

els [Shao *et al.*, 2022a; Zhang *et al.*, 2023] and Transformer-based models [Xu *et al.*, 2020]. Consequently, when dealing with large-scale graphs, these models demand significant memory allocation that increases with the complexity of model parameters, posing a notable memory burden. While some research efforts aim to improve the training efficiency of STGNNs for large-scale graphs [Wang *et al.*, 2023c; Wang *et al.*, 2023e], we argue that the memory constraint is a decisive factor that directly affects the availability of these advanced models. The memory limitation not only hampers researchers but also poses challenges for Chinese researchers who may face restrictions imposed by manufacturers, limiting their access to high-end GPU devices with substantial memory capacities.

In this paper, we take a step to address the challenge of large-scale spatio-temporal graph learning with limited GPU memory. An instinct is that we can divide a large spatio-temporal graph into multiple small-scale subgraphs, conquer each subgraph individually, and then merge them incrementally. In this way, GPUs only need to store the representation of the subgraph for saving memory. In response, multiple subgraphs are used to train the model sequentially in a task streaming manner. This involves a continual learning paradigm, enabling graph learning from a static large graph to streaming subgraphs. However, the special challenge posed by continual learning is known as the tradeoff between the plasticity of new knowledge and the stability of old knowledge. In such a setting, achieving the balance between shared information among subgraphs and maintaining personalized information for each subgraph is an open question.

To address challenges, we propose a novel framework for large spatio-temporal graph learning, namely LarSTL. Specifically, we first employ the METIS algorithm as a solution to partition a large spatio-temporal graph into multiple subgraphs. Subsequently, we adopt a task streaming approach, treating each subgraph as an independent task, to sequentially train the spatio-temporal graph convolution model. To mitigate the issue of catastrophic forgetting during sequential training, we introduce a continuous learning strategy based on experience replay. This strategy allows us to consolidate the knowledge gained from previous subgraphs by saving a few nodes. Additionally, we utilize feature adaptors to extract personalized features for each subgraph. Through extensive experiments conducted on various large-scale traffic network datasets containing thousands of nodes, we demonstrate that our proposed method achieves competitive predictive performance while maintaining high training efficiency. Our contributions are as follows:

- We take a step to explore large-scale spatio-temporal graph representation learning and propose a divide-and-conquer strategy based on the continuous learning paradigm, namely LarSTL.
- LarSTL first divides a large spatio-temporal graph into small-scale subgraphs and trains a model sequentially in the task streaming manner. In order to effectively consolidate the previously learned knowledge of subgraphs and their individual features, we propose an experiment-replay strategy and feature adaptors from the data and

parametric perspectives.

- We evaluate the proposed LarSTL on three large-scale traffic datasets with thousands of nodes, and the experimental results show that our model can achieve competitive performance and high efficiency.

2 Related Work

Traffic forecasting. Recently, the accuracy of spatio-temporal graph prediction has significantly improved with the emergence of deep learning [Zhou *et al.*, 2023; Wang *et al.*, 2023b; Xia *et al.*, 2023; Huang *et al.*, 2024b; Zhou *et al.*, 2020b]. Spatio-Temporal Graph Neural Networks (STGNNs) are the most representative approaches to capture spatio-temporal dependencies in this field [Miao *et al.*, 2022b; Miao *et al.*, 2024]. For example, D2STGNN [Shao *et al.*, 2022b] employs diffusion graph convolutional networks with RNNs to capture temporal patterns. STAEformer [Liu *et al.*, 2023a] proposes Spatio-Temporal Adaptive Embedding transformer. ST-GAT [Zhang *et al.*, 2019] employs graph attention networks with LSTM for efficient capture of time dependencies to improve the prediction performance of the Transformer. SSTBAN [Guo *et al.*, 2023] follows a multi-task framework by incorporating a self-supervised learner to produce robust latent representations for historical traffic data. Some works move to solving the problem of low training efficiency, for example, STHMLP [Qin *et al.*, 2023] and ST-MLP [Wang *et al.*, 2023f] are committed to introducing an all-MLP architecture. PFNet [Wang *et al.*, 2023c] captures temporal correlations using deep multi-view sequence encoders and spatial correlations using graph embedding technologies. CANet [Wang *et al.*, 2023e] effectively captures spatial dependencies by mining the implicit spatial centers of variables. Overall, when dealing with large road networks, these models are difficult to scale to large network structures due to GPU memory requirements.

Large-scale graph learning. Large-scale graph learning has garnered significant interest in various domains [Zhou *et al.*, 2020a; Wu *et al.*, 2020], such as recommended tasks and graph classification tasks. For example, GraphSage [Hamilton *et al.*, 2017] employs a sampling strategy where a tree is sampled with each node as the root, progressively expanding the neighborhood by K steps with a fixed sample size. This method computes the hidden representation of the root node by hierarchically aggregating hidden representations of nodes from the bottom to the top of the sampled tree. PinSage [Ying *et al.*, 2018] proposes short random walks to generate subgraphs for convolution operation to generate graph representation for web-scale recommendation systems. Additionally, Cluster-GCN [Chiang *et al.*, 2019] leverages a graph pooling algorithm to sample a few nodes and subsequently performs graph convolutions on this node. SSE [Dai *et al.*, 2018] updates node hidden states recurrently in a stochastic and asynchronous fashion. It alternatively samples a batch of nodes for state update and a batch of nodes for gradient computation. However, these models specifically designed for recommendation and node classification tasks cannot model spatio-temporal dependencies and cannot be directly applied to spatio-temporal graph learning.

