Non-Neighbors Also Matter to Kriging: A New Contrastive-Prototypical Learning

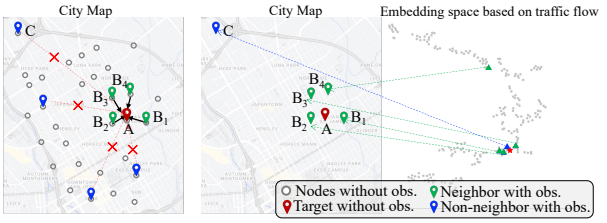
Zhishuai Li[1], Yunhao Nie[1,2], Ziyue Li[3], Lei Bai[4] , Yisheng Lv[2], Rui Zhao[1]
[1]SenseTime Research, [2]Chinese Academy of Sciences, [3]University of Cologne, [4]Shanghai AI Laboratory

Abstract

Kriging aims to estimate the attributes of unseen geo-locations from observations in the spatial vicinity or physical connections. Existing works assume that neighbors' information offers the basis for estimating the unobserved target while ignoring non-neighbors. However, neighbors could also be quite different or even misleading, and the non-neighbors could still offer constructive information. To this end, we propose "**C**ontrastive-**P**rototypical" self-supervised learning for **K**riging (KCP): (1) The neighboring contrastive module coarsely pushes neighbors together and non-neighbors apart. (2) In parallel, the prototypical module identifies similar representations via exchanged prediction, such that it refines the misleading neighbors and recycles the useful non-neighbors from the neighboring contrast component. As a result, *not all* the neighbors and *some* of the non-neighbors will be used to infer the target. (3) To learn general and robust representations, we design an adaptive augmentation module that encourages data diversity. Theoretical bound is derived for the proposed augmentation. Extensive experiments on real-world datasets demonstrate the superior performance of KCP compared to its peers with **6%** improvements and exceptional transferability and robustness.

1 INTRODUCTION

Spatially-distributed sensors are commonly deployed to perceive the environment, such as temperature-humidity sensors in weather stations in meteorology (Gad et al., 2017), landside tilt sensors in geology (Li et al., 2021), water-level sensors in hydrogeology (Tonkin and Larson, 2002), geomagnetic sensors (Kwong et al., 2009) or cameras (Lin et al., 2021; Han et al., 2022a) in transportation, and so on.



(a) Existing Kriging using neighbors' feature

(b) Non-neighbor embeddings can be even closer (C) and neighbor embedding can be far (B_4)

Figure 1: When looking through the node representations in the embedding space, spatial neighbors do not always appear nearby and non-neighbors may also be close to the target.

However, it is prohibitively costly to deploy sensors with high coverage rates, resulting in under-sampling and skewed monitoring. For example, in the traffic domain, some cities only have less than 20% intersections installed with the sensors to collect the traffic flow. To leverage data-driven applications, the data collected from spatially sporadic sensors must be expanded into spatially fine-grained data. Tailored to such a spatial super-resolution task, *Kriging*, also known as *spatial interpolation*, infers the attributes of targets at unsampled locations from observations in the spatial vicinity or physical connections. Shown in Fig. 1(a), Kriging exploits a few locations' observed data (blue/green pins) to infer the rest that is not even equipped with sensors (grey circles).

"Common Practice" of Kriging: Neighbors only. Existing Kriging models (Bostan, 2017; Appleby et al., 2020; Wu et al., 2021a,b; Lei et al., 2022) are built upon the assumption that the information of neighbors offers the basis to estimate the attribute values of the target. As shown in Fig. 1(a), closer neighbor nodes (e.g., green nodes B_1 to B_4) have larger impacts on the interpolation weight of the target node (e.g., the red

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

node A), but the non-neighbors (e.g., the blue nodes) should be ignored, called "law of neighborhood". To achieve so, statistical Kriging introduces graph Laplacian penalty (Rao et al., 2015), and deep Kriging based on Graph Convolution Networks (GCN) (Wu et al., 2021a; Appleby et al., 2020) also relies on adjacency matrix, pushing neighbors' embeddings closer.

A New Perspective: Solely relying on the neighborhood is not always error-proof. Representation speaks louder than neighbors. Here we take a closer look at Fig. 1(b): we conduct graph encoding via GCN (Kipf and Welling, 2016) based on node attribute - traffic flow, and visualize the embeddings via t-SNE (Van der Maaten and Hinton, 2008), with the target A as red ⋆, its neighbors B_1 to B_4 as green △, and its non-neighbor, e.g., C, as blue △. The insights are two-fold: (1) the red ⋆ is close to the most of green △: it aligns with the general assumption - neighbors have similar representations in embedding space. (2) Some non-neighbors' embeddings can be even closer despite not being physical neighbors, e.g., the blue △ is even closer; some neighbor, i.e., B_4 yet has quite a different embedding. Inspired by the context, we rethink Kriging from a novel perspective: **to interpolate by using the embeddings rather than the raw input**. Specifically, we learn the target embedding first and then recover the values of unknown targets via downstream modules.

