathcal{V}_\tau, \mathcal{E}_\tau, A_\tau\}$ represents the graph during the τ-th month, $\mathcal{E}_\tau$ is a set of edges and $\mathcal{V}_\tau$ is a set of nodes with $|\mathcal{V}_\tau| = N_\tau$. $A_\tau \in \mathbb{R}^{N_\tau \times N_\tau}$ is the adjacency matrix. $\mathbf{X}_\tau = \left[X_\tau^t \in \mathbb{R}^{N_\tau \times F} | t = 0, \cdots, T_h\right]$

is the node feature matrix with feature dimension F of N_τ nodes in the past T_h time steps.

Problem (Dynamic spatial-temporal graph learning). Given a dynamic spatial-temporal graph $\mathbb{G} = (G_1, G_2, \cdots, G_\tau, \cdots)$, our goal is to sequentially learn a model function $\mathcal{F}$ to predict future graph signals. For example, we use the graph information in τ-th month G_τ and archived data over a period of time $\mathcal{D}_\tau$ to train a prediction funciton $\mathcal{F}_\tau$ with the parameters ω_τ. Given a test sample $\mathbf{x}_t$, this function can predict future graph signals of all nodes in the next T_P time-steps, which is denoted as $\hat{y}$:

$$\Pr\left(\hat{y} \mid \mathbf{x}_\tau^t, \mathcal{D}_\tau\right) = \int \Pr\left(\hat{y} \mid \mathbf{x}_\tau^t, \omega_\tau\right) \Pr\left(\omega_\tau \mid \mathcal{D}_\tau\right) d\omega_\tau \tag{1}$$

Where $\Pr(\hat{y} \mid \cdot)$ represents the distribution of the predicted values $\hat{y}$. A optimal model $\mathcal{F}_\tau$ should make the prediction distribution approximate to the ground-truth distribution. To obtain $\mathcal{F}_\tau$, we first identify two training strategies in this paper: the retraining method and the online method.

The retraining method. In the spatial-temporal graph learning community, the retraining method requires collected data from G_τ for training (at least 20-day data in the general setting). Otherwise, they suffer from few-shot data challenge, if there is not enough data, the prediction performance of these models is unpromising (Li, Tang, and Ma 2022). In this paper, we utilize all the graph data that is available until the τ-th month as training data. This means that training data can encompass a time period of $(\tau - 1)$ months.

$$\mathcal{F}_\tau^{\text{re}} : \omega_\tau = \text{argmax}_{\omega_\tau} P\left(\omega_\tau \mid \mathcal{T}\left(G_{1:\tau}\right)\right) \tag{2}$$

where $\mathcal{T}(\cdot)$ means corresponding training data. The learned model with optimal parameters ω_τ is used to directly predict spatial-temporal graph signals of G_τ in the τ-th month without any further fine-tuning.

The online learning method. Our online learning framework only uses a representative subset of the graph G_τ to fine-tune the saved model, which can allow us to integrate new knowledge efficiently. In this paper, we fine-tune the model every half month, thus, there would be two fine-tuning processes every month. For example, if we have collected the first half of the month's data in τ-th month to fine-tune the saved model with parameters $\omega_{\tau-1}$:

$$\mathcal{F}_\tau^{\text{on}} : \omega_\tau = \text{argmax}_{\omega_\tau} P\left(\omega_\tau \mid \omega_{\tau-1}, \mathcal{T}\left(\widetilde{G}_\tau\right)\right) \tag{3}$$

where $\widetilde{G}_\tau$ means the selected subset of the graph G_τ, then we select the corresponding training data for fine-tuning. Finally, the trained model will be used as a predictor in next half a month.

Method

In this section, we provide a detailed description of the proposed framework, which includes a disentangled spatial-temporal graph learning model (DSTG) and a decoupling training strategy. The details are shown in Fig. 1 and the pseudo-code is shown in Algorithm.1.

Spatial-temporal graph learning

Our DSTG consists of three modules: an input module, a disentangled spatial-temporal module, and an output module. In the input module, the input spatial-temporal graph signal is separated into seasonal factors and trend factors using the moving average kernel layer. Additionally, inspired by (Zhou et al. 2021), we incorporate time position information into the model. The proposed disentangled spatial-temporal graph network consists of a GCN layer and a disentangled temporal layer with residual technology. The GCN are employed to capture spatial correlations among nodes, while the disentangled temporal module interprets complex temporal patterns as seasonal and trend patterns. Finally, fully connected layers serve as the decoder to generate predictions as shown in Fig. 1.