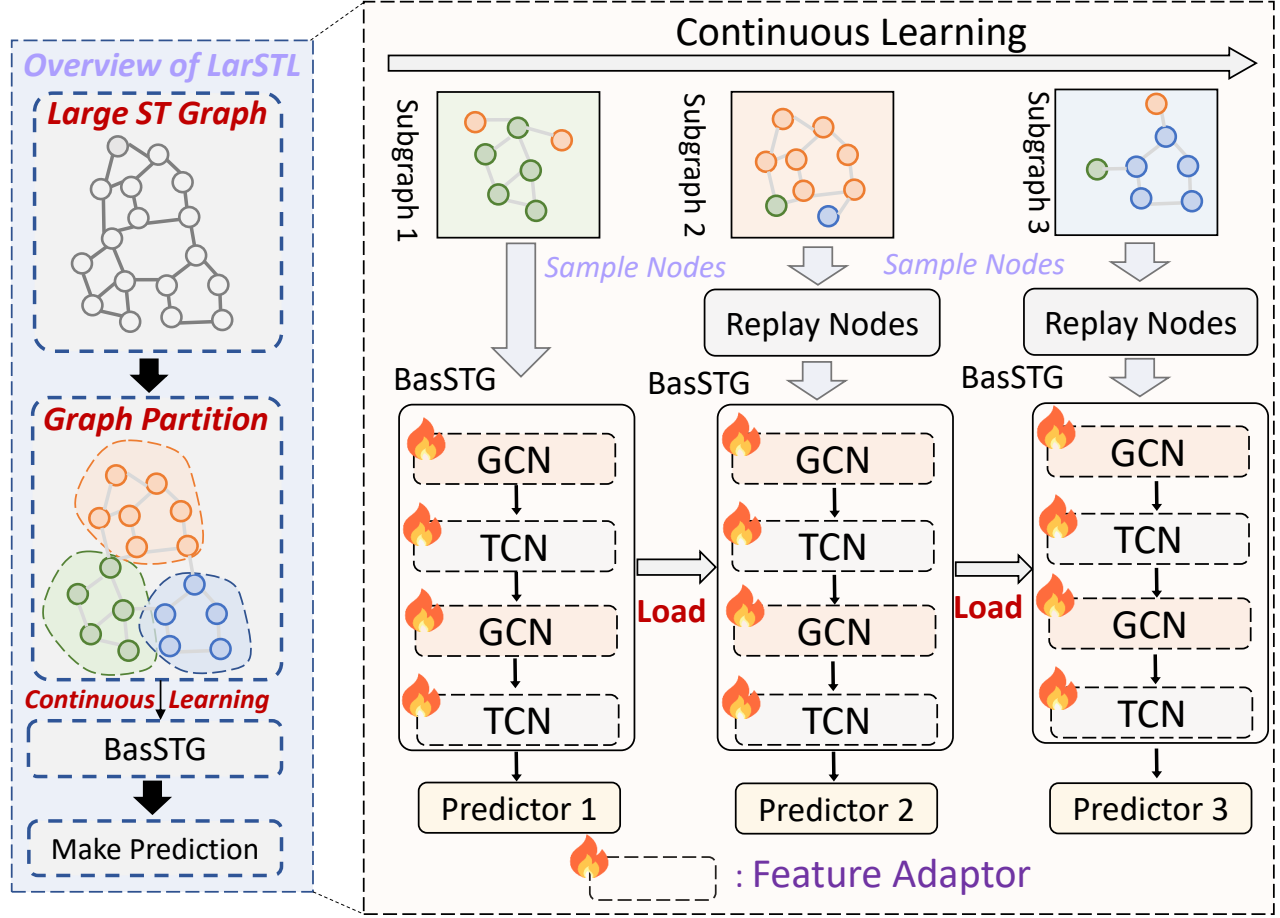


Figure 1: Overall structure of the method. The proposed method LarSTL uses a continuous learning paradigm to train models with multiple partition subgraphs.

3 Problem Definitions

We use a graph $G = (V, E, A)$ to represent a large-scale traffic network, where V denotes the set of nodes, and E represents the weight between the nodes. $A \in \mathbb{R}^{N \times N}$ represents the adjacency matrix of G , where N means the number of nodes and is very large for a large-scale graph. Additionally, let $X_t = \{x_1, x_2, \dots, x_N\}$ represent the historical graph data at t -th time step, and $x_i \in \mathbb{R}^d$ represents the state of i -th node, where d means the number of features. Given G and observations of past T time steps $\mathbf{X} \in \mathbb{R}^{T \times N \times d} = \{X_1, \dots, X_T\}$, The spatio-temporal graph prediction task aims to learn a spatio-temporal prediction model with parameter ω , which can effectively predict graph signals next T_p time steps $\bar{Y} \in \mathbb{R}^{T_p \times N \times d} = \{X_{T+1}, \dots, X_{T+T_p}\}$. The parameters ω can be decoupled into two parts: spatio-temporal feature extractor ϕ and a predictor φ which can be optimized in an end-to-end manner:

$$\omega(\phi, \varphi) = \operatorname{argmin}_{\phi, \varphi} \|\bar{Y} - Y\|^2 \quad (1)$$

where $\bar{Y}$ and Y represent the prediction and ground-truth values, respectively.

4 Method

In this section, for clarity, we first introduce a simple spatio-temporal graph model as the backbone, namely BasSTG. Subsequently, we describe the proposed division-and-conquest strategy, which is based on the continuous learning paradigm to facilitate large-scale spatio-temporal graph learning with BasSTG under limited GPU memory. The details and pseudocode of our proposed method LarSTL are shown in Figure 1 and Algorithm 1.

4.1 Spatio-Temporal Graph Model

GCNs have demonstrated their effectiveness for graph structures across various tasks [Huang *et al.*, 2024a; Wang *et al.*, 2024; Du *et al.*, 2023]. We introduce a simple spatio-temporal graph convolution network as a backbone, BasSTG, which includes two spatio-temporal (ST) layers. Each ST layer includes a diffusion graph convolution module and a TCN module. Specifically, given an adjacency matrix $\mathbf{A}_\tau \in \mathbb{R}^{N_\tau \times N_\tau}$ and the representation of the l -th layer $\mathbf{H}^l \in \mathbb{R}^{N_\tau \times T_l \times C_l}$ as input, where N_τ means the number of nodes in the input graph, T_l means the length of time steps, and C_l means the number of channels, a diffusion graph convolution layer is as

follows:

$$\mathbf{H}^{l+1} = \sum_{k=0}^K \mathbf{P}_f^k \mathbf{H}^l \mathbf{W}_1 + \mathbf{P}_b^k \mathbf{H}^l \mathbf{W}_2 \quad (2)$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{C_l \times C_{l+1}}$ are the learnable parameters. $\mathbf{P}_f^k$ and $\mathbf{P}_b^k$ means forward and backward transfer matrices with k-power series, respectively, and they can computed by normalizing $\mathbf{A}_\tau$. The output $\mathbf{H}^{l+1}$ is fed into a TCN layer to capture the dynamics, which is composed of causal convolution networks. After two ST layers, the output representation is fed into a forecaster to generate future traffic flow.

4.2 Continuous Learning for Large Graph

The restricted GPU memory capacity imposes limitations on the size of the input graph. To address this challenge, we propose a divide-and-conquer strategy. Initially, we divide a large spatio-temporal graph into small-scale subgraphs using a graph partitioning strategy. Subsequently, we train the model sequentially by treating these subgraphs as separate task streams. To enable this sequential learning process, we introduce a continuous learning strategy that facilitates the model's ability to acquire new knowledge from the current subgraph while consolidating the knowledge learned from the previous subgraph.

Graph Partition

Several methods for graph clustering and community detection have been developed, such as spectral clustering [Von Luxburg, 2007] and METIS [Karypis and Kumar, 1998]. We choose METIS due to its excellent balance between accuracy and speed. It partitions a graph into a pre-defined number of clusters such that the number of within-cluster links is much higher than between-cluster links to better capture good community structure [He *et al.*, 2023]. However, METIS is limited to finding non-overlapping clusters, which could imply losing important edge information. Especially for edge nodes at the boundary of any partition, the neighboring correlated node will be lost, and this issue will affect the prediction accuracy of the models.

To tackle this challenge, we use an overlapping subgraph partition approach. In this approach, for each partition, we identify and include spatially correlated nodes from other partitions. As a result, nodes located near the boundary of a partition will be present in multiple partitions. This strategy allows for the incorporation of nodes that exhibit spatial correlation across partitions, enabling a more comprehensive representation of the graph's connectivity and spatial relationships. Expanding all neighbors of the edge nodes into the partition is effective, but it also increases the number of nodes within this partition, leading to a high memory burden. Therefore, we downsample the spatially correlated nodes based on the shortest driving distance, which can be computed by Open Source Routing Machine (OSRM) [Luxen and Vetter, 2011].

Continuous Training Strategy

After dividing graph G into multiple overlapping subgraphs $\{g_1, \dots, g_K\}$, where K means the number of subgraphs. This series of subgraphs is treated as a task streaming to train

the model sequentially. The model can continuously absorb new knowledge from the task streaming to adapt to different subgraphs. After such training, the model is expected to extract shared features among subgraphs, while maintaining the personalized features of each subgraph, so that it can effectively make predictions for the entire large graph.

However, when we fine-tune the model with subgraph G_τ , the model may forget previously learned knowledge from the previous subgraph $G_{\tau-1}$, i.e., catastrophic forgetting. This notorious phenomenon causes the decline of the model's prediction performance for previous tasks. To mitigate catastrophic forgetting, continual graph learning has been actively investigated in various fields [Wang *et al.*, 2023a; Liu *et al.*, 2021b]. In this paper, we propose two strategies based on experience replay and parameter adaptor to consolidate old knowledge from data and parameter perspectives.