To conclude, we will formulate the Kriging task in a pre-training and finetuning paradigm. We try to solve the following three research questions (RQ):
RQ1: How to first conform to the *common practice of Kriging*, that is, estimating the target node with its neighbors and improving the aggregation effecicy?
RQ2: On the top of **RQ1**, how to further utilize the non-neighbors' useful information and discard the neighbors' nonconstructive information and maintain the balance of the two (some may use more no-neighbors and others may use more neighbors)?
RQ3: For representation learning, how to learn a more general embedding that respects the spatiotemporal nature of the data, reflects the Kriging task, and is also potentially robust enough to noise?

To answer all the questions, we propose a **K**riging **C**ontrastive-**P**rototypical Learning (KCP). Since selecting which nodes to infer the target is based on representation, high-quality embeddings are now the first key, while the second key is how to select the rational nodes. Self-supervised learning (SSL) has proved its superiority in learning general embedding, so this work will be also the first Kriging solution based on contrastive learning. Three critical modules are designed, shown in Fig. 2: The contrastive module respects the "common practice" by attracting the neighbors' embeddings together and repelling non-neighbors' embeddings away. It can elaborate the embedding of the target node by its neighbors, which answers **RQ1**; the prototypical module instead learns to discard the neighbors' nonconstructive embeddings (*refine*) and keep the non-neighbors' useful embeddings (*recycle*) via exchanged prediction, which responses **RQ2**. To get a robust representation, the third component, a spatiotemporal adaptive augmentation module, is proposed, which augments the input graph from (spatial) topology and (temporal) feature views, in an adaptive and probabilistic manner to diversify the augmented data, answering the **RQ3**.

The main contributions are summarized as follows:

(1) To the best of our knowledge, we are the first to break the common practice of Kriging, i.e., neighbors only, and propose a novel self-supervised learning framework to better aggregate information from *not all* neighbors and *some* non-neighbors for Kriging.

(2) We let the two SSL modules, i.e., neighboring contrast and prototypical head, collaborate to refine and recycle constructive information for target nodes. To facilitate more robust representation learning, we also propose an adaptive augmentation module to generate diverse and Kriging-related data for the SSL modules.

(3) We conduct extensive experiments on three real-world datasets to evaluate the superiority of the proposed KCP under various settings, which achieves $3\% \sim 6\%$ improvements over its peers and demonstrates the best transferability and robustness.

2 RELATED WORK

Kriging methods can be categorized as *transductive* and *inductive*. The details can be referred to in Supplementary Materials 1.

2.1 Inductive Kriging

The message-passing mechanism in graph neural networks (GNNs), such as GraphSAGE (Hamilton et al., 2017), makes them well-suited for inductive Kriging, in that they can effectively aggregate the helpful information to evaluate unknown data points. By predefining and investigating the K nearest neighbors around the unobserved node, KCN (Appleby et al., 2020) estimates the targets by averaging the neighbors' labels with learnable weights. Furthermore, with randomly selected K neighbors, PE-GNN (Klemmer et al., 2023) plugs a general and highly modular positional encoding component to learn the context-aware embedding for geographic coordinates. By the modified Moran's I auxiliary task, the module can be well-trained in parallel with the main task, incorporating spatial context and correlation explicitly. Similarly, Egressy and Wat-

tenhofer (2022) finds that positional node embeddings derived from the coordinate position under the stress function can be very effective for graph-based applications, and can learn to generalize with limited training data. These provide novel insights into graph-based Kriging. IGNNK (Wu et al., 2021a) generates random subgraphs and reconstructs the signals on them (especially on unobserved nodes) by learning the spatial correlations. But it may lack the capture of node-level temporal dependencies. Aiming at better information aggregation from neighbors, SATCN (Wu et al., 2021b) designs several node message-passing modules for graph learning and mines temporal correlations through a temporal convolutional network. INCREASE (Zheng et al., 2023) explicitly defines geographic and semantic neighbors for the target node, thus providing additionally referred candidates. But it may also introduce more false positive neighbors as confused terms. Overall, the key insight of GNNs for Kriging is how to improve the information aggregation efficiency, especifically spatiotemporal correlations from either neighbors or no-neighbors, which relates to the model's robustness and inductive ability.

2.2 Graph structure learning on road network

Pioneering work, such as STGCN (Yu et al., 2018) and DCRNN (Li et al., 2018a), emphasizes the inherent topology of the road network, such as a binary adjacency graph, or utilizes predefined graphs based on specific metrics like Euclidean distance to indicate the graph structure. Tailored to the traffic data input, GWNet (Wu et al., 2019) proposes the learnable embedding metric for pairwise node distance construction, which automatically constructs adaptive graphs for road networks. According to the node embeddings, AGCRN (Bai et al., 2020) introduces node-specific convolution filters to infer the inter-dependencies among different traffic series automatically. Rather than the Euclidean distance metric in road works, MFFB (Li et al., 2020b) proposes evaluating the node distance by Spearman similarity with trainable bias, thus building a dynamic adaptive graph. MTGODE (Jin et al., 2022) abstracts multivariate time series into dynamic graphs with time-evolving node features and unknown graph structures. Based on the formulation, it designs and solves a neural ODE to complement missing graph topologies and unify both spatial and temporal message passing. These studies enhance the feasibility of building adaptive and general graphs but cannot be directly applied to the Kriging task, since we only recognize the basic attributes of unseen nodes and lack any historical observations, while historical data is indispensable in adaptive graph structure learning.