Graph convolutional network Recently, GCNs have shown attractive performance in handling non-Euclidean data (Huang et al. 2023b,a), which is widely used in spatial-temporal learning tasks. Given the input at l-th layer $H^l \in \mathbb{R}^{N \times d_l}$, the computing process of a GCN layer can be as follows:

$$\text{GCN}\left(H^l, G\right) = \sigma\left(\left(A + I\right) H^l W^l + b_\tau^l\right) \tag{4}$$

where $W^l \in \mathbb{R}^{d_l \times d_{l+1}}$ and $b_\tau^l \in \mathbb{R}^{d_l}$ are learnable parameters, and A represents the adjacency matrix of the graph G, and I is the corresponding degree matrix.

Disentangled temporal module This module is an asymmetric structure, which consists of a seasonal component and a trend component to disentangle complex temporal patterns. The seasonal component exploits transformer architecture to capture long-term seasonal patterns, owing to its powerful capability to explicitly model long-term dependencies adaptively via the pairwise query-key interaction (Woo et al. 2022). The trend component is based on the lightweight and efficient TCN (Hewage et al. 2020) architecture to learn short-term patterns.

Specifically, given the spatial-temporal representation of a node $H \in \mathbb{R}^{T_h \times d_h}$, we decompose this time series into a trend part H^t and a seasonal part H^r by a moving average kernel. And then the seasonal part H^r is input to the seasonal module, which uses self-attention to capture long-term dependencies: $\text{Att}\left(H^r\right) = \text{softmax}\left(\frac{\left(H^r W^Q\right)\left(H^r W^K\right)^\top}{\sqrt{d}}\right)\left(H^r W^V\right)$, where W^Q, W^K, and W^V are learnable parameters. Then a position-wise feedforward layer is applied to generate outputs. The efficient transformer architecture is a viable alternative (e.g., DLiner (Zeng et al. 2023) or Autoformer (Wu et al. 2021)). The trend part H^t is input into a TCN layer to capture short-term patterns, which are stacked by causal convolution layers. Finally, the output vectors from both components are combined by taking their sum, which is then fed into the output module for generating predictions.

Decoupled training strategy

To facilitate the processing of the dynamic spatial-temporal graph by DSTG, we implement a decoupled training strat-

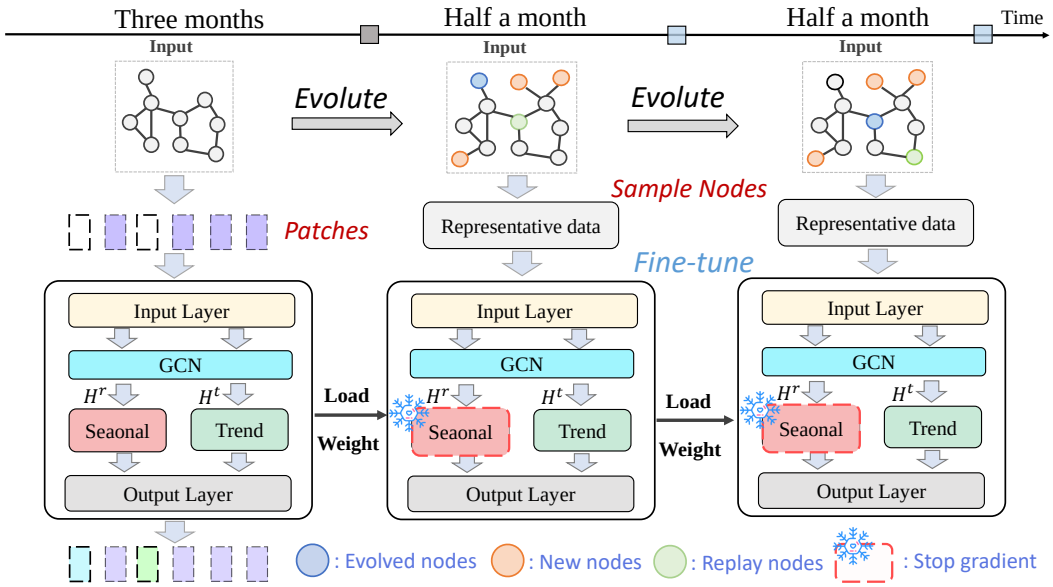

Figure 1: The proposed online learning framework. We update two patterns with unequal-length data. Seasonal patterns are updated with three-month data. We fine-tune the model (except the weights of seasonal patterns) to promptly adapt to the dynamic graph by sampling representative nodes.

egy involving updating two patterns in an alternating manner. The strategy entails utilizing three-month archive data for comprehensive training of the model, thereby enabling it to effectively capture complex seasonal patterns. Throughout this period, we recommend biweekly fine-tuning of the weights of both the trend module and GCNs in order to adapt to the dynamic nature of the graph. To further enhance efficiency, we propose a subnet sampling strategy, whereby representative data is carefully selected for fine-tuning.