Experience-replay strategy. In the realm of continuous learning, the effectiveness of replaying data samples from previous tasks in consolidating old knowledge has been demonstrated [Wang *et al.*, 2023a]. Taking inspiration from this approach, we adopt a similar strategy in our framework. After each training iteration, we select a subset of nodes from the subgraph and add them to the node container $\mathbf{B}$. In the subsequent training process, the nodes in the container $\mathbf{B}$, along with their N-hop neighbors, are utilized to fine-tune the saved model alongside nodes from a new subgraph.

In order to improve the consolidation effect, these replay nodes should representatively cover what the model has learned in the previous task. Naturally, the model is compatible with these nodes. Based on the above analysis, during $\tau - 1$ -th learning step, we sort the model's prediction performance for all nodes only within $G_{\tau-1}$ in the validation dataset. Then we select top $K_\tau\%$ nodes with small prediction errors as the replay nodes. However, some nodes with large prediction errors need to be re-learned, which can improve the prediction accuracy. Therefore, $K_c\%$ nodes with large errors are also selected. After training, these nodes are added into $\mathbf{B}$. When learning subgraph G_τ , these nodes are trained together with the new subgraph. In fact, these cross-subgraph nodes facilitate information interaction between different subgraphs.

Feature adaptor. Striking a balance between extracting new knowledge and consolidating previous knowledge poses a challenge in continuous learning. To tackle this issue, we propose the development of individualized feature adaptors for each subgraph. These adaptors are designed to extract personalized features from each subgraph, empowering the model to effectively capture and comprehend the specific spatio-temporal information within each subgraph.

Specifically, we construct an individual adaptor, θ_i , for i -th subgraph, which is a learnable parameter matrix. The shapes of these matrices are consistent with the ones of each parameter matrix in the spatio-temporal feature extractor ϕ . Then this adaptor θ_i is used to scale the parameters ϕ by dot product: $\theta_i \otimes \phi$. In essence, the adaptor can be viewed as a learnable attention weight that characterizes the importance of spatio-temporal features for this subgraph, i.e. personalized spatio-temporal features. And adaptors are trained end-

Algorithm 1 Large-scale spatio-temporal graph learning

Input: A large-scale spatio-temporal graph G , observation historical graph data $\mathbf{X}$.

Parameter: spatio-temporal feature extractor ϕ , feature adaptor θ , Predictor φ .

Output: A spatio-temporal prediction function with optimal parameters $\omega(\phi, \theta, \varphi)$.

```

1: Using METIS algorithm to divide  $G$  into  $k$  overlapping
   subgraphs  $\{g_1, \dots, g_k\}$ .
2: while  $\{g_1, \dots, g_k\}$  do
3:   if  $\tau == 1$  then
4:     Random initialization parameters  $\omega_1(\phi, \theta_1, \varphi_1)$ .
5:     Initial replay node container  $\mathbf{B}$ .
6:     Train the model with the data of the subgraph  $g_1$ .
7:   else
8:     Random initialization parameters  $\theta_\tau$  and  $\varphi_\tau$ .
9:     Load the saved parameters  $\phi$  in  $\omega_{\tau-1}$ .
10:    Load replay-nodes in container  $\mathbf{B}$ .
11:    Use replay-nodes with their N-hop neighbors and
        nodes in  $g_\tau$  to train the model  $\omega_\tau(\phi, \theta_\tau, \varphi_\tau)$ .
12:  end if
13:  Add replay nodes in  $g_\tau$  into the container  $\mathbf{B}$ .
14: end while
    #####Make Prediction#####
15: for  $\{g_1, \dots, g_k\}$  do
16:   Make predictions for the subgraph  $g_\tau$  with
         $\omega_\tau(\phi, \theta_\tau, \varphi_\tau)$ .
17: end for
    
```

to-end along with the model. For smooth learning, we freeze the model parameters trained on the first subgraph and only update feature adapters in the subsequent training process.

Furthermore, different subgraphs can exhibit distinct traffic patterns, for example, the traffic trends in shopping malls and residential areas can vary significantly. Recognizing this, we propose the development of dedicated predictors tailored for specific subgraphs. These specialized predictors are designed to effectively capture the unique features and dynamics of each traffic subgraph.

5 Experiment

In this chapter, we evaluate the effectiveness of LarSTL on a large-scale network dataset with thousands of nodes and answer the following underlying concerns: Q1. What is the prediction performance and efficiency of the model? Please refer to Section 5.2. Q2. Is every component of the method valid? Please refer to Section 5.3. Q3. How sensitive is the model to the number of subgraphs and the graph partition algorithm? Please refer to Section 5.4.

5.1 Experiment Setting

Data. We evaluate the validity of the model on three large traffic datasets, GLA, GBA, and CA, proposed by from [Liu *et al.*, 2023b]. These three emerging datasets, with thousands of nodes, are far larger than those previously used in the traffic forecasting community [Liu *et al.*, 2023b]. All datasets

are collected from the California Department of Transportation (CalTrans) Performance Measurement System (PeMS) and aggregate the traffic readings from 5-minute intervals into 15-minute windows. The details of traffic networks in each dataset are shown in Table.1.

Dataset	CA	GLA	GBA
Nodes	8,600	3,834	2,352
Edges	98,703	201,363	61,246
Degree	23.4	25.7	26
Time Range	2018	2019	2020
Train frames	21,010	21,010	21,010
Val/test frames	7,003/7,004	7,003/7,004	7,003/7,004

Table 1: The details of the three datasets.

Restricted GPU device. All models are trained on a V100 GPU with 16G memory. We give priority to running a model with the optimal hyperparameters from their official codes. And we specify a maximum batch size of 64. If a model is unable to run with this setting, we would reduce the hyperparameter and the batch size of the models until it fully occupies the memory on this V100 GPU. Eventually, if the model still doesn't work, we consider this model to be out of memory.

Experiment setting. We chronologically split the data into train, validation, and test sets, with a ratio of 6:2:2 for all datasets. AdamW [Loshchilov and Hutter, 2017] is used as the optimizer with an initial learning rate of 0.001. we use an early stopping strategy to accelerate the training process. We divide the entire graph into three subgraphs. To evaluate the performance of our framework, we adopt Mean Absolute Errors (MAE), Root Mean Squared Errors (RMSE), and Mean Absolute Percentage Errors (MAPE) as the metrics for effectiveness. The replay node numbers K_τ and K_c are set to 1%. We report the overall efficiency by considering the total training time and evaluation time of all models, along with the average training time per epoch.

Baseline. We compare some advanced spatio-temporal prediction models as follows:

- LSTM is an RNN variant to model long-term temporal dependencies.
- TCN is a lightweight temporal modeling framework using causal convolution.
- DCRNN [Li *et al.*, 2017] introduces graph diffusion graph convolutional networks and LSTM to capture long-term dependencies.
- Z-GCNETs [Chen *et al.*, 2021] propose to enhance DL architectures with the most salient time-conditioned topological information of the data.
- AGCRN [Bai *et al.*, 2020] is also a RNN-based spatio-temporal graph learning model.
- ASTGCN [Guo *et al.*, 2019] focuses on adaptive modeling of spatial dependencies using attention mechanism.
- HGCN [Guo *et al.*, 2021] designs hierarchical GCN to capture both the spatial relationships of microscopic nodes and relationships between macroscopic regions.

GBA											
Model	Horizon 3			Horizon 6			Horizon 12			All Time	Train Time
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	Hours	Epoch/s
TCN	26.78	42.45	21.87	34.99	49.13	24.66	43.32	71.38	33.09	2.15	113.82
LSTM	25.64	39.15	22.42	31.59	48.95	26.50	42.33	67.05	34.81	4.86	157.66
HGCN	25.57	37.93	21.43	27.19	44.26	23.47	35.32	53.31	27.16	10.26	404.27
Z-GCNETs	25.62	43.27	19.56	30.16	47.53	24.92	38.59	53.98	32.40	15.42	526.38
ASTGCN	24.71	36.42	22.73	26.09	40.57	26.59	36.36	50.42	33.58	39.86	1702.39
DCRNN	<u>19.17</u>	31.30	<u>17.72</u>	23.74	36.91	21.16	<u>30.70</u>	<u>46.24</u>	26.04	71.08	2155.61
AGCRN	19.02	<u>29.64</u>	17.26	<u>23.61</u>	34.92	19.81	32.33	46.26	<u>25.42</u>	26.41	726.73
STGODE	19.32	29.48	20.11	24.26	36.69	23.17	31.34	47.04	27.97	34.83	902.79
DGCRN	23.02	35.82	20.67	26.63	42.39	23.42	34.25	53.43	29.28	170.42	5742.98
DSTAGNN	25.41	37.28	24.01	27.62	45.66	26.47	35.49	55.32	30.78	63.72	2841.27
D ² STGNN	21.42	31.10	19.74	23.75	40.10	22.86	32.36	49.91	26.17	188.05	6681.23
Ours	19.28	30.45	17.84	23.15	<u>35.96</u>	<u>20.43</u>	30.02	45.47	25.11	8.97	47.34

GLA											
Model	Horizon 3			Horizon 6			Horizon 12			All Time	Train Time
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	Hours	Epoch/s
TCN	27.23	44.21	18.81	35.77	50.46	24.42	45.32	75.41	35.49	4.68	156.42
LSTM	24.86	37.92	13.94	32.97	48.48	21.87	43.20	67.42	27.31	7.04	242.24
HGCN	25.75	43.77	14.93	31.35	46.14	20.80	40.28	62.77	28.43	17.91	846.17
Z-GCNETs	23.04	37.05	13.68	28.28	43.56	17.45	37.90	57.68	23.70	23.48	1024.65
ASTGCN	24.73	41.27	14.97	31.18	51.31	19.97	45.03	61.60	26.42	68.17	4015.29
DCRNN	22.02	33.95	12.10	27.67	41.74	16.05	36.17	56.26	21.88	134.72	3526.62
AGCRN	21.01	32.88	11.51	<u>26.80</u>	<u>40.94</u>	<u>15.21</u>	<u>35.02</u>	54.44	21.04	132.87	3952.64
STGODE	<u>20.74</u>	<u>32.41</u>	<u>11.48</u>	27.06	41.21	15.55	35.06	<u>54.27</u>	<u>20.76</u>	55.93	1953.21
DSTAGNN	23.08	36.01	12.78	29.96	48.16	17.80	40.15	60.21	24.26	227.51	9440.83
Ours	20.35	31.92	11.20	26.16	40.27	14.81	34.01	52.89	20.26	12.39	114.26

CA											
Model	Horizon 3			Horizon 6			Horizon 12			All Time	Train Time
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	Hours	Epoch/s
TCN	26.28	42.19	21.72	35.77	51.34	26.24	43.02	72.49	33.45	7.01	441.39
LSTM	27.12	36.41	20.65	31.26	49.23	28.98	43.09	69.44	32.69	9.51	580.60
HGCN	25.51	38.74	19.51	30.04	47.52	31.69	37.59	61.22	39.72	56.22	3261.98
Z-GCNETs	23.69	36.04	18.33	28.41	43.92	26.76	35.18	57.60	33.49	116.04	4343.24
ASTGCN	26.13	40.46	19.77	30.98	49.02	25.24	39.32	62.28	32.35	269.20	13293.15
DCRNN	22.12	35.16	17.00	26.95	42.71	22.37	35.03	53.36	29.79	237.45	12739.41
STGODE	<u>21.93</u>	<u>34.34</u>	<u>16.89</u>	<u>26.82</u>	<u>42.23</u>	<u>21.93</u>	<u>33.91</u>	<u>53.17</u>	<u>28.00</u>	155.73	6841.09
DSTAGNN	25.76	41.29	19.70	31.03	48.99	25.74	39.45	62.48	33.63	364.83	31271.24
Ours	21.42	34.15	16.46	26.17	41.42	21.41	32.73	52.04	27.78	21.06	143.37

Table 2: Prediction performance and efficiency of each model. 'All Times' means total training time and evaluation time (in hours). We also report time cost per epoch (in seconds) for efficiency comparison. The best results are bold, and the second best results are underlined.

- DGCRN [Li *et al.*, 2023] leverages and extracts dynamic characteristics from node attributes, while the parameters of dynamic filters are generated at each time step.
- STGODE [Fang *et al.*, 2021] uses neural ordinary differential equations to effectively model the continuous changes of traffic signals.
- DSTAGNN [Lan *et al.*, 2022] uses a dynamic spatial-temporal aware graph to replace the predefined static graphs used in traditional graph convolution.
- D²STGNN [Shao *et al.*, 2022b] proposes a unique estimation gate with residual decomposition mechanism to decouple traffic signals into diffusion and inherent traffic information.

5.2 Performance and Efficiency Analysis (Q.1)

The prediction performance of each model on the three datasets is presented in Table 2, showcasing the competitive efficiency and prediction accuracy of our model.

Comparing different methods on the GBA, we observe that temporal models, such as TCN and LSTM, perform poorly because they fail to capture spatial information efficiently. DGCRN and D²STGNN demonstrate impressive performance on the GBA datasets, highlighting the importance of considering the dynamic features of spatial topology. However, their performance can not meet expectations because the device’s memory capacity limited the size of their parameters, resulting in inadequate learning of the spatio-temporal information. Meanwhile, their complex model designs limit their scalability when applied to larger datasets like GLA and CA datasets. On the other hand, RNN-based models, such as DCRNN and AGCRN, maintain good predictive performance across multiple datasets, which can be attributed to their ability to effectively capture spatio-temporal information. However, when running in the limited GPU, these models all struggle to effectively handle the large road network, resulting in a rapid deterioration of prediction performance. In terms of efficiency, TCN-based approaches such as Z-GCNets and HGCN are generally faster than other baselines, thanks to their parallel computation of temporal convolution operations. Among the RNN-based models, DCRNN and AGCRN are slower than TCN-based methods due to the use of RNNs as decoders.

Our model achieves superior performance and efficiency, especially for large-scale datasets, thanks to the proposed divide-and-conquer strategy, which reduces the learning complexity of spatio-temporal information.

5.3 Ablation Experiment (Q.2)

In this section, we evaluate the effectiveness of each component. We make the following variants: w/o tem means that we remove the temporal module in LarST. w/o rep means that we remove the experience-replay strategy for consolidating old knowledge learned from the previous subgraphs. w/o adp means that we directly fine-tune the saved model each time instead of configuring feature adaptors for each subgraph. The prediction results of each variant are shown in Figure.2.

We can see that each variant is more or less inferior to our proposed model, proving the validity of each component of

the model. The prediction performance of w/o tem is not good, this is because it does not model temporal correlations and has not learned sufficient spatiotemporal information, resulting in suboptimal prediction performance. The prediction error of w/o adp is high, which indicates that the feature adaptor can effectively help the model consolidate the old knowledge. At the same time, it shows that learning the personalized features of each subgraph improves the model’s prosperity in the new subgraph.

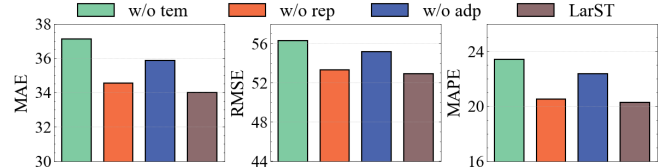


Figure 2: Prediction performance of each variant on GLA dataset.

5.4 Hyperparameter Experiment (Q.3)

We assess the impact of the number of graph partitions on clustering and its subsequent effect on model prediction efficacy. The outcomes derived from a 60-horizon analysis on the GLA dataset are presented in Table 3, revealing that optimal model performance is achieved when K is equal to 3. This because that a higher value of K results in clusters with fewer nodes, which fails to furnish adequate spatio-temporal data, leading to model underfitting. Moreover, such a setup causes the model to readily forget previously learned information when it encounters new knowledge within current subgraphs. Conversely, a small value of K implies a larger number of nodes per cluster, and the scale of model parameters is constrained by GPU memory, resulting in underfitting of the model.

	K	MAE	RMSE	MAPE
GLA	2	34.13	53.32	20.72
	3	34.01	52.89	20.26
	4	34.62	54.07	21.48
	5	35.49	55.64	22.63

Table 3: Sensitivity analysis of hyperparameter K .

6 Conclusion

We propose an efficient approach for large-scale spatio-temporal graph learning, employing a divide-and-conquer strategy. Our method partitions a large graph into smaller subgraphs, allowing individual processing and analysis. To enhance the learning process, we introduce two strategies: experience replay and feature adaptor. Experience replay strategy can consolidate knowledge from preceding subgraphs, while the feature adaptor strategy ensures scalability for incorporating new knowledge from subsequent subgraphs. We comprehensively evaluate our approach on datasets featuring large traffic networks, demonstrating its effectiveness and efficiency in large-scale spatio-temporal graph learning under limited GPUs.

Acknowledgments

This paper is supported by 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), and Academic Leaders Cultivation Program, USTC.

References

- [Bai *et al.*, 2020] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Proc. of NeurIPS*, 2020.
- [Chen *et al.*, 2021] Yuzhou Chen, Ignacio Segovia, and Yulia R Gel. Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting. In *Proc. of ICML*, 2021.
- [Chiang *et al.*, 2019] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proc. of KDD*, 2019.
- [Dai *et al.*, 2018] Hanjun Dai, Zornitsa Kozareva, Bo Dai, Alex Smola, and Le Song. Learning steady-states of iterative algorithms over graphs. In *Proc. of ICML*, 2018.
- [Du *et al.*, 2023] Wenjie Du, Lianliang Chen, Haoran Wang, Ziyang Shan, Zhengyang Zhou, Wenwei Li, and Yang Wang. Deciphering urban traffic impacts on air quality by deep learning and emission inventory. *Journal of environmental sciences*, 2023.
- [Fang *et al.*, 2021] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proc. of KDD*, 2021.
- [Guo *et al.*, 2019] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proc. of AAAI*, 2019.
- [Guo *et al.*, 2021] Kan Guo, Yongli Hu, Yanfeng Sun, Sean Qian, Junbin Gao, and Baocai Yin. Hierarchical graph convolution network for traffic forecasting. In *Proc. of AAAI*, 2021.
- [Guo *et al.*, 2023] Shengnan Guo, Youfang Lin, Letian Gong, Chenyu Wang, Zeyu Zhou, Zekai Shen, Yiheng Huang, and Huaiyu Wan. Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting. In *Proc. of ICDE*, 2023.
- [Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Proc. of NeurIPS*, 2017.
- [He *et al.*, 2023] Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. A generalization of vit/mlp-mixer to graphs. In *Proc. of ICML*, 2023.
- [Huang *et al.*, 2024a] Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In *Proc. of AAAI*, 2024.
- [Huang *et al.*, 2024b] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgcn: Confronting noisy multivariate time series via cross interaction refinement. *Proc. of NeurIPS*, 2024.
- [Karypis and Kumar, 1998] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 1998.
- [Lan *et al.*, 2022] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *Proc. of ICML*, 2022.
- [Li *et al.*, 2017] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.
- [Li *et al.*, 2023] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data*, 2023.
- [Liu *et al.*, 2021a] Chenxi Liu, Zhu Xiao, Dong Wang, Lei Wang, Hongbo Jiang, Hongyang Chen, and Jiangxia Yu. Exploiting spatiotemporal correlations of arrive-stay-leave behaviors for private car flow prediction. *IEEE Transactions on Network Science and Engineering*, 2021.
- [Liu *et al.*, 2021b] Huihui Liu, Yiding Yang, and Xinchao Wang. Overcoming catastrophic forgetting in graph neural networks. In *Proc. of AAAI*, 2021.
- [Liu *et al.*, 2023a] Hangchen Liu, Zheng Dong, Renhe Jiang, Jiewen Deng, Jinliang Deng, Qunjun Chen, and Xuan Song. Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. In *Proc. of CIKM*, 2023.
- [Liu *et al.*, 2023b] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhengguang Liu, Bryan Hooi, and Roger Zimmermann. Largest: A benchmark dataset for large-scale traffic forecasting. *arXiv preprint arXiv:2306.08259*, 2023.
- [Liu *et al.*, 2024] Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*, 2024.
- [Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [Luxen and Vetter, 2011] Dennis Luxen and Christian Vetter. Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2011.
- [Miao *et al.*, 2022a] Hao Miao, Yan Fei, Senzhang Wang, Fang Wang, and Danyan Wen. Deep learning based origin-

- destination prediction via contextual information fusion. *Multimedia Tools and Applications*, 2022.
- [Miao *et al.*, 2022b] Hao Miao, Jiaxing Shen, Jiannong Cao, Jiangnan Xia, and Senzhang Wang. Mba-stnet: Bayes-enhanced discriminative multi-task learning for flow prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2022.
- [Miao *et al.*, 2024] Hao Miao, Yan Zhao, Chenjuan Guo, Bin Yang, Zheng Kai, Feiteng Huang, Jiandong Xie, and Christian S Jensen. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *Proc. of ICDE*, 2024.
- [Qin *et al.*, 2023] Yanjun Qin, Haiyong Luo, Fang Zhao, Yuchen Fang, Xiaoming Tao, and Chenxing Wang. Spatio-temporal hierarchical mlp network for traffic forecasting. *Information Sciences*, 2023.
- [Shao *et al.*, 2022a] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proc. of CIKM*, 2022.
- [Shao *et al.*, 2022b] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *arXiv preprint arXiv:2206.09112*, 2022.
- [Von Luxburg, 2007] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 2007.
- [Wang *et al.*, 2020] Senzhang Wang, Hao Miao, Hao Chen, and Zhiqiu Huang. Multi-task adversarial spatial-temporal networks for crowd flow prediction. In *CIKM*, 2020.
- [Wang *et al.*, 2023a] Binwu Wang, Yudong Zhang, Jiahao Shi, Pengkun Wang, Xu Wang, Lei Bai, and Yang Wang. Knowledge expansion and consolidation for continual traffic prediction with expanding graphs. *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [Wang *et al.*, 2023b] Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proc. of KDD*, 2023.
- [Wang *et al.*, 2023c] Chen Wang, Kaizhong Zuo, Shaokun Zhang, Hanwen Lei, Peng Hu, Zhangyi Shen, Rui Wang, and Peize Zhao. Pfnet: Large-scale traffic forecasting with progressive spatio-temporal fusion. *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [Wang *et al.*, 2023d] Xu Wang, Pengkun Wang, Binwu Wang, Yudong Zhang, Zhengyang Zhou, Lei Bai, and Yang Wang. Latent gaussian processes based graph learning for urban traffic prediction. *IEEE Transactions on Vehicular Technology*, 2023.
- [Wang *et al.*, 2023e] Yuan Wang, Zezhi Shao, Tao Sun, Chengqing Yu, Yongjun Xu, and Fei Wang. Clustering-property matters: A cluster-aware network for large scale multivariate time series forecasting. In *Proc. of CIKM*, 2023.
- [Wang *et al.*, 2023f] Zepu Wang, Yuqi Nie, Peng Sun, Nam H Nguyen, John Mulvey, and H Vincent Poor. St-mlp: A cascaded spatio-temporal linear framework with channel-independence strategy for traffic forecasting. *arXiv preprint arXiv:2308.07496*, 2023.
- [Wang *et al.*, 2024] Binwu Wang, Pengkun Wang, Yudong Zhang, Xu Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. Towards dynamic spatial-temporal graph learning: A decoupled perspective. In *Proc. of AAAI*, 2024.
- [Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.
- [Xia *et al.*, 2023] Yutong Xia, Yuxuan Liang, Haomin Wen, Xu Liu, Kun Wang, Zhengyang Zhou, and Roger Zimmermann. Deciphering spatio-temporal graph forecasting: A causal lens and treatment. *arXiv preprint arXiv:2309.13378*, 2023.
- [Xu *et al.*, 2020] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.
- [Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proc. of KDD*, 2018.
- [Zhang *et al.*, 2019] Chenhan Zhang, James J. Q. Yu, and Yi Liu. Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting. *IEEE Access*, 2019.
- [Zhang *et al.*, 2022] Yudong Zhang, Binwu Wang, Ziyang Shan, Zhengyang Zhou, and Yang Wang. Cmt-net: A mutual transition aware framework for taxicab pick-ups and drop-offs co-prediction. In *Proc. of WSDM*, 2022.
- [Zhang *et al.*, 2023] Zijian Zhang, Ze Huang, Zhiwei Hu, Xiangyu Zhao, Wanyu Wang, Zitao Liu, Junbo Zhang, S Joe Qin, and Hongwei Zhao. Mlpst: Mlp is all you need for spatio-temporal prediction. In *Proc. of CIKM*, 2023.
- [Zhou *et al.*, 2020a] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 2020.
- [Zhou *et al.*, 2020b] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Chaochao Zhu. Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective. *IEEE TKDE*, 2020.
- [Zhou *et al.*, 2023] Zhengyang Zhou, Qihe Huang, Kuo Yang, Kun Wang, Xu Wang, Yudong Zhang, Yuxuan Liang, and Yang Wang. Maintaining the status quo: Capturing invariant relations for ood spatiotemporal learning. 2023.