3 METHODOLOGY

3.1 Problem Definition

In the Kriging task, different locations in the studied region can be formulated as a potential graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the nodes and edges sets, respectively. Each geo-location is treated as a node $v \in \mathcal{V}$ and the connections are reflected by the graph adjacent matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$. Thus, it can also be viewed as a graph super-resolution problem: Given limited observed nodes $\mathcal{V}_o$ attributes $\mathbf{X}$ ($\mathbf{x}_i$ for node v_i's attributes), the attributes $\mathbf{y}_j$ of an unobserved node $v_j \in \mathcal{V}_u$ can be inferred from the spatial/temporal correlations across nodes. To perform the inductive Kriging, the topological information of unobserved nodes is unavailable in the training stage. As shown in Fig. 2(a), training is conducted on the subgraph $\mathcal{G}_o \subseteq \mathcal{G}$ composed of N_o observed nodes (with adjacency $\mathbf{A}_o$). In the testing stage, the un-observations serve as the newly added nodes, and there is $\mathbf{A} \in \mathbb{R}^{(N_o+N_u)\times(N_o+N_u)}$, where N_u denote the number of unobserved nodes and $N_o + N_u = |\mathcal{V}|$. In spatiotemporal Kriging, node attributes are usually represented by time series with horizon T, i.e., $\mathbf{x}_i = [x_i^{t+1}, \ldots, x_i^{t+T}]$. The objective is to find a dedicated function $\mathcal{F}$ which can estimate the attributes of N_u unknown nodes $\hat{\mathbf{Y}} \in \mathbb{R}^{N_u\times T}$ ($\mathbf{Y}$ for ground truth) with the available nodes' information $\mathbf{X} \in \mathbb{R}^{N_o\times T}$ and adjacent matrix $\mathbf{A}$. That is $\mathbf{Z} = \mathcal{F}(\mathbf{X}; \mathbf{A}_o)$ (training) then $\hat{\mathbf{Y}} = \mathcal{F}(\mathbf{Z}; \mathbf{A})$ (testing).

3.2 The architecture of KCP

We elaborate on the details of KCP, with the overall framework in Fig. 2. Our model mainly contains three tailored components based on a graph feature extraction module: adaptive data augmentation, neighboring contrast, and prototypical head. The first (Fig. 2(b1)) is dedicated to performing attribute-level and topological augmentations from the canonical view, and then provides the augmented view for feature extraction. The last two components achieve self-supervision according to the extracted representations' consistency between the canonical and augmented views: The former (Fig. 2(b2)) conducts the contrastive learning by exploiting neighbor and non-neighbor information. Considering that not all neighbors are definitely similar, and non-neighbors are not always incompatible (as we argued in Fig. 1), the latter (Fig. 2(b3)) aims to identify similar representations regardless of the adjacency via exchanged prediction, thus refining the positive and recycling the negative from the neighboring contrast.

3.2.1 Graph feature extraction module To encode the node representation, we adopt a GNN backbone with two graph aggregation layers (following GraphSAGE (Hamilton et al., 2017)) for message passing. The rationale lies in 1) it is inductive and thus

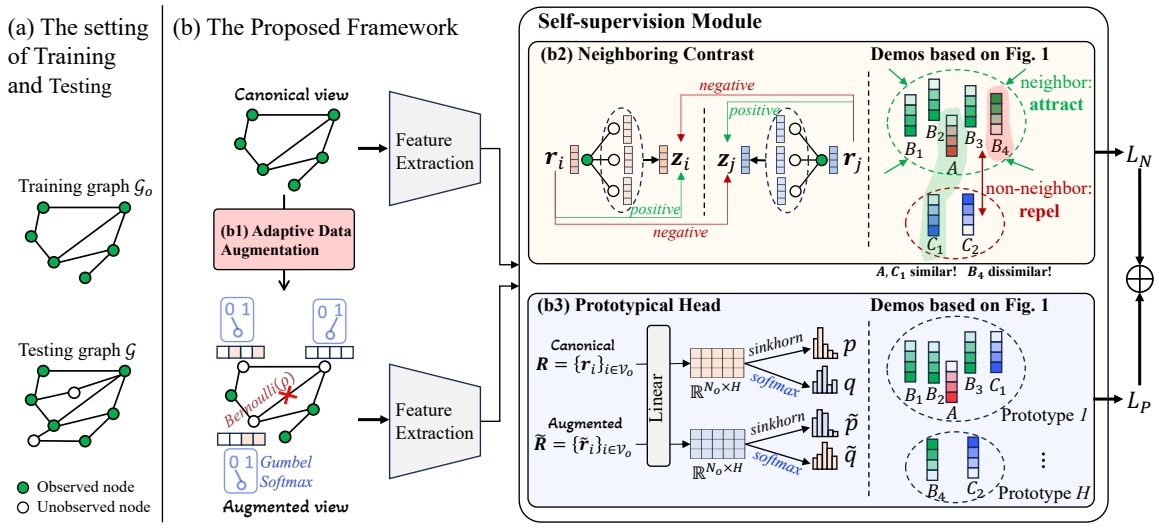


Figure 2: The framework of the proposed KCP

can be easily applied to new-coming nodes even unseen scenarios; 2) it can model the node-attribute projection (temporal patterns in the Kriging task) and the spatial aggregation across nodes by learnable weights and adjacent matrix $\mathbf{A}$, respectively. For each layer, with the attribute $\mathbf{x}_i$ from v_i, the output $\mathbf{x}'_i$ after aggregation:

$$\mathbf{x}'_i = ReLU\left(\mathbf{W}\cdot\left[\mathbf{x}_i, \mathrm{Agg}_{j\in\{j\in\mathcal{V}|\mathbf{A}_{ji}>0\}}\left(\mathbf{W}^t\mathbf{x}_j+\mathbf{b}\right)\right]\right), \tag{1}$$

where $ReLU(\cdot)$ is an activation function, $[,]$ represents concatenation, and $\mathrm{Agg}(\cdot)$ means the aggregation function (we simply use mean in the model). $\mathbf{W}$, $\mathbf{W}^t$, and $\mathbf{b}$ conduct the projection which can simply model the temporal dependencies for the node attributes. After the feature extraction module, the output representation $\mathbf{r}\in\mathbb{R}^E$ for each node in two views is yielded, where E is the dimension of the representation, and there are $\mathbf{R}\in\mathbb{R}^{N_o\times E}$ for all the nodes in the canonical view and $\tilde{\mathbf{R}}\in\mathbb{R}^{N_o\times E}$ in the augmented view.

3.2.2 Adaptive data augmentation Data augmentation is a key mechanism for SSL (especially contrastive learning). For graph SSL, there are two families of techniques based on node attribute and topology. However, as argued in (Zhu et al., 2021), simple data augmentation in attribute/topology domains may not generate diverse contexts. This issue is more prominent in the Kriging since there are no inherent clues in unknown nodes, and the estimations are completely dependent on contextual information. Additionally, noise, such as random missing caused by sensor breakdown in observed nodes, is also notable, which complicates the learning of node representations.

Given the issues, in the training stage, we devise an adaptive data augmentation strategy to corrupt the canonical view, which is illustrated in Fig. S1 in Suppl. Specifically, among the observed nodes, we first randomly select N nodes while keeping the remaining nodes unchanged. Then, attribute-level and topological augmentations are sequentially applied to each selected node (More details in Fig. S1 in Suppl. 3.1).

(1) At the attribute-level augmentation, we allow each sampled node to pick one option between two strategies: 1) feature mask, some attributes on a sampled node are masked under a pre-specified ratio r_m, which aims to analog the noise that appears in the known nodes (mimic random missing). We believe it can force the model to focus on the temporal dependencies in each node and thus benefit the robustness of the model; 2) node mask, the attributes of a sampled node are all filled with zeros. This can be regarded as a special case of feature mask where $r_m = 1$, which acts as the unknown nodes (mimic Kriging).

Since it is hard to specify which one option should be picked for each sampled node, inspired by Zhu et al. (2021), we design a learning-based operator to achieve the specification adaptively. It can be treated as a binary classification (implemented by a 3-layer multilayer perceptron (MLP)) with the node feature $\mathbf{x}_i$ as input and the binary logit π^i as output: $\pi^i = \mathrm{MLP}(\mathbf{x}_i) = [\pi^i_0, \pi^i_1]$. Considering that the specification is non-differentiable, we introduce the Gumbel Softmax (Jang et al., 2016) trick as the differentiable approximation. During the forward propagation:

$$out_i = \arg\max_m\left(\log\left(\pi^i_m\right)+g_m\right), m\in\{0,1\}, \tag{2}$$

where out_i is the output specification for node v_i. $out_i = 0$ means feature mask and $out_i = 1$ denotes node mask. $g_m \sim \mathrm{Gumbel}(0,1)$ is a noise term drawn from the standard Gumbel distribution. The backward propagation (MLP update) is conducted with the

temperature parameter τ by taking the derivative of:

$$out_i = \frac{\exp\left(\left(\log\left(\pi_m^i\right) + g_m\right)/\tau\right)}{\sum_{n=0}^{1}\exp\left(\left(\log\left(\pi_n^i\right) + g_n\right)/\tau\right)}. \tag{3}$$

(2) In topological augmentation, edge drop is performed. Different from traditional random edge dropping, we only drop the edges connected to a high-centrality node with a probability ρ. The underlying prior is that the edge missing does not alter the neighboring information aggregation. Based on the centrality, the edges around a sampled node v_i will be dropped according to Bernoulli distribution, i.e., Bernoulli$(\rho_i), \rho_i = \max((\mathrm{D}_{ii} - d_{\text{avg}})/d_{\max}, 0)$, where D_{ii} is the i,i-th entry for degree matrix D, d_{avg} is the average degree across all the nodes and $d_{\max}$ is the maximum. This setting ensures that the plain nodes are ignored while the edges around critical nodes are augmented: for the node with a few edges, e.g., only one edge, dropping the only edge will cause the node to have no neighbor, thus during the training, no reference may cause bad inference; in contrast, the central nodes have enough edges, dropping its edges will encourage robust inference from different subsets of neighbors.