Training for seasonal patterns We aim to utilize three-month archived data to thoroughly train the model for capturing spatial-temporal patterns, particularly seasonal patterns. To enhance the model's ability to analyze complex patterns, we introduce a masked autoencoding mechanism, which draws inspiration from computer vision models (He et al. 2022). Specifically, to implement this mechanism, we start with a continuous long time series as the input. Next, we can divide this input sequence into P patches which can either overlap or not following by (Nie et al. 2022) where P is set to 168 in this paper. Each patch has a shape of $(L \times N_\tau \times F)$, where L represents the length of one patch (for example, in PeMS dataset, L is set to 12). Then we create a challenging self-supervised task by randomly masking a subset of patches with a masking ratio r. This strategy can reduce computational complexity while providing sufficient long-term information. Once the model has learned from this task, we further fine-tune it for downstream prediction tasks.

Fine-tuning for new knowledge The streaming nature of the dynamic graph introduces previously unseen patterns and neighborhood information. To promptly incorporate these knowledge, we select a subset of each graph to continuously fine-tune the weights of trend modules and GCNs. The core concept behind this subset sampling strategy is to detect evolved nodes whose patterns have changed significantly for expanding unseen patterns and replay nodes whose patterns are consistent for reinforcing learned patterns. This strategy can enhance the efficiency of the training process and allow the model to efficiently adapt to updated graphs.

Specifically, to compute the evolution degree of each node, we first select the archived data (half-month data) used in the last update, which is termed as Z_r and the one used for the current update Z_c. Then we sum these two sequences along the time dimension and obtain daily average flow vectors, which are termed as $\widetilde{Z}_r \in \mathbb{R}^{N_r \times L_h \times F}$ and $\widetilde{Z}_c \in \mathbb{R}^{N_c \times L_h \times F}$, where L_h is equal to time-steps of one-day data (e.g., $L_h = 288$ if the sampling frequency is five minutes), N_r and N_c represent the number of nodes recorded in the archived data, please note that we only consider nodes that have been recorded in Z_r. Given two such vectors of node v_i, $\widetilde{Z}_r^i = [x_r^t | t = 0, \cdots, L_h] \in \mathbb{R}^{L_h \times F}$ and $\widetilde{Z}_c^i = [x_c^t | t = 0, \cdots, L_h] \in \mathbb{R}^{L_h \times F}$, where x_c^t represents a data point. Then we calculate the similarity between $\widetilde{Z}_r^i$ and $\widetilde{Z}_c^i$ based on Wasserstein distance (Adler and Lunz 2018).

$$\inf_{\gamma \in \Pi[\widetilde{Z}_r^i, \widetilde{Z}_c^i]} \int_v \int_u \gamma(u,v) \left(1 - \frac{x_r^u x_c^v}{\sqrt{\sum_{t=1}^{L_h} (x_r^t)^2} \sqrt{\sum_{t=1}^{L_h} (x_c^t)^2}}\right) dudv$$
$$\text{s.t.} \int \gamma(u,v) du = \frac{x_r^u}{\sqrt{\sum_{t=1}^{L_h} (x_r^t)^2}}, \int \gamma(u,v) dv = \frac{x_c^v}{\sqrt{\sum_{t=1}^{L_h} (x_c^t)^2}} \tag{5}$$

A high distance means that the patterns of this node have

Algorithm 1: Decoupled Training Strategy for Dynamic Spatiotemporal Graph Learning

Input: A dynamic graph $\mathbb{G} = (G_0, G_1, \cdots, G_\tau, \cdots)$ and corresponding observation data.
Output: A prediction model $\mathcal{F}$.

while $\mathbb{G}_\tau$ keeps evoluting **do**
  if $(\tau \% 3 == 0)$ **then**
    Training $\mathcal{F}_{\tau-1}$ with three-month data.
  else
    Fine-tuning the saved model every half month:
    (1). Select new nodes, significantly evolved nodes, and replay nodes to construct a subgraph.
    (2). Fine-tune the model parameters (except seasonal module) using this subgraph.
  end if
  Return the prediction model $\mathcal{F}_\tau$
end while
return solution

changed significantly, which may be attributed to alterations in node features or the introduction and disappearance of the neighborhood. In this paper, we sample the top 1% of nodes with high distances, as well as newly added nodes (if there have been changes in the graph structure), along with their N-hop neighbors. This subgraph is also used for fine-tuning.

By continuously fine-tuning the process of acquiring new information, there is a possibility that previously acquired knowledge may be replaced or forgotten, even though it remains valuable for generating accurate node representations (Wang et al. 2023b). To overcome this challenge, we propose replaying the learned patterns to consolidate old knowledge. To evaluate the consistency of node patterns with the model's learned patterns, we reconsider the Wasserstein distance of each node in Eq.5. A low value indicates that the node patterns align well with the pattern used for training in the previous process. We choose the top 4% of nodes with the smallest distances as replay for fine-tuning.

Experiment

Experiment setting

Dataset The used dataset is collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS) in real-time every 30 seconds in one year. The data is aggregated into 5-minute intervals from 30-second data instances. This dataset includes total 1408 nodes, and we artificially created a traffic dataset with dynamic road networks by masking 5% of the nodes every month from December to January.

Setting We optimize all the models by the Adamw (Loshchilov and Hutter 2017) optimizer. During the self-supervised learning phase, the initial learning rate is set to 10^{-3}, and it is equal to 10^{-4} in the fine-tuning phase. The maximum epoch is 100 with an early stop strategy. We set the forecasting length and lookback length to 12. For two training strategies, we split the training data along the temporal dimension into training datasets and validation datasets with a ratio of 7:3. The test dataset is obtained following by description in the Section problem formulation. All model hyperparameters are chosen through a carefully parameter-tuning process independently. We reported the total time of training and validation for efficiency evaluation, and three widely used metrics, MAE, MAPE, and RMSE are used to evaluate prediction performance, and we report the average metrics in three granularities (i.e. 3 horizons, 6 horizons, and 12 horizons).

Baseline Methods The baselines for comparison primarily include three categories: retraining methods, continuous learning methods, and online learning methods.

For retraining methods, some advanced models are used as baselines[1], including (1) **TCN** which is effective in learning local and global temporal correlations; (2) **STGCN** (Yu, Yin, and Zhu 2017) which employs graph convolution and gated CNN to capture spatial-temporal patterns; (3) **STNN** (Yang, Liu, and Zhao 2021) which employs graph convolution and TCN layers to learn the universal spatial-temporal correlations; (4) **ST-GAM** (Wang et al. 2022) which is an encoder-decoder architecture to alleviate error propagation among predicted time steps; (5) **DSTG+AD** which can decouple the time patterns into the seasonal patterns and the trend patterns using **A**ll **D**ata.

For continuous learning methods, we completely train the model without decoupling mechanism every three months with all data, during which time we fine-tune the model based on different strategies once a month. The methods include (6) **DSTG+SK**: STKEC (Wang et al. 2023a) which uses a memory module to store long-term patterns, and they also select some important nodes based on the influence function to consolidate old knowledge; (7) **DSTG+TS**: TrafficStream (Chen, Wang, and Xie 2021) which proposes an experience-replay strategy, and it samples some nodes based on JS divergence and EWC regularized parameter term (Kirkpatrick et al. 2017) to consolidate previous knowledge; (8) **DSTG+NN** which selects only **N**ew **N**odes with their N-hops neighbors to fine-tune all parameters.

Online learning methods include (9) **DSTG-Static** which trains the model every three months and directly predicts traffic during this period without any further fine-tuning; (10) **DLF** which is our proposed framework and updates dual-scale pattern alternately with unequal-length data.

Method performance and efficiency analysis

The average prediction performance for all nodes over nine months is shown in Table. 1. We can observe that STNN achieves better performance than STGCN because it constructs a local-spacetime context of a traffic sensor comprising the data from its neighbors. The proposed DSTG achieves better prediction performance due to the decoupling of complex time patterns into seasonal patterns and trend patterns, which prevents the model from capturing spurious time patterns. However, these models with retraining methods are trained with all the data every month,

[1]The selected baselines should be independent of the size of the graph, so we have made slight adjustments to a few non-essential function codes in some models for this purpose.

Table 1: Average prediction performance and total traning and validation time of all methods for **all nodes** over nine months.

Model		15 min			30 min			60 min			Time
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	Time(s)
Retrained	TCN	30.35	44.78	21.15	34.78	47.12	24.68	38.72	54.23	27.71	42349.93
	STGCN	24.14	38.24	17.04	28.36	40.14	19.35	31.65	45.61	23.36	76750.18
	STNN	23.05	35.37	16.15	26.12	39.87	18.06	30.64	46.88	23.14	119083.72
	ST-GAM	24.13	37.09	17.21	27.87	41.23	19.61	33.17	50.71	25.37	145146.83
	DSTG+AD	23.27	35.54	16.03	25.14	39.06	17.28	28.25	43.76	22.41	95418.83
Continual	DSTG+NN	27.31	42.37	20.49	29.77	44.90	24.33	33.76	49.92	28.47	52824.36
	DSTG+SK	24.65	38.13	17.93	27.44	41.82	20.81	29.55	46.49	26.03	60412.90
	DSTG+TS	26.43	40.20	19.19	28.56	43.91	22.04	31.07	47.36	25.40	58339.42
Online	DSTG-Static	30.67	44.83	25.70	32.63	46.94	27.83	36.41	51.47	31.93	33852.42
	DLF (Ours)	**22.78**	**35.25**	**15.81**	**24.88**	**38.44**	**17.02**	**28.16**	**43.22**	**21.95**	47531.21

so they consume high training time. For continual methods, they sample some data with different strategies for fine-tuning the model to reduce computational complexity. For example, STKEC calculates the importance of each node and selects some important nodes as replay nodes for consolidating old knowledge. Although the efficiency of DSTG+SK is significantly improved, their performance is worse than retraining methods. In short, both retraining and continuous learning strategies exhibit larger errors than our framework, because the changes in the dynamic road network structure and the evolution of traffic distribution features render the model trained in the previous month obsolete.

Comparing other methods, DSTG-Static achieves high errors because it fails to incorporate new knowledge. DSTG+NN is worse than DLF because it catastrophically forgets the learned patterns during fine-tuning. Our framework achieves better performance because updating different patterns alternately is beneficial for learning more spatial-temporal patterns. Overall, our framework is superior in terms of performance and efficiency, which is explained theoretically in the following section.

Ablation studies

To verify the effectiveness of the individual components in our proposed framework, we make the following variants: (1). w/o Dec: we remove time decomposition mechanism and only employ a TCN layer to capture temporal patterns; (2). w/o Sel: we remove self-supervised learning strategy and follow general traffic prediction tasks; (3). w/o Fre: we use half a month's data to fine-tune all the parameters in fine-tuning; (4). OnlyNew: we use only new nodes to fine-tune the model (exclude the weights of seasonal patterns).

As shown in Fig 2, it is evident that our framework outperforms other variants, thus validating the efficacy of each component in our model. When we no longer freeze weights of seasonal patterns, the model may learn spurious seasonal patterns from short-term time series in fine-tuning, thus w/o Fre achieves poor performance. OnlyNew has higher errors because this model only focuses on new patterns and ignores patterns that have changed significantly.

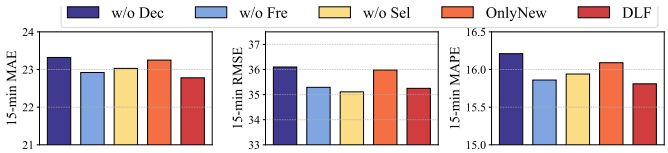

Figure 2: Ablation experiment of component effectiveness.

Method scalability analysis

The dynamic nature of the graph introduces new neighbor information and temporal patterns, and it is valuable to evaluate the scalability for these new knowledge. We present their prediction performance on newly added nodes with their neighbors. Table 2 shows the average 60-minute performance.

GCN-based models can learn new neighbor information through their inductive structure. However, their performance is still limited because their parameters fail to consistently absorb new knowledge from the dynamic graph. Our framework DLF achieves strong scalability through the collaboration between captured seasonal patterns and fast learning ability. On the one hand, explicitly preserved seasonal patterns can provide historical information of the entire road network. New and evolved nodes can access this information to enhance spatial-temporal representation. On the other hand, representative samples based on the subset sampling strategy can efficiently integrate new information into the model, allowing it to adapt to the updated graph.

Generalization on another dataset

To evaluate the generalization performance of the framework, we evaluate it on the Knowair dataset from the atmospheric domain, which include in total of 184 nodes. The task goal is to predict the next 12 horizon $PM_{2.5}$ concentrations given the $PM_{2.5}$ observed concentrations at starting point and next 12 horizons weather forecasting data. We generate a dynamic spatial-temporal atmosphere graph by a masking strategy. We reported the MAE, RMSE, and critical success index (CSI) which is widely used in this domain,

Table 2: Average 60-min prediction performance for new nodes over nine months.

Model		MAE	RMSE	MAPE
Retrained	STGCN	34.89	50.31	25.27
	STNN	33.64	51.88	25.63
	DSTG+AD	32.18	48.46	23.98
Continual	DSTG+SK	36.17	53.27	33.03
	DSTG+TS	35.64	56.16	37.22
	DSTG+NN	39.77	54.69	32.47
Online	DSTG-Static	41.67	59.47	36.93
	DLF (Ours)	**30.24**	**45.21**	**23.17**

please note the higher CSI value means better performance. The results are shown in Table 3.

We observe that DLF can still achieve excellent performance and high efficiency in this dataset. The proposed DSTG has better prediction performance due to the decoupling of complex temporal patterns into seasonal patterns and trend patterns, which prevents the model from capturing spurious information. The performance of continuous learning strategy is lower than that of retraining method. DLF achieves lower errors than other methods because updating different patterns alternately can help the model better adapt to the dynamics of graphs.

Complexity and Uncertainty Analysis

Computational complexity The time complexity of a GCN layer is $\mathcal{O}\left(N^2\right)$ because of the matrix multiplication (Chen, Wang, and Xie 2021). Thus, for a dynamic graph, the time complexity of GCN layers in the retraining method in this paper is $\mathcal{O}\left(N_1^2+\left(N_1^2+N_2^2\right)+\left(N_1^2+N_2^2+N_3^2\right)+\cdots\right)$, where N_τ means the number of nodes in G_τ. Our proposed framework involves training the model every three months, utilizing all nodes. During this time, we use incrementally train the model twice a month using only new nodes and a few sampled nodes, thus, the time complexity reduces to $\mathcal{O}(N_1^2+\left((\Delta N_2+n_2)^2+n_2^2\right))+\left((\Delta N_3+n_3)^2+n_3^2\right)+\left((\Delta N_4+n_4)^2\right)+\cdots)$, where ΔN_τ means the number of new nodes in τ-th month and is equal to $|\mathcal{V}_\tau-\mathcal{V}_{\tau-1}|$, and n_τ means the number of sampled nodes. In a contemporary transportation system, the number of nodes is vast and new nodes typically constitute only a minor fraction. For example, in the third month, $\Delta N_3 \ll N_3$ and n_3 are also set to a small value, so the training time $\mathcal{O}\left(\left(\Delta N_3+n_3\right)^2+n_3^2\right)$ is far less than $\mathcal{O}\left(N_3^2\right)$. Thus, once the graph changes, retraining a new model poses significant time complexity. Overall, our framework is theoretically efficient.

Uncertainty quantization The dispersion or error of an estimate can be characterized by uncertainty, and a lower level of uncertainty indicates a smaller margin of error of predictions. Inspired by (Kendall and Gal 2017; Bai, Ling, and Zhao 2022), our analysis shows that our online learning framework has less uncertainty of prediction distribution compared to the retraining method.

Table 3: Prediction performance for all nodes on Knowair.

Model	MAE	RMSE	CSI	Time
STGCN	30.84	40.76	46.10%	37608.35
STNN	32.23	41.59	48.24%	40316.49
DSTG+AD	30.43	38.81	51.32%	34412.67
DSTG+TS	32.99	42.93	45.37%	20541.20
DSTG+SK	34.22	44.04	45.36%	24564.55
DLF(Ours)	**29.37**	**38.31**	**53.05%**	**18531.42**

PROOF. Let us rethink the prediction distribution in Eq.1, the first term is $\Pr\left(\hat{y} \mid \mathbf{x}_T, \omega_\tau\right)$, which means that the given model parameterized by ω_τ takes a test sample $\mathbf{x}_T$ to make a prediction, hence the variance of $\Pr\left(\hat{y} \mid \mathbf{x}_T, \omega_\tau\right)$ only depends on the noise or randomness coming from $\mathbf{x}_{T+1}$. Thus, given the same test sample, the uncertainty difference for two training strategies only derives from the variance of the second term of Eq. 1, namely $\Pr\left(\omega_\tau \mid \mathcal{D}_\tau\right)$. Let us refer back to Eq.2 and Eq.3, which indicate that the variability of the two approaches is influenced by the randomness of the training data samples, which can be quantified using dataset variance. We achieve proof by demonstrating the following chain inequality: $\operatorname{Var}\left(\mathcal{T}\left(\widetilde{G}_\tau\right)\right) \leq \operatorname{Var}(\mathcal{T}\left(G_\tau\right)) \leq \operatorname{Var}(\mathcal{T}\left(G_{1:\tau}\right))$, where $\operatorname{Var}(\cdot)$ represents the variance of data. Please refer to the appendix for complete proof.

Discussion

We have taken an initial step towards comprehending dynamic space-time graphs. However, our framework does have some limitations that are worth addressing for future research. Firstly, Our developed model, DSTG, utilizes a simple spatial-temporal graph learning model. It is evident that prediction performance could be enhanced by incorporating models with more powerful representation abilities. (2). A challenging aspect is how to generalize learned knowledge to unseen nodes without historical data, which requires the model to possess strong inductive learning capabilities. Unfortunately, the existing inductive capability of STGNNs is inefficient, which has not yet received sufficient attention. Lastly, an important research direction is to explore few-shot learning techniques to enhance the learning of dynamic spatial-temporal graphs, particularly for newly added nodes.

Conclusion

This paper introduces a novel study on spatial-temporal graph prediction in the dynamic scenario by proposing a decoupled learning framework. Our approach involves a disentangled spatial-temporal graph convolutional network and a decoupled training strategy. The DSTG decomposes temporal correlations into seasonal and trend patterns, and the training strategy updates these patterns alternately to facilitate dynamic spatial-temporal graph learning. Through extensive experiments on real-world datasets, we illustrate the effectiveness, efficiency, and scalability of our framework.

Acknowledgments

This paper is partially supported by the National Key R&D Program of China(NO.2022ZD0160101), the National Natural Science Foundation of China (No.62072427, No.12227901), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No.YSBR-005), Academic Leaders Cultivation Program, USTC.

References

Adler, J.; and Lunz, S. 2018. Banach wasserstein gan. *Advances in neural information processing systems*, 31.

Bai, G.; Ling, C.; and Zhao, L. 2022. Temporal Domain Generalization with Drift-Aware Dynamic Neural Networks. *arXiv preprint arXiv:2205.10664*.

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33: 17804–17815.

Bi, J.; Zhang, X.; Yuan, H.; Zhang, J.; and Zhou, M. 2022. A Hybrid Prediction Method for Realistic Network Traffic With Temporal Convolutional Network and LSTM. *IEEE Transactions on Automation Science and Engineering*, 19(3): 1869–1879.

Chahal, A.; Gulia, P.; Gill, N. S.; and Priyadarshini, I. 2023. A Hybrid Univariate Traffic Congestion Prediction Model for IoT-Enabled Smart City. *Information*, 14(5): 268.

Chen, X.; Wang, J.; and Xie, K. 2021. Trafficstream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning. *arXiv preprint arXiv:2106.06273*.

Gu, B.; Zhan, J.; Gong, S.; Liu, W.; Su, Z.; and Guizani, M. 2023. A Spatial-Temporal Transformer Network for City-Level Cellular Traffic Analysis and Prediction. *IEEE Transactions on Wireless Communications*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16000–16009.

Hewage, P.; Behera, A.; Trovati, M.; Pereira, E.; Ghahremani, M.; Palmieri, F.; and Liu, Y. 2020. Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24: 16453–16482.

Huang, Q.; Shen, L.; Zhang, R.; Ding, S.; Wang, B.; Zhou, Z.; and Wang, Y. 2023a. CrossGNN: Confronting Noisy Multivariate Time Series Via Cross Interaction Refinement. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Huang, W.; Wan, G.; Ye, M.; and Du, B. 2023b. Federated graph semantic and structural learning. In *Proc. Int. Joint Conf. Artif. Intell*, 139–143.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023. PDFormer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction. *arXiv preprint arXiv:2301.07945*.

Jin, G.; Liang, Y.; Fang, Y.; Huang, J.; Zhang, J.; and Zheng, Y. 2023. Spatio-Temporal Graph Neural Networks for Predictive Learning in Urban Computing: A Survey. *arXiv preprint arXiv:2303.14483*.

Kendall, A.; and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; and Li, P. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International Conference on Machine Learning*, 11906–11917. PMLR.

Li, M.; Tang, Y.; and Ma, W. 2022. Few-Sample Traffic Prediction With Graph Networks Using Locale as Relational Inductive Biases. *IEEE Transactions on Intelligent Transportation Systems*.

Liu, X.; Xia, Y.; Liang, Y.; Hu, J.; Wang, Y.; Bai, L.; Huang, C.; Liu, Z.; Hooi, B.; and Zimmermann, R. 2023. LargeST: A Benchmark Dataset for Large-Scale Traffic Forecasting. *arXiv preprint arXiv:2306.08259*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *arXiv preprint arXiv:2211.14730*.

Ren, Q.; Li, Y.; and Liu, Y. 2023. Transformer-enhanced periodic temporal convolution network for long short-term traffic flow forecasting. *Expert Systems with Applications*, 227: 120203.

Shao, Z.; Zhang, Z.; Wei, W.; Wang, F.; Xu, Y.; Cao, X.; and Jensen, C. S. 2022. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *arXiv preprint arXiv:2206.09112*.

Tian, Y.; Zhang, K.; Li, J.; Lin, X.; and Yang, B. 2018. LSTM-based traffic flow prediction with missing data. *Neurocomputing*, 318: 297–305.

Varga, B.; Pereira, M.; Kulcsár, B.; Pariota, L.; and Péni, T. 2023. Data-Driven Distance Metrics for Kriging-Short-Term Urban Traffic State Prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Wang, B.; Zhang, Y.; Wang, P.; Wang, X.; Bai, L.; and Wang, Y. 2023a. A Knowledge-Driven Memory System for Traffic Flow Prediction. In *International Conference on Database Systems for Advanced Applications*, 192–207. Springer.

Wang, B.; Zhang, Y.; Wang, X.; Wang, P.; Zhou, Z.; Bai, L.; and Wang, Y. 2023b. Pattern Expansion and Consolidation on Evolving Graphs for Continual Traffic Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2223–2232.

Wang, P.; Zhu, C.; Wang, X.; Zhou, Z.; Wang, G.; and Wang, Y. 2022. Inferring intersection traffic patterns with sparse

video surveillance information: An st-gan method. *IEEE Transactions on Vehicular Technology*, 71(9): 9840–9852.

Wang, S.; Li, Y.; Zhang, J.; Meng, Q.; Meng, L.; and Gao, F. 2020. PM2. 5-GNN: A domain knowledge enhanced graph neural network for PM2. 5 forecasting. In *Proceedings of the 28th international conference on advances in geographic information systems*, 163–166.

Wang, X.; Wang, P.; Wang, B.; Zhang, Y.; Zhou, Z.; Bai, L.; and Wang, Y. 2023c. Latent Gaussian Processes based Graph Learning for Urban Traffic Prediction. *IEEE Transactions on Vehicular Technology*.

Wang, X.; Zhang, H.; Wang, P.; Zhang, Y.; Wang, B.; Zhou, Z.; and Wang, Y. 2023d. An Observed Value Consistent Diffusion Model for Imputing Missing Values in Multivariate Time Series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2409–2418.

Woo, G.; Liu, C.; Sahoo, D.; Kumar, A.; and Hoi, S. 2022. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

Xia, Y.; Liang, Y.; Wen, H.; Liu, X.; Wang, K.; Zhou, Z.; and Zimmermann, R. 2023. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. *arXiv preprint arXiv:2309.13378*.

Yan, H.; Ma, X.; and Pu, Z. 2021. Learning dynamic and hierarchical traffic spatiotemporal features with transformer. *IEEE Transactions on Intelligent Transportation Systems*, 23(11): 22386–22399.

Yang, S.; Liu, J.; and Zhao, K. 2021. Space meets time: Local spacetime neural network for traffic flow forecasting. In *2021 IEEE International Conference on Data Mining (ICDM)*, 817–826. IEEE.

Yin, X.; Zhang, W.; and Jing, X. 2023. Static-dynamic collaborative graph convolutional network with meta-learning for node-level traffic flow prediction. *Expert Systems with Applications*, 120333.

Yu, B.; Yin, H.; and Zhu, Z. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 11121–11128.

Zhang, X.; Huang, C.; Xu, Y.; Xia, L.; Dai, P.; Bo, L.; Zhang, J.; and Zheng, Y. 2021. Traffic flow forecasting with spatial-temporal graph diffusion network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 15008–15015.

Zhang, Y.; Wang, B.; Shan, Z.; Zhou, Z.; and Wang, Y. 2022. CMT-Net: A mutual transition aware framework for taxicab pick-ups and drop-offs co-prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 1406–1414.

Zhou, F.; and Cao, C. 2021. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4714–4722.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

Zhou, Z.; Huang, Q.; Lin, G.; Yang, K.; Bai, L.; and Wang, Y. 2022. Greto: remedying dynamic graph topology-task discordance via target homophily. In *The Eleventh International Conference on Learning Representations*.

Zhou, Z.; Huang, Q.; Yang, K.; Wang, K.; Wang, X.; Zhang, Y.; Liang, Y.; and Wang, Y. 2023a. Maintaining the Status Quo: Capturing Invariant Relations for OOD Spatiotemporal Learning.

Zhou, Z.; Yang, K.; Liang, Y.; Wang, B.; Chen, H.; and Wang, Y. 2023b. Predicting collective human mobility via countering spatiotemporal heterogeneity. *IEEE Transactions on Mobile Computing*.