KITS: Inductive Spatio-Temporal Kriging with Increment Training Strategy

Qianxiong Xu[1], Cheng Long[1*], Ziyue Li[2*], Sijie Ruan[3], Rui Zhao[4], Zhishuai Li[4]

[1]S-Lab, Nanyang Technological University
[2]University of Cologne
[3]Beijing Institute of Technology
[4]SenseTime Research
{qianxiong.xu, c.long}@ntu.edu.sg, zlibn@wiso.uni-koeln.de, sjruan@bit.edu.cn, {zhaorui, lizhishuai}@sensetime.com

Abstract

Sensors are commonly deployed to perceive the environment. However, due to the high cost, sensors are usually sparsely deployed. Kriging is the tailored task to infer the unobserved nodes (without sensors) using the observed nodes (with sensors). The essence of kriging task is transferability. Recently, several inductive spatio-temporal kriging methods have been proposed based on graph neural networks, being trained based on a graph built on top of observed nodes via pretext tasks such as masking nodes out and reconstructing them. However, the graph in training is inevitably much sparser than the graph in inference that includes all the observed and unobserved nodes. The learned pattern cannot be well generalized for inference, denoted as *graph gap*. To address this issue, we first present a novel *Increment* training strategy: instead of masking nodes (and reconstructing them), we add virtual nodes into the training graph so as to mitigate the graph gap issue naturally. Nevertheless, the empty-shell virtual nodes without labels could have inferior features and lack supervision signals. To solve these issues, we pair each virtual node with its most similar observed node and fuse their features together; to enhance the supervision signal, we construct reliable pseudo labels for virtual nodes. As a result, the learned pattern of virtual nodes could be safely transferred to real unobserved nodes for reliable kriging. We name our new Kriging model with Increment Training Strategy as KITS. Extensive experiments demonstrate that KITS consistently outperforms existing methods by large margins, e.g., the improvement over MAE score could be as high as 18.33%.

Code — https://github.com/Sam1224/KITS

Introduction

Sensors play essential roles in various fields like vision (Xu et al. 2023b, 2024a, 2025), traffic monitoring (Zhou et al. 2021), energy control (Liu et al. 2022b), road extraction (Xu et al. 2023a), trajectory learning (Liu et al. 2022a, 2024c), forecasting (Miao et al. 2024; Liu et al. 2024b,a) and anomaly detection (Xu et al. 2024b). For example, loop detectors are installed on roads to perceive traffic dynamics, such as vehicle flows and speeds. Nevertheless, due to the high cost of devices and maintenance (Liang et al. 2019), the actual sparsely deployed sensors are far from sufficient to support various services that require fine-grained data. To address this problem, ***Inductive*** **Spatio-Temporal Kriging** (Wu et al. 2021a) is proposed to estimate the values of *unobserved nodes* (without sensors) by using the values of *observed nodes* (with sensors) across time.

The common settings of inductive kriging are: (1) the training is only based on observed nodes; (2) when there are new unobserved nodes inserted during inference, the model can naturally transfer to them without re-training, i.e., being inductive. To enable such transferability, existing inductive methods mainly adopt the following training strategy (illustrated in Figure 1(a)): it constructs a graph structure on top of observed nodes (e.g., based on the spatial proximity of the nodes (Barthélemy 2011)), randomly masks some observed nodes' values, and then trains a model (which is mostly Graph Neural Network based) to reconstruct each node's value. In this strategy, the graph structure is commonly used to capture node correlations and the inductive GNNs such as GraphSAGE (Hamilton, Ying, and Leskovec 2017) accommodate different graph structures during inference, as shown in Figure 1(b), where the values of new nodes 4-5 will be inferred by the model trained from nodes 1-3. We name this strategy as **Decrement training strategy** since it *decrements* the number of observed nodes during training (by masking their values) and use them to mimic the unobserved nodes to be encountered during inference.

Unfortunately, such strategy inevitably suffers from the *graph gap* issue, i.e., the graph used for training is much sparser than that for inference, and it will aggravate with more nodes unobserved. Expressly, the *training graph* is based only on observed nodes, whereas the *inference graph* would be based on observed and unobserved nodes. There would be a clear gap between the two graphs: (1) the latter has more nodes than the former; and (2) their topologies are different (e.g., the latter could be denser). This graph gap would pose a challenge to transfer from the training graph to the inference graph. For example, as a scenario shown in Figure 1(c), where red pins represent observed nodes, and blue pins represent unobserved nodes, the training graph (based on only red pins nodes) would significantly differ from the inference graph (based on red and blue pins). As shown in our empirical studies in Appendix, the average node degree of the training graphs from the decrement meth-

*Co-corresponding authors

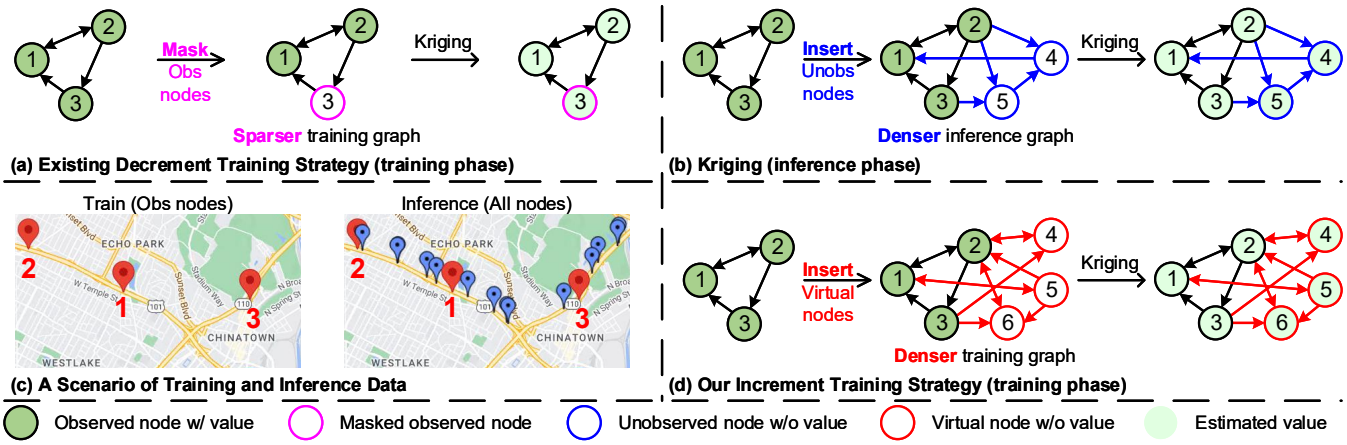

Figure 1: Decrement and Increment training strategies. (a) Decrement training strategy: observe nodes 1-3 during training, and **mask** node-3 out to reconstruct. (b) Kriging (inference phase): observe nodes 1-3, infer the values of new nodes 4-5. (c) A scenario of training and inference data. (d) Increment training strategy: observe nodes 1-3, **insert** virtual nodes 4-6 to mimic the target unobserved nodes in inference, and learn to directly estimate their values.

ods like (Wu et al. 2021a,b) can be more than 70% lower than that of the real inference graph.

To mitigate this issue, we propose a new training strategy, shown in Figure 1(d): (1) it inserts some empty-shell *virtual nodes* and obtains the corresponding expanded graph, (2) it then trains a kriging model based on the graph with the observed nodes' values as labels in a semi-supervised manner. With this strategy, the gap between the training and inference graphs would be naturally reduced since the observed nodes in the two graphs are the same, and the virtual nodes in the former mimic the unobserved nodes in the latter. To further close the graph gap, it inserts virtual nodes in different ways and generates various training graphs (for covering different inference graphs). We name it as **Increment training strategy** since it *increments* the number of nodes of the graph during training. Empirical study in Appendix shows our method narrows the degree difference to only 15%.

However, due to the abundant labels for observed nodes and the absence of labels on virtual nodes, the Increment training strategy faces the *fitting* issue: it could easily get overfitting and underfitting on observed and virtual nodes, leading to superior and inferior features, respectively. We present two solutions: Firstly, we design **Reference-based Feature Fusion** module to align each virtual node with its most similar observed node and vice versa, then fuse their features. As a result, (1) virtual nodes could improve their inferior features with superior features from similar observed nodes; (2) observed nodes, affected by inferior features, are less likely to get overfitting. Secondly, we present a **Node-aware Cycle Regulation** to provide reliable pseudo labels (Cascante-Bonilla et al. 2021) for virtual nodes so that they would be well-regulated. Overall, our contributions are threefold:

- We identify the *graph gap* issue of **Decrement** training strategy that has been adopted by existing kriging methods, and propose a novel **Increment** training strategy.
- We further develop the Reference-based Feature Fusion module and the Node-aware Cycle Regulation for handling the *fitting* issue of the proposed strategy.
- We conduct experiments on eight datasets of three types, showing that our KITS outperforms existing methods consistently by large margins (as high as **18.33%**).

Related Work

Kriging is a widely used technique in geostatistics for spatial interpolation (Krige 1951; Goovaerts 1998), which involves predicting the value of a variable at an unsampled location based on the observed values of the variable at nearby locations. It has a wide range of applications in meteorology (Gad, Manjunatha et al. 2017), geology (Li et al. 2021), transportation (Mao et al. 2022; Li et al. 2022; Han et al. 2022; Li 2021; Lin et al. 2021; Han et al. 2022), oceanology (Tonkin and Larson 2002), and so on. Kriging is normally categorized as transductive setting and inductive setting: (1) Transductive kriging requires all the nodes to be present during training, and it cannot learn the representation for the unseen nodes directly: usually, re-training the model with the new nodes is needed. Classic models such as matrix factorization, DeepWalk, and GCN are by default transductive, which will be introduced in detail. (2) Inductive models instead can directly handle the new nodes that are unseen during training. It can accommodate dynamic graphs and learn the representations of unseen nodes. This is the main scope of this work. Recently, spatio-temporal kriging (Wu et al. 2021a,b) has extended this technique to include the temporal dimension, enabling the estimation of values of unobserved locations at different times.

Transductive Kriging: Matrix factorization and tensor factorization is one of the most representative methods for spatio-temporal kriging (Bahadori, Yu, and Liu 2014; Zhou et al. 2012; Takeuchi, Kashima, and Ueda 2017; Li et al. 2020a,b; Deng et al. 2021; Lei et al. 2022). For example, GLTL (Bahadori, Yu, and Liu 2014) takes an input tensor $\mathcal{X}^{\text{location}\times\text{time}\times\text{variables}}$ with unobserved locations set to zero and then uses tensor completion to recover the values at

the observed locations. GE-GAN (Xu et al. 2020) is another method, which builds a graph on top of both observed and unobserved nodes and utilizes node embeddings (Yan et al. 2006) to pair each unobserved node with the most relevant observed nodes for kriging. It then uses generative models (Goodfellow et al. 2020) to generate values for unobserved nodes. Besides, imputation methods (Lai et al. 2024; Cini, Marisca, and Alippi 2021) can be applied for transductive kriging, e.g., GRIN combines message passing mechanism (Gilmer et al. 2017) with GRU (Cho et al. 2014) to capture complex spatio-temporal patterns for kriging. These methods are limited by their "transductive" setting, i.e., they require the unobserved nodes to be known during the training phase: to handle new unobserved nodes that were not known before, they need model re-training.

Inductive Kriging: More recently, quite a few methods, including KCN (Appleby, Liu, and Liu 2020), IGNNK (Wu et al. 2021a), LSJSTN (Hu et al. 2021), SpecKriging (Zhang et al. 2022), SATCN (Wu et al. 2021b), and INCREASE (Zheng et al. 2023) have been proposed to conduct spatio-temporal kriging in an "inductive setting" (which we call inductive spatio-temporal kriging). That is, during their training phase, the unobserved nodes are not known and they can handle new unobserved nodes without model retraining. These methods mainly adopt the Decrement training strategy: (1) it constructs a graph on top of the observed nodes, (2) it then randomly masks the values of some nodes of the constructed graph (which mimics the unobserved nodes), and (3) it then learns to recover the values of the unobserved nodes. However, as explained in Introduction, this Decrement training strategy would suffer from the *graph gap* issue, i.e., the training graph is based on all observed nodes, while the inference graph is based on both observed and unobserved nodes. In this paper, we propose a new *Increment training strategy*, which inserts virtual nodes to the training graph so as to mitigate the *graph gap* issue - with this strategy, the training graph is based on observed nodes and virtual nodes (which mimic unobserved nodes).

Methodology

Overview

Problem Definition. Let $\mathbf{X}^o_{T-t:T} \in \mathbb{R}^{N_o \times t}$ denote the values of N_o observed nodes in t time intervals. We follow existing studies (Wu et al. 2021a) and construct a graph structure on the observed nodes (e.g., creating an edge between two nodes if they are close enough). We denote the adjacency matrix of the graph structure by $\mathbf{A}^o \in [0,1]^{N_o \times N_o}$. The **inductive spatio-temporal kriging** problem is to estimate the values of N_u unobserved nodes, which are not known until inference, based on $\mathbf{X}^o_{T-t:T}$ and the graph on top of the observed nodes and unobserved nodes.

KITS. To mitigate the issues suffered by existing methods, we first propose a new **Increment training strategy**, which inserts *virtual nodes* to the training graph and aims to estimate the values of all nodes with a kriging model in a semi-supervised manner. We then design a **kriging model**, which involves two components, namely Spatio-Temporal Graph Convolution (STGC) and Reference-based Feature Fusion (RFF). Finally, we incorporate a Node-aware Cycle Regulation (NCR) for regulating those virtual nodes since they lack of labels. An overview of is illustrated in Figure 2.

Increment Training Strategy for Kriging

To mitigate the graph gap issue suffered by the existing Decrement training strategy, we propose a new **Increment training strategy**: It first inserts some empty-shell *virtual nodes* and obtains the corresponding expanded graph, and then trains a kriging model based on the graph with the values of the observed nodes as labels in a semi-supervised manner (see Figure 1(d) for an illustration).

The core procedure of this training strategy is to insert some "virtual" nodes in the training graph to mimic the "unobserved" nodes in the inference graph, yet the unobserved nodes are not known during training. To implement this procedure, two questions need to be answered: (1) how many virtual nodes should be inserted; (2) how to create edges among observed nodes and virtual nodes.

Our solution about virtual nodes is as follows. First, we follow existing studies (Wu et al. 2021a; Hu et al. 2021) to assume the availability of some rough estimate of the unobserved nodes to be encountered during inference. That is, we assume the availability of a *missing ratio* α, the *approximate* ratio of unobserved nodes over all nodes in the inference graph. We denote N_o as the number of observed nodes. Then, we insert N_v virtual nodes, where $N_v = N - N_o = \frac{N_o}{1-\alpha} - N_o = \frac{\alpha \cdot N_o}{1-\alpha}$. Considering inference graphs with varying numbers of unobserved nodes, we add a random noise ϵ to α. Besides, the values of the virtual nodes are initialized as 0, a common practice for nodes without readings.

To answer the second question about virtual edges, we adopt the following graph augmentation method. For each virtual node, we (1) pick an observed node randomly, (2) create an edge between the virtual node and the picked node; and (3) create an edge between the virtual node and each *neighbor node* of the picked node with a probability $p \sim$ *Uniform*$[0,1]$. The rationale is: the virtual node is created to mimic an unobserved node, which should have relations with a local neighborhood of observed nodes, e.g., a sensor has relations with others within a certain spatial proximity.

In addition, we generate multiple batches of training graphs and train batch by batch. Due to the randomness of the above procedure of inserting virtual nodes, we will generate various training graphs yet similar to the inference graph to some extent in different batches (see Figure 2(a) for illustration). This diversity achieves better generality of the model to handle different inference graphs. Note that the pseudo code of Increment training strategy is included in Appendix.

Kriging Model

Spatio-Temporal Graph Convolution (STGC). STGC acts as the basic building block of the kriging model, and is responsible for aggregating spatio-temporal features from neighboring nodes (e.g., nearby nodes) to the current node with graph convolution (Cini, Marisca, and Alippi 2021). Specifically, we denote the input features of STGC as $\mathbf{Z}_i \in \mathbb{R}^{N \times D}$, where the subscript i means the time interval is T_i,

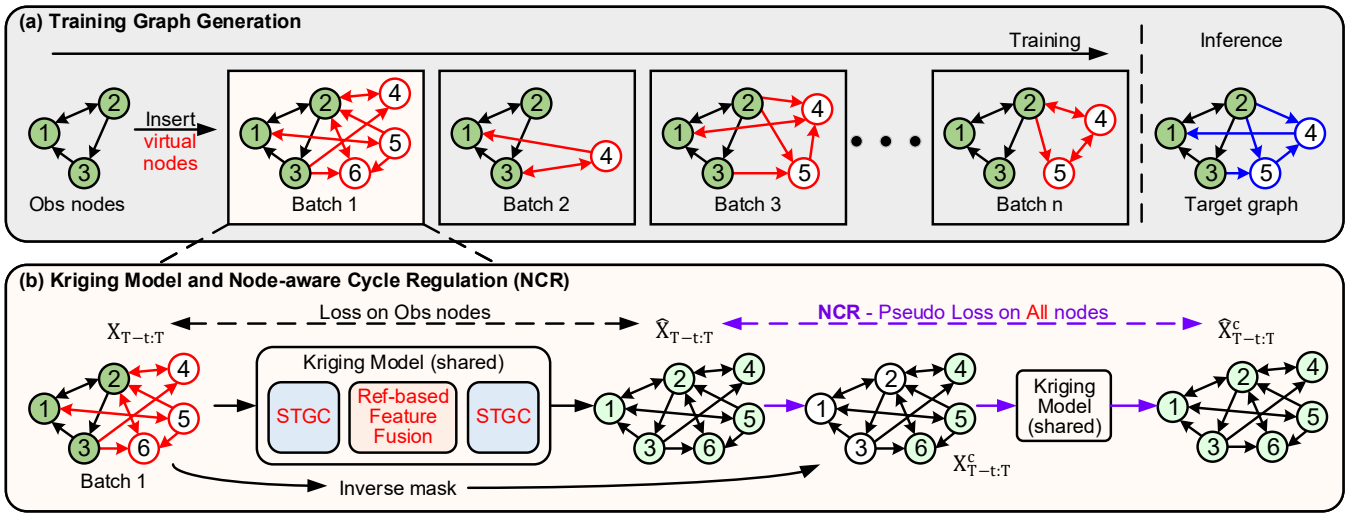

Figure 2: Overview of KITS. (a) Illustration of the procedure of generating multiple training graphs by inserting virtual nodes with randomness (so as to cover different possible inference graphs); (b) Illustration of the kriging model and the Node-aware Cycle Regulation (NCR) (based on Batch 1).

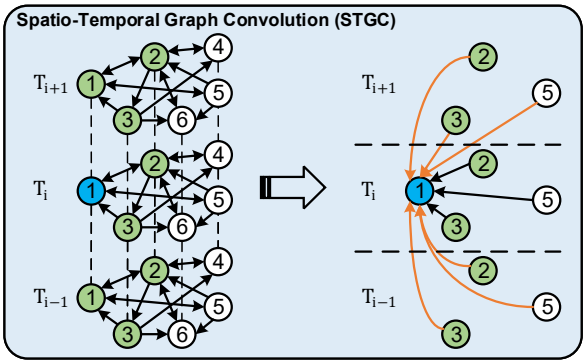

Figure 3: Details of Spatio-Temporal Graph Convolution (STGC). Take the data from three time intervals, and node-1 as an example, its neighbors' information ($T_{i-1:i+1}$) would be propagated to node-1 (T_i) for features aggregation.

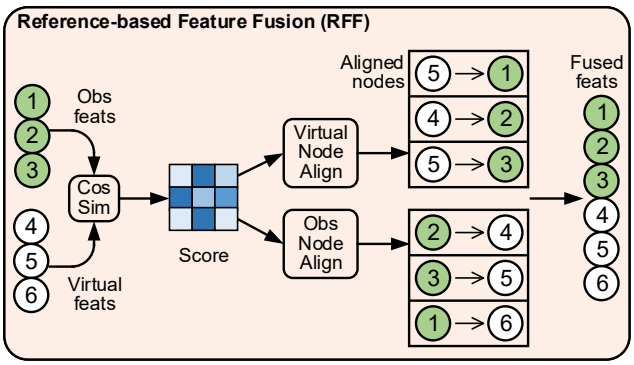

Figure 4: Details of Reference-based Feature Fusion (RFF) for training. In the inference phase, the target unobserved nodes/features would take the role of virtual nodes/features.

$N = N_o + N_v$ represents the total number of observed and virtual nodes, and D is feature dimension. We have the following designs in STGC. First, to aggregate the features across different time intervals, for features $\mathbf{Z}_i$, we directly concatenate it with the features in the previous and following m time intervals and denote the concatenated features as $\mathbf{Z}_{i-m:i+m} \in \mathbb{R}^{N\times(2m+1)D}$. Second, we aggregate the features across different nodes based on the training graph (indicated by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$). Yet we prevent from aggregating features for a node from itself by masking the diagonal elements of adjacency matrix $\mathbf{A}$ (i.e., we remove the self-loops in the graph), denoted as $\mathbf{A}^-$. The rationale is that in spatio-temporal kriging, observed nodes have values in all time intervals, while virtual nodes have values missing in all time intervals. As a result, the observed/virtual nodes would have superior/inferior features all the time. In the case we allow each node to aggregate features from itself, the observed/virtual nodes would learn to improve their superior/inferior features with superior/inferior features, and thus the gap between their features quality would be widened, which would aggravate the overfitting/underfitting issues of observed/virtual nodes as mentioned in Introduction. An illustration of STGC is shown in Figure 3. Formally, STGC can be written as:

$$\mathbf{Z}_i^{(l+1)} = FC(GC(\mathbf{Z}_{i-m:i+m}^{(l)}, \mathbf{A}^-)) \quad (1)$$

where (l) and $(l+1)$ represent the layer indices, $FC(\cdot)$ is a fully-connected layer, and $GC(\cdot)$ is an inductive graph convolution layer (Cini, Marisca, and Alippi 2021).

Reference-based Feature Fusion (RFF). As mentioned earlier, there exists a quality gap between observed nodes' and the virtual nodes' features. To deal with the gap, we propose a RFF module, which pairs observed nodes and virtual

Method	Traffic Speed									Traffic Flow		
	METR-LA (207)			PEMS-BAY (325)			SEA-LOOP (323)			PEMS07 (883)		
	MAE	MAPE	MRE	MAE	MAPE	MRE	MAE	MAPE	MRE	MAE	MAPE	MRE
Mean	8.6687	0.2534	0.1515	4.7068	0.1135	0.0752	6.0264	0.1794	0.1045	103.5164	0.9802	0.3380
OKriging	8.1886	0.2391	0.1431	4.7006	0.1131	0.0751	-	-	-	-	-	-
KNN	8.3068	0.2274	0.1452	4.4898	0.1000	0.0718	6.5510	0.1842	0.1136	109.4717	0.9425	0.3575
KCN	7.5972	0.2341	0.1326	5.7177	0.1404	0.0914	-	-	-	-	-	-
IGNKK	6.8571	0.2050	0.1197	3.7919	0.0852	0.0606	5.1865	0.1370	0.0901	80.7719	0.9314	0.2635
LSJSTN	7.0666	0.2066	0.1234	3.8609	0.0836	0.0617	-	-	-	101.7706	0.8500	0.3325
INCREASE	6.9096	0.1941	0.1206	3.8870	0.0835	0.0621	4.8537	0.1267	0.0842	93.7737	1.1683	0.3062
KITS (Ours)	**6.1276**	**0.1714**	**0.1071**	**3.5911**	**0.0819**	**0.0574**	**4.2313**	**0.1141**	**0.0734**	**75.1927**	**0.6847**	**0.2456**
Improvements	**10.64%**	**11.70%**	**10.53%**	**5.30%**	**1.92%**	**5.28%**	**12.82%**	**9.94%**	**12.83%**	**6.91%**	**19.45%**	**6.79%**

Method	Air Quality						Solar Power					
	AQI-36 (36)			AQI (437)			NREL-AL (137)			NREL-MD (80)		
	MAE	MAPE	MRE	MAE	MAPE	MRE	MAE	MAPE	MRE	MAE	MAPE	MRE
Mean	20.9346	0.6298	0.2905	39.0240	1.4258	0.5973	4.5494	1.6164	0.3664	12.2784	3.9319	0.6927
KNN	18.2021	0.5130	0.2526	23.1718	0.7376	0.3547	4.4118	1.2381	0.3554	12.5239	3.3277	0.7066
KCN	20.6381	0.6190	0.2896	21.9771	0.6207	0.3278	4.6349	2.0685	0.3733	11.5863	4.5994	0.6537
IGNKK	22.3862	0.7892	0.3141	22.3997	0.7200	0.3341	2.1939	1.7267	0.1767	4.3950	2.4813	0.2480
LSJSTN	22.7715	0.9022	0.3209	20.1396	0.5314	0.3003	2.0827	1.0960	0.1677	4.3206	1.9160	0.2436
INCREASE	22.9031	1.0682	0.3214	19.9140	0.6130	0.2970	2.0936	1.1342	0.1686	4.7302	1.8029	0.2669
KITS (Ours)	**16.5892**	**0.3873**	**0.2302**	**16.2632**	**0.4187**	**0.2489**	**1.8315**	**0.7812**	**0.1475**	**4.1181**	**1.2523**	**0.2323**
Improvements	**8.86%**	**24.50%**	**8.87%**	**18.33%**	**21.21%**	**16.20%**	**12.06%**	**28.72%**	**12.05%**	**4.69%**	**30.54%**	**4.64%**

Table 1: Comparisons with **inductive** kriging baselines. "-" means some methods require the input of GPS coordinates or full distance information, which is not available in SEA-LOOP and PEMS07 datasets. The best results are shown in **bold**, and the second best are underlined. "Improvements" show the improvement of our KITS over the best baseline.

nodes and then fuses their features. The rationale is that for a virtual node, its inferior features would be improved with the superior features of its paired observed node - this would help to mitigate the underfitting issue of the virtual node; and for an observed node, its superior features would be affected by the inferior features of its paired virtual node - this would help to mitigate the overfitting issue of the observed node.

The RFF module, shown in Figure 4, works as follows. First, we calculate a similarity matrix $\mathbf{M}_s \in [0,1]^{N_o \times N_v}$, where $\mathbf{M}_{s,[i,j]}$ is the re-scaled cosine similarity between the i^{th} observed node and the j^{th} virtual node based on their features. Second, we apply the $\arg\max$ operation to each row/column of $\mathbf{M}_s$ to obtain an index vector $\mathbf{Ind}^*$ and an similarity vector $\mathbf{S}^*$, which indicates the most similar observed/virtual node of each virtual/observed node. Third, we pair each observed/virtual node to its most similar virtual/observed node based on the index vector $\mathbf{Ind}^*$. Fourth, we fuse the features of the nodes that are paired with each other with a shared FC layer and re-scale them based on the similarity vector $\mathbf{S}^*$. The procedure (for the i^{th} virtual node) is given as:

$$\mathbf{Z}_i^v = FC(\mathbf{Z}_i^v || \mathbf{S}_i^* \odot Align(\mathcal{N}^o, \mathbf{Ind}_i^*)) \tag{2}$$

where $Align(\cdot)$ extracts the features of the most similar observed node according to its index $\mathbf{Ind}_i^*$, $\odot$ is the element-wise matrix multiplication, and $\mathcal{N}^o$ denotes the set of observed nodes. Some evidences, showing the effectiveness of RFF for the fitting issue, are provided in Appendix.

Node-aware Cycle Regulation (NCR)

Recall that virtual nodes do not have supervision signals during training. Thus, we propose to construct pseudo labels for better regulating the learning on virtual nodes. Specifically, we propose NCR (as illustrated in Figure 2(b)) as follows. We first conduct the kriging process once (first stage) and obtain the estimated values of all nodes. We then swap the roles of observed nodes and virtual nodes with an *inverse mask* and conduct the kriging process again (second stage) with the estimated values (outputted by the first stage) as *pseudo labels*. The key intuition is that during the second stage of the kriging process, the virtual nodes would have supervision signals (i.e., pseudo labels) for regulation. We note that similar cycle regulation techniques have been used for traffic imputation tasks (Xu et al. 2022), and our NCR differs from existing techniques in that it uses an *inverse mask* but not a *random mask*, i.e., it is node aware and more suitable for the kriging problem. NCR can be written as:

$$\hat{\mathbf{X}}_{T-t:T} = KM(X_{T-t:T}, \mathbf{A}^-) \tag{3}$$

$$\mathbf{X}^c_{T-t:T} = (\mathbf{1} - \mathbf{M}_{T-t:T}) \odot \hat{\mathbf{X}}_{T-t:T} \tag{4}$$

$$\hat{\mathbf{X}}^c_{T-t:T} = KM(\mathbf{X}^c_{T-t:T}, \mathbf{A}^-) \tag{5}$$

where $\mathbf{X}_{T-t:T}$ is the input data, $KM(\cdot)$ is the kriging model, $\hat{\mathbf{X}}_{T-t:T}$ is the output of the kriging model (first stage), $(\mathbf{1} - \mathbf{M}_{T-t:T})$ is the inverse mask, and $\hat{\mathbf{X}}^c_{T-t:T}$ is the output of the kriging model (second stage). Finally, the overall loss function could be written as:

$$\begin{aligned}\mathcal{L} = &MAE(\hat{\mathbf{X}}_{T-t:T}, \mathbf{X}_{T-t:T}, \mathbf{I}_{obs}) + \\ &\lambda \cdot MAE(\hat{\mathbf{X}}^c_{T-t:T}, \hat{\mathbf{X}}_{T-t:T}, \mathbf{I}_{all})\end{aligned} \tag{6}$$

where $MAE(\cdot)$ is mean absolute error, $\mathbf{I}_{obs}$ and $\mathbf{I}_{all}$ mean calculating losses on observed and all nodes, and λ is a hyperparameter controlling the importance of pseudo labels.

Method	Traffic Speed									Traffic Flow		
	METR-LA (207)			PEMS-BAY (325)			SEA-LOOP (323)			PEMS07 (883)		
	MAE	MAPE	MRE	MAE	MAPE	MRE	MAE	MAPE	MRE	MAE	MAPE	MRE
GLTL	8.6372	0.2627	0.1508	4.6986	0.1128	0.0741	-	-	-	-	-	
MPGRU	6.9793	0.2223	0.1220	3.8799	0.0951	0.0620	4.6203	0.1393	0.0802	97.2821	1.1475	0.3177
GRIN	6.6096	0.1959	0.1155	3.8322	0.0845	0.0613	4.2466	0.1262	0.0743	95.9157	0.6844	0.3132
KITS (Ours)	**6.0604**	**0.1708**	**0.1059**	**3.5809**	**0.0788**	**0.0572**	**4.1773**	**0.1132**	**0.0725**	**76.1451**	**0.6673**	**0.2487**
Improvements	**8.31%**	**12.81%**	**8.31%**	**6.56%**	**6.75%**	**6.69%**	**1.63%**	**10.30%**	**2.42%**	**20.61%**	**2.50%**	**20.59%**

Method	Air Quality						Solar Power					
	AQI-36 (36)			AQI (437)			NREL-AL (137)			NREL-MD (80)		
	MAE	MAPE	MRE	MAE	MAPE	MRE	MAE	MAPE	MRE	MAE	MAPE	MRE
GLTL	21.8970	0.6643	0.3073	30.9248	1.0133	0.4612	4.8230	1.5635	0.3885	12.4229	3.6326	0.7009
MPGRU	22.5312	0.9439	0.3126	22.7233	0.8289	0.3478	2.4439	1.8900	0.1968	5.3504	3.7190	0.3018
GRIN	16.7497	0.5235	0.2324	17.4716	0.5615	0.2674	1.9623	1.2376	0.1581	5.0114	2.5790	0.2827
KITS (Ours)	**16.3307**	**0.3559**	**0.2266**	**16.2431**	**0.4097**	**0.2486**	**1.8090**	**0.6661**	**0.1465**	**4.1088**	**1.4056**	**0.2318**
Improvements	**2.50%**	**32.02%**	**2.50%**	**7.03%**	**27.03%**	**7.03%**	**7.81%**	**46.18%**	**7.34%**	**18.01%**	**45.50%**	**18.00%**

Table 2: Comparisons with **transductive** kriging baselines.

Experiments

Experimental Settings

Datasets. We employ 8 public datasets and conduct extensive experiments on them, so as to validate the effectiveness of KITS. These datasets are collected from different real-world application scenarios, including 4 datasets in the field of traffic (METR-LA, PEMS-BAY, SEA-LOOP, PEMS07), 2 in air quality (AQI-36, AQI), and 2 in solar power (NREL-AL, NREL-MD). More details about these datasets, including basic statistics of each dataset, detailed description, data pre-processing techniques, and the construction of adjancency matrices, are explained in Appendix.

Baselines. Apart from the existing inductive kriging baselines, we also include some transductive kriging baselines for further comparisons. Note that transductive kriging is a relatively easier setting than the inductive one, and their differences are explained in Related Work. The selected kriging baselines include: (1) Inductive kriging: Mean imputation, OKriging (Cressie and Wikle 2015), K-nearest neighbors (KNN), KCN (Appleby, Liu, and Liu 2020), IGNNK (Wu et al. 2021a), LSJSTN (Hu et al. 2021) and INCREASE (Zheng et al. 2023); (2) Transductive kriging: GLTL (Bahadori, Yu, and Liu 2014), MPGRU (Cini, Marisca, and Alippi 2021) and GRIN (Cini, Marisca, and Alippi 2021). More details are provided in Appendix.

Evaluation metrics. We mainly adopt Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Relative Error (MRE) (Cini, Marisca, and Alippi 2021) as the evaluation metrics. In some experiments, we additionally include some results of Root Mean Square Error (RMSE) and R-Square (R2) for better evaluation. The details of each evaluation metric are explained in Appendix.

Main Results

Apart from the inductive setting that we target in this paper, we consider the transductive setting to cover a broad range of settings of kriging. The difference between these two settings is that in the former, the unobserved nodes are not available for training while in the latter, they are available. Our KITS can be applied in the transductive setting by replacing the virtual nodes with actual unobserved nodes. We set the missing ratios $\alpha = 50\%$ for all datasets and present the results in Table 1 for inductive setting, and in Table 2 for transductive setting. Besides, we include some additional results evaluated by RMSE and R2, and the error bars evaluation in Appendix.

Inductive kriging comparisons. Table 1 shows that our KITS consistently achieves state-of-the-art performance, e.g., KITS outperforms the best baseline by **18.33%** on MAE (AQI dataset) and **30.54%** on MAPE (NREL-MD). We attribute it to the fact that the proposed KITS has taken aforementioned *graph gap* and *fitting* issues into consideration: (1) we apply the novel Increment training strategy for kriging to benefit from diverse and dense training graphs. (2) utilizes a well-designed STGC module and another RFF module to improve the feature quality of virtual (unobserved) nodes; (3) further presents an NCR to create reliable pseudo labels for kriging.

Transductive kriging comparisons. As mentioned before, we also test our method in a transductive setting, where the target nodes and the full graph structure are **known** during the training phase, which is easier than the inductive setting that this paper focuses on. Both settings have their application scenarios in practice. Table 2 shows similar conclusions: KITS improves MAE by **20.61%** on PEMS07 dataset and MAPE by up to **45.50%** on NREL-MD dataset. Even with unobserved node topology revealed in training (no more graph gap), the transductive methods still cannot beat KITS, since STGC, RFF, NCR modules better fuse features and offer pseudo supervision.

Ablation Study

Different ways of node insertion. The first row (Random) of Table 3 means the newly-inserted virtual nodes are randomly connected to known nodes (observed nodes and in-

Method	MAE	MAPE	MRE
w/ all nodes			
Random	6.4553	0.1886	0.1128
w/ first-order neighbors			
p=1	6.4254	0.1817	0.1123
p=0.75	6.4397	0.1863	0.1125
p=0.5	6.4094	0.1854	0.1120
p=0.25	6.4670	0.1860	0.1130
p=0	6.8282	0.2089	0.1193
p=random	**6.3754**	**0.1806**	**0.1114**

Table 3: Different ways of node insertion on METR-LA.

Method	INC	NCR	STGC	RFF	MAE	MAPE	MRE
M-0					6.7597	0.1949	0.1181
M-1	✓				6.3754	0.1806	0.1114
M-2	✓	✓			6.2607	0.1791	0.1094
M-3	✓		✓		6.2107	0.1738	0.1117
M-4	✓			✓	6.3269	0.1810	0.1106
M-5	✓	✓	✓	✓	**6.1276**	**0.1714**	**0.1071**

Table 4: Component-wise ablation study. Column "INC" means whether Increment training strategy (✓) is used.

serted virtual nodes). In this case, the virtual node might connect to distant nodes. Several existing studies (Gilmer et al. 2017; Li, Cai, and He 2017; Ishiguro, Maeda, and Koyama 2019; Pham et al. 2017) have adopted this strategy for graph augmentation. Other rows randomly connect a virtual node to a known node and a fraction p of its first-order neighbors (this is the strategy adopted in this paper). According to the results, (1) our strategy of creating edges between virtual nodes and a chosen node's neighbors works better than the commonly used strategy of creating edges based on all nodes, and (2) it works better to use a random p since it would generate more diverse training graphs.

Ablation study on different modules. Table 4 validates the effectiveness of each proposed module. (1) We first compare between M-0 (Decrement training strategy) and M-1 (Increment training strategy), which share the same GCN model (Cini, Marisca, and Alippi 2021). The results show that Increment training strategy outperforms Decrement training strategy by a large margin (e.g., **5.69%** on MAE). (2) We then compare M-2, M-3, and M-4 with M-1. The results verify the benefit of each of the NCR, STGC and RFF modules. (3) The full model achieves the lowest MAE 6.1276, which is **9.37%** lower than that of M-0, which demonstrates the effectiveness of our design.

Training strategies with different missing ratios. To fairly compare two training strategies' behaviors under different missing ratios, we create a decrement version of our model: we change model M-5 in Table 4 with standard Decrement training strategy (Wu et al. 2021a) and the rest modules remain untouched. We use METR-LA and vary missing ratios α from 50% to 80%. (1) Red v.s. Green in Figure 5: With the increase of α, increment one (red)'s advantage margin becomes larger over the decrement one (green). Since the

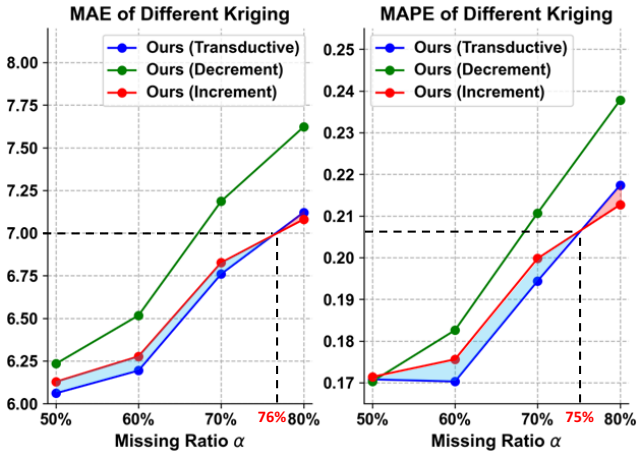

Figure 5: Comparisons between Decrement and Increment training strategies.

graph gap issue becomes severer, the Decrement training strategy's performance deteriorates faster. (2) **Virtual nodes have similar positive impacts as the real nodes** (Red v.s. Blue in Figure 5): We conduct a transductive version of KITS, which replaces the virtual nodes with real unobserved nodes (without values) for comparison. With all α, the increment strategy could achieve similar results to the transductive setting, demonstrating that the created training graphs could achieve similar improvement as the real full graphs with unobserved nodes. Before α hits 76%, the difference of our virtual graph (red line, increment inductive) and the real graph (blue line, transductive) is highlighted with the light-blue region, which is only around 1.10% MAE; when α is larger than 76%, our virtual graph can offer even better performance than the real graph, highlighted in the light-red region on the most right side of each subplot.

Conclusion

In this paper, we study the inductive spatio-temporal kriging problem. We first show that existing methods mainly adopt the Decrement training strategy, which would cause a gap between training and inference graphs (called the *graph gap* issue). To mitigate the issue, we propose a new *Increment training strategy*, which inserts *virtual nodes* in the training graph to mimic the unobserved nodes in the inference graph, so that the gap between the two graphs would be naturally reduced. We further design two modules, namely Reference-base Feature Fusion (RFF) and Node-aware Cycle Regulation (NCR), for addressing the fitting issues caused by the lack of labels for virtual nodes. We finally conduct extensive experiments on eight datasets, which consistently demonstrate the superiority of our proposed model.

Acknowledgments

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

Appleby, G.; Liu, L.; and Liu, L.-P. 2020. Kriging convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3187–3194.

Bahadori, M. T.; Yu, Q. R.; and Liu, Y. 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. *Advances in neural information processing systems*, 27.

Barthélemy, M. 2011. Spatial networks. *Physics reports*, 499(1-3): 1–101.

Cascante-Bonilla, P.; Tan, F.; Qi, Y.; and Ordonez, V. 2021. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6912–6920.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Cini, A.; Marisca, I.; and Alippi, C. 2021. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. *arXiv preprint arXiv:2108.00298*.

Cressie, N.; and Wikle, C. K. 2015. *Statistics for spatio-temporal data*. John Wiley & Sons.

Deng, L.; Liu, X.-Y.; Zheng, H.; Feng, X.; and Chen, Y. 2021. Graph spectral regularized tensor completion for traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems*, 23(8): 10996–11010.

Gad, I.; Manjunatha, B.; et al. 2017. Performance evaluation of predictive models for missing data imputation in weather data. In *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1327–1334. IEEE.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272. PMLR.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Goovaerts, P. 1998. Ordinary cokriging revisited. *Mathematical Geology*, 30: 21–42.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Han, T.; Bai, L.; Gao, J.; Wang, Q.; and Ouyang, W. 2022. Dr. vic: Decomposition and reasoning for video individual counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3083–3092.

Hu, J.; Liang, Y.; Fan, Z.; Yin, Y.; Zhang, Y.; and Zimmermann, R. 2021. Decoupling Long-and Short-Term Patterns in Spatiotemporal Inference. *arXiv preprint arXiv:2109.09506*.

Ishiguro, K.; Maeda, S.-i.; and Koyama, M. 2019. Graph warp module: an auxiliary module for boosting the power of graph neural networks in molecular graph analysis. *arXiv preprint arXiv:1902.01020*.

Krige, D. G. 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6): 119–139.

Lai, Z.; Zhang, D.; Li, H.; Zhang, D.; Lu, H.; and Jensen, C. S. 2024. ReCTSi: Resource-efficient Correlated Time Series Imputation via Decoupled Pattern Learning and Completeness-aware Attentions. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1474–1483.

Lei, M.; Labbe, A.; Wu, Y.; and Sun, L. 2022. Bayesian kernelized matrix factorization for spatiotemporal traffic data imputation and kriging. *IEEE Transactions on Intelligent Transportation Systems*, 23(10): 18962–18974.

Li, J.; Cai, D.; and He, X. 2017. Learning graph-level representation for drug discovery. *arXiv preprint arXiv:1709.03741*.

Li, W.; Tsung, F.; Song, Z.; Zhang, K.; and Xiang, D. 2021. Multi-sensor based landslide monitoring via transfer learning. *Journal of Quality Technology*, 53(5): 474–487.

Li, Z. 2021. Tensor topic models with graphs and applications on individualized travel patterns. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2756–2761. IEEE.

Li, Z.; Sergin, N. D.; Yan, H.; Zhang, C.; and Tsung, F. 2020a. Tensor completion for weakly-dependent data on graph for metro passenger flow prediction. In *proceedings of the AAAI conference on artificial intelligence*, volume 34, 4804–4810.

Li, Z.; Yan, H.; Zhang, C.; and Tsung, F. 2020b. Long-short term spatiotemporal tensor prediction for passenger flow profile. *IEEE Robotics and Automation Letters*, 5(4): 5010–5017.

Li, Z.; Yan, H.; Zhang, C.; and Tsung, F. 2022. Individualized Passenger Travel Pattern Multi-clustering based on Graph Regularized Tensor Latent Dirichlet Allocation. *Data Mining and Knowledge Discovery*, 36(4): 1247–1278.

Liang, Y.; Ouyang, K.; Jing, L.; Ruan, S.; Liu, Y.; Zhang, J.; Rosenblum, D. S.; and Zheng, Y. 2019. Urbanfm: Inferring fine-grained urban flows. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 3132–3142.

Lin, Z.; Zhang, G.; He, Z.; Feng, J.; Wu, W.; and Li, Y. 2021. Vehicle Trajectory Recovery on Road Network Based on Traffic Camera Video Data. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, 389–398.

Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2024a. TimeCMA: Towards LLM-Empowered Time Series Forecasting via Cross-Modality Alignment. *arXiv preprint arXiv:2406.01638*.

Liu, C.; Yang, S.; Xu, Q.; Li, Z.; Long, C.; Li, Z.; and Zhao, R. 2024b. Spatial-temporal large language model for traffic prediction. *arXiv preprint arXiv:2401.10134*.

Liu, K.; Ruan, S.; Xu, Q.; Long, C.; Xiao, N.; Hu, N.; Yu, L.; and Pan, S. J. 2022a. Modeling trajectories with multi-task learning. In *2022 23rd IEEE International Conference on Mobile Data Management (MDM)*, 208–213. IEEE.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022b. SCINet: time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, Z.; Miao, H.; Zhao, Y.; Liu, C.; Zheng, K.; and Li, H. 2024c. LightTR: A Lightweight Framework for Federated Trajectory Recovery. *arXiv preprint arXiv:2405.03409*.

Mao, Z.; Li, Z.; Li, D.; Bai, L.; and Zhao, R. 2022. Jointly contrastive representation learning on road network and trajectory. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 1501–1510.

Miao, H.; Zhao, Y.; Guo, C.; Yang, B.; Zheng, K.; Huang, F.; Xie, J.; and Jensen, C. S. 2024. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. *arXiv preprint arXiv:2404.14999*.

Pham, T.; Tran, T.; Dam, H.; and Venkatesh, S. 2017. Graph classification via deep learning with virtual nodes. *arXiv preprint arXiv:1708.04357*.

Takeuchi, K.; Kashima, H.; and Ueda, N. 2017. Autoregressive tensor factorization for spatio-temporal predictions. In *2017 IEEE international conference on data mining (ICDM)*, 1105–1110. IEEE.

Tonkin, M. J.; and Larson, S. P. 2002. Kriging water levels with a regional-linear and point-logarithmic drift. *Groundwater*, 40(2): 185–193.

Wu, Y.; Zhuang, D.; Labbe, A.; and Sun, L. 2021a. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4478–4485.

Wu, Y.; Zhuang, D.; Lei, M.; Labbe, A.; and Sun, L. 2021b. Spatial Aggregation and Temporal Convolution Networks for Real-time Kriging. *arXiv preprint arXiv:2109.12144*.

Xu, D.; Wei, C.; Peng, P.; Xuan, Q.; and Guo, H. 2020. GE-GAN: A novel deep learning framework for road traffic state estimation. *Transportation Research Part C: Emerging Technologies*, 117: 102635.

Xu, Q.; Lin, G.; Loy, C. C.; Long, C.; Li, Z.; and Zhao, R. 2025. Eliminating feature ambiguity for few-shot segmentation. In *European Conference on Computer Vision*, 416–433. Springer.

Xu, Q.; Liu, X.; Zhu, L.; Lin, G.; Long, C.; Li, Z.; and Zhao, R. 2024a. Hybrid mamba for few-shot segmentation. *arXiv preprint arXiv:2409.19613*.

Xu, Q.; Long, C.; Yu, L.; and Zhang, C. 2023a. Road extraction with satellite images and partial road maps. *IEEE Transactions on Geoscience and Remote Sensing*, 61: 1–14.

Xu, Q.; Ruan, S.; Long, C.; Yu, L.; and Zhang, C. 2022. Traffic Speed Imputation with Spatio-Temporal Attentions and Cycle-Perceptual Training. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2280–2289.

Xu, Q.; Zhao, W.; Lin, G.; and Long, C. 2023b. Self-calibrated cross attention network for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 655–665.

Xu, R.; Miao, H.; Wang, S.; Yu, P. S.; and Wang, J. 2024b. PeFAD: a parameter-efficient federated framework for time series anomaly detection. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 3621–3632.

Yan, S.; Xu, D.; Zhang, B.; Zhang, H.-J.; Yang, Q.; and Lin, S. 2006. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE transactions on pattern analysis and machine intelligence*, 29(1): 40–51.

Zhang, Y.; Li, A.; Li, J.; Han, D.; Li, T.; Zhang, R.; and Zhang, Y. 2022. SpecKriging: GNN-Based Secure Cooperative Spectrum Sensing. *IEEE Transactions on Wireless Communications*, 21(11): 9936–9946.

Zheng, C.; Fan, X.; Wang, C.; Qi, J.; Chen, C.; and Chen, L. 2023. INCREASE: Inductive Graph Representation Learning for Spatio-Temporal Kriging. *arXiv preprint arXiv:2302.02738*.

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, T.; Shan, H.; Banerjee, A.; and Sapiro, G. 2012. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *Proceedings of the 2012 SIAM international Conference on Data mining*, 403–414. SIAM.