Theorem 3.1 *Given the canonical graph $\mathcal{G}$ and augmented graph, the homomorphism density t (definations referred to (Han et al., 2022b)) of the augmented graphon W' is determined by Φ, and $W' = (\mathbf{1} - \Phi) \odot W$, where the i,j-th entry of weighted matrix Φ is sampled from* Bernoulli(ρ_i). *The difference in the homomorphism densities of the canonical graphons W and augmented graphons W' is still upper bounded by*

$$\left|t\left(\mathcal{G}, W'\right) - t\left(\mathcal{G}, W\right)\right| \leqslant (1-\lambda)\mathrm{e}\left(\mathcal{G}\right)\|W\|_{\square} \tag{4}$$

where $\lambda = \Pi_{i,j=1}^{N}(1 - \Phi_{ij})$, $\mathrm{e}\left(\mathcal{G}\right)$ is the number of the edges in $\mathcal{G}$, and $\|W\|_{\square}$ means cut norm(Lovász, 2012).

Theorem 3.1 suggests that topological augmentation is a special case of G-Mixup while sampling entries in Bernoulli(ρ) and the augmentation still retains the benefits of G-Mixup (i.e., promising generalization and robustness). The detailed proof is in Suppl. 3.2.

3.2.3 Neighboring Contrast To guide the model to estimate unobserved nodes by encoding neighboring information, we propose to supervise the target nodes with the representations of their neighbors. Rather than node-to-node or node-to-graph contrast, we primarily emphasize node-to-neighbor contrast since the Kriging task should not refer to a specified node nor to the whole graph; instead, the neighbors. For an anchor node v_i, its representation $\mathbf{r}_i$ and its aggregated neighboring representations in the other view constitute a positive pair $(\mathbf{r}_i, \mathbf{z}_i)$, while the neighboring aggregation of other nodes is set as the negative pair of $\mathbf{r}_i$. Since neighbors are not equally important, the aggregation is adopted by an attention-based readout:

$$\mathbf{z}_i = \mathbf{W_2} \cdot \left(\sum\nolimits_{j \in N_k(i)} \alpha_j \mathbf{r}_j\right), \tag{5}$$

where $\alpha_i = {\exp(\mathbf{W_1}\mathbf{r}_i)}/{\sum_{j \in N_k(i)}\exp(\mathbf{W_1}\mathbf{r}_j)}$. $\mathbf{W_1}, \mathbf{W_2}$ are the trainable matrices, and $N_k(i)$ is a set that contains the top-k nearst neighbors of v_i. With the canonical view and the augmented view, we maximize the agreement between node representations by the noise contrastive estimation loss:

$$\begin{aligned}\mathcal{L}_N = -\frac{1}{|\mathcal{V}|}\sum_{i \in \mathcal{V}}\Big[&\log\sigma\left(\mathcal{D}\left(\mathbf{r}_i, \mathbf{z}_i\right)\right) \\ &+ \mathbb{E}_{w \sim \hat{\mathbb{P}}}\left[\log\left(1 - \sigma\left(\mathcal{D}\left(\mathbf{r}_i, \mathbf{z}_w\right)\right)\right)\right]\Big],\end{aligned} \tag{6}$$

where σ is sigmoid function and $\mathbf{z}_w$ is a negative sample. $\hat{\mathbb{P}}$ is the distribution of negative samples. $\mathcal{D}$ measures the agreement between two vectors, i.e., their cosine similarity $\mathcal{D}(\mathbf{x}, \mathbf{y}) = {\mathbf{x}\cdot\mathbf{y}}/{|\mathbf{x}||\mathbf{y}|}$. As a result, in Fig. 2(b2), target A will be attracted together with its neighbors B_1-B_4, and repelled away from its non-neighbors C_1-C_2, even though C_1 is similar to A, not to B_4.

3.2.4 Prototypical Head Although neighboring contrast utilizes neighbor and non-neighbor information to obtain effective representations under the guidance of Kriging's common practice, it is not always rational since non-neighboring nodes may share similarities in certain time series patterns such as change trends, peaks, or slopes. For example, in traffic, similar traffic flow patterns can appear even in distant intersections due to similar functional areas. To play a "refine and recycle" role for the neighboring contrast component, we further introduce a prototypical head (Liu et al., 2021) module for self-supervision.

Assume there are H typical patterns, called prototypes, among all node attributes that can be clustered. The prototypical head aims to uniformly assign prototype labels to each node with subject to $\sum_i q_{ih} = 1$ and $\sum_h q_{ih} = 1$, where q_{ih} is the assignment probability with prototype h to node v_i. The supervisory signal comes from an intuitive principle: the representations from two views for an anchor node should yield similar assignment probabilities to the same prototype. To this end, we project the representation into a new latent space $\mathbb{R}^H$ by a learnable head $\mathbf{H} \in \mathbb{R}^{E \times H}$. Specifically, for node v_i, there is $\mathbf{c}_i = \mathbf{r}_i \cdot \mathbf{H} = [c_{i1}, c_{i2}, ...c_{iH}]$. The score for assigning prototype h to node v_i can be calculated as $p_{ih} = {\exp(c_{ih})}/{\sum_j^H \exp(c_{ij})}$. Then, the assignment problem can be cast as an optimal transport problem and the optimal assignment probability q_{ih} can be computed as the soft labels by the off-the-shelf iterative Sinkhorn algorithm (Cuturi, 2013) under the same input $\mathbf{c}_i$, i.e., $q_{ih} = \text{Sinkhorn}(c_{ih})$. For the prototype vectors from the canonical view and augmented

Table 1: Performance comparison. For deep models, the results are obtained through three independent executions and in format *mean*±*std*. The best results are in **bold** and the second-best are underlined.

Dataset	Metric(↓)	Statistical models				Deep models			
		KNN-IDW	XGBoost	OKriging	GPMF	BGRL	IGNNK	KCN	KCP(ours)
PeMS	MAE(±std)	50.95	61.52	49.03	54.38	45.09±0.311	43.98±0.233	43.59±0.101	**42.29**±0.122
	RMSE	76.33	77.00	**64.86**	70.12	70.94±0.255	67.11±0.145	68.14±0.177	65.37±0.154
	MAPE	36.5%	42.3%	50.1%	37.4%	31.9%	32.6%	31.6%	**29.7%**
NREL	MAE	1.778	1.878	1.977	1.645	1.702±0.019	1.813±0.012	1.571±0.015	**1.569**±0.013
	RMSE	2.713	3.261	2.980	2.451	2.584±0.018	2.588±0.017	2.366±0.011	**2.353**±0.015
	MAPE	33.2%	44.7%	45.8%	42.7%	73.8%	36.4%	33.4%	**32.2%**
Wind	MAE	1.657	1.562	2.107	1.566	1.564±0.115	1.760±0.013	1.568±0.022	**1.556**±0.018
	RMSE	2.339	2.139	2.789	2.154	**2.049**±0.016	2.489±0.017	2.123±0.018	2.096±0.012
	MAPE	39.1%	32.3%	55.3%	37.6%	**30.3%**	55.1%	33.4%	31.6%

view (denoted by $\mathbf{c}_i$ and $\tilde{\mathbf{c}}_i$, respectively) of node v_i, we construct the supervision by cross-predicting the pair-wise loss from two views: using the assignment probability from augmented view $\tilde{q}$ to guide the score p from the canonical view and vice versa:

$$\begin{aligned}\ell\left(\mathbf{c}_i, \tilde{\mathbf{c}}_i\right) &= -\sum_{h}^{H}\left[\tilde{q}_{ih}\log p_{ih} + q_{ih}\log \tilde{p}_{ih}\right] \\ &= -\sum_{h}^{H}\left[\tilde{q}_{ih}\log\frac{\exp\left(c_{ih}\right)}{\sum_j \exp\left(c_{ih}\right)} + q_{ih}\log\frac{\exp\left(\tilde{c}_{ih}\right)}{\sum_j \exp\left(\tilde{c}_{ih}\right)}\right].\end{aligned} \tag{7}$$

When migrating all the nodes, the total self-supervised loss for the prototypical head module is:

$$\mathcal{L}_P = \frac{1}{|\mathcal{V}_o|}\sum_{i\in\mathcal{V}_o}\ell\left(\mathbf{c}_i, \tilde{\mathbf{c}}_i\right). \tag{8}$$

As a result, shown in Fig. 2(b3), only the similar embeddings for target A are selected into the same prototype, i.e., nodes B_1-B_3 and C_1, for A's inference.

The final training loss is $\mathcal{L}_{SSL} = \mathcal{L}_N + \mathcal{L}_P$. We also summarize the computational complexity of the KCP, which is $\mathcal{O}(\sum_{i=1}^{L-1} NE_iE_{i+1} + \sum_{i=1}^{L_n}(N+K)E_i + \eta HN + NEH)$ (details in Suppl. 2).

4 EXPERIMENTS AND ANALYSIS

4.1 Experiments setup

Dataset For a broader performance evaluation of the proposed model, we conduct Kriging experiments on three publicly available time series datasets, since they are representative in particular application domains and are all with geographical properties. They are 1) **PeMS**: The dataset aggregates a 5-minute traffic flow across 325 stations, which is collected from the Caltrans Performance Measurement System over 2 months (January 1st, 2017 to March 1st, 2017). 2) **NREL**: It records 5-minute solar power output from 137 photovoltaic plants in Alabama in 2006, which is extracted from (Wu et al., 2021a). 3) **Wind**: This dataset records onshore renewable energy generation for Greece, which contains hourly wind speed aggregation on 18 installations over 4 years (Vartholomaios et al., 2021).

In each dataset, 80% randomly selected nodes are set as observations (i.e., $\mathcal{V}_o$) for training, and the remaining are regarded as unobserved nodes $v_u \in \mathcal{V}_u$ for testing. The data are normalized by the min-max scaler. More details are in Suppl. 3.3.

Training and testing For a full graph $\mathcal{G}$, we only sample a subgraph $\mathcal{G}_o$ with $|\mathcal{V}_o|$ nodes for training. The Kriging task is conducted under a pretraining and then fine-tuning paradigm. During pretraining, we optimize the proposed SSL loss for estimating the representations of augmented nodes in the augmented view. When fine-tuning, we use a 3-layer MLP to recover the node attributes according to the estimated representations of target nodes and finetune the parameters of the feature extraction module under the mean absolute error loss. In the testing stage, the $|\mathcal{V}_o|$ nodes are treated as known nodes while the remaining nodes in $\mathcal{G}$ serve as new-coming and unobserved locations, which are unavailable at the training stage.

Baselines We choose the following benchmarks: (1) statistical models: KNN with inverse distance weights, **KNN-IDW** for short, **XGBoost** (Chen and Guestrin, 2016), **OKriging** (Bostan, 2017), **GPMF** (Strahl et al., 2020), and (2) GNN-based models: **BGRL** (Thakoor et al., 2021), **IGNNK** (Wu et al., 2021a), and **KCN** (Appleby et al., 2020). The details about the methods are in Suppl. 3.4.

4.2 Results and detailed analysis

We conduct extensive experiments by answering the following questions: **Q1**: 1) Does the proposed model outperform the state-of-the-art baselines? 2) How will the models perform with a higher percentage of unseen nodes? **Q2**: What are the distinctive/advantages of the learned representations by SSL for Kriging? **Q3**: Which component is the most important in the proposed model? **Q4**: Any cases to support: misleading neighbors and constructive non-neighbors?

Kriging task performance (Q1.1): In Table 1, we summarize the performance of the proposed KCP and baselines on the three datasets. The best results are in bold and the second-best ones are underlined. Three commonly used metrics are adopted to evaluate

Table 2: The Kriging results on PeMS across different unsampled proportions

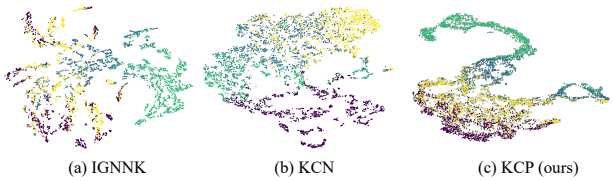
Models	MAE			RMSE		
	20%	50%	70%	20%	50%	70%
IGNNK	43.98	68.21	110.93	67.11	93.07	163.60
KCN	43.59	62.19	88.53	68.14	93.61	122.60
KCP	**42.29**	**59.50**	**87.74**	**65.37**	**86.49**	**114.53**
Imp. (↑)	3.0%	4.5%	0.9%	3.1%	4.3%	6.7%

the Kriging models, i.e., MAE, RMSE, and MAPE (definitions in Suppl. 3.5). The lower metrics indicate better performance.

It can be seen that the KCP outperforms its peers across almost all the datasets (about a maximum of +6% improvements than the second-best one), achieving the lowest metrics except for the RMSE in the PeMS dataset. Specifically, compared to statistical models, GNN-based deep models typically obtain better Kriging performance. KNN-IDW, the straightforward spatial interpolation, is surprisingly competitive in Kriging, since it goes beyond simple neighbor averaging and incorporates inverse distance-weighted interpolation which facilitates the neighboring information aggregation. XGBoost also shows promising performance in the Wind dataset. The reason may be that it takes latitude and longitude as input, which is more sensitive to geographic location, whereas wind speed often exhibits non-skewed spatial distributions and relates to geolocation. The BGRL achieves moderate outcomes, as there is no specific SSL design for the Kriging task. KCN outperforms the baselines on the PeMS and NREL datasets, while IGNNK is also impressive under RMSE. Additionally, a representative scheme of matrix factorization, GPMF, also achieves good performance. However, it lacks inductive capability, thus cannot handle the new nodes without re-training.

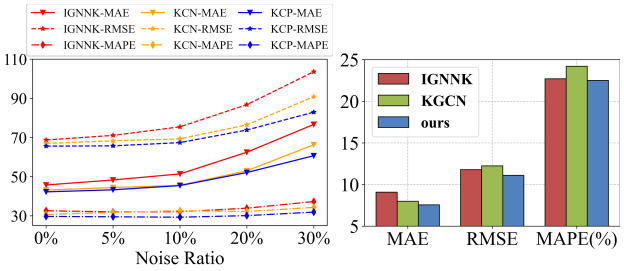
Different observation ratio (Q1.2): To evaluate the model's performance under different ratio observations, we look through the Kriging results by varying the unsampled ratio of unknown nodes. The unobserved ratio ranges from 20%, 50%, and 70%. For clarity, 20% means to infer 20% unobserved nodes by 80% known nodes. The results of the advanced GNN-based baselines and our model are summarized in Table 2. It can be seen that the superiority of the proposed model still holds with the changes in the unobserved ratio, which demonstrates its powerful Kriging ability across various scenarios.

The learned representations (Q2): Next, we visualize the learned representations of the time series across all the unobserved nodes. To identify inherent patterns, KMeans is adopted to cluster them according to the ground truth, making similar data appears in the same class. We use t-SNE to visualize the learned



(a) IGNNK (b) KCN (c) KCP (ours)

Figure 3: Embeddings of learned representations on PeMS. According to the ground truth, similar time series are clustered into the same class (same color).



(a) **Robustness**: The model performance over noise (random missing) ratios

(b) **Transferability**: Training on PeMS-Bay and transfer to META-LA

Figure 4: (a) Robustness and (b) Transferability.

representations of the unobserved time series in Fig. 3. It can be found that our representations are more compact, and different classes are with better separability in the latent space than IGNNK and KCN. ***This proves that our KCP learns the representation with promising quality.***

Robustness to confronting noise (Q2): In practical situations, noise is a common problem. For example, in the traffic monitoring system, the records in some monitored intersections may be missing when the devices are temporarily offline caused by unexpected events such as high temperature and network error. Therefore, the robustness to noise is also an important factor to be accounted for in the Kriging model. To verify the robustness of the models against noise, in the inference stage on PeMS, we set 5%, 10%, 20%, and 30% random missing for the attributes in the known nodes and investigate the estimations of the unknown nodes. The results are shown in Fig. 4(a). With the noise ratio increasing, the performance of all the models is degraded, while our model can still beat the strong GNN-based baselines. As the noise ratio increases, its superiority becomes more pronounced. ***This proves the robustness of the representations from KCP.***

Transferability validation (Q2): To evaluate the inductive performance with the unseen scenario, we exploit two independent datasets with the same attributes for evaluation, i.e., PeMS-Bay and METR-LA. Both two datasets are extracted from (Li et al., 2018b) with 5-min traffic speed aggregation. The model is trained on PeMS-Bay while transferring without retraining in METR-LA. The testing results are shown

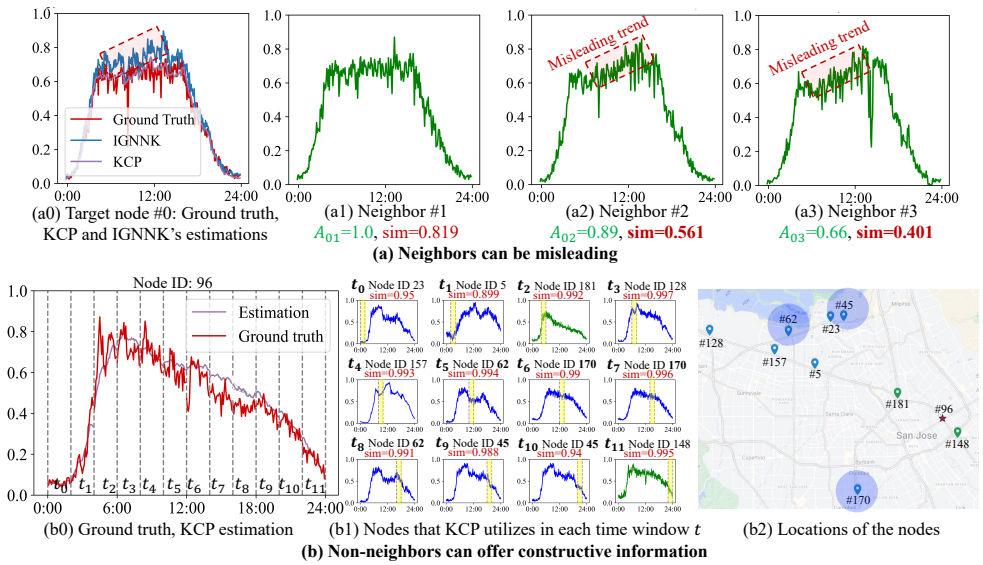


Figure 5: **(a) Neighbors can be misleading:** Baseline (IGNNK) uses misleading patterns from neighbors #2 and #3, who are close but less similar; **(b) Non-neighbors can be constructive**: KCP dynamically utilizes the patterns from the most similar nodes at each time t, e.g., at t_0, #23 is used since similarity = 0.95: some are neighbors (green), and some are non-neighbors (blue). Non-neighbors #45, #62, #170 used twice.

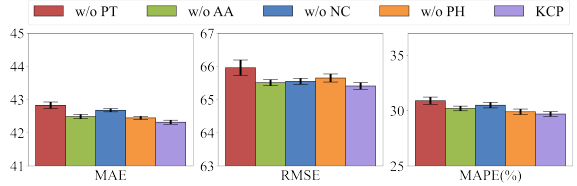


Figure 6: Ablation results on PeMS

in Fig. 4(b). The experiments show that the KCP can be generalized to an unseen scenario and achieves better transferability.

Ablations study (Q3): Since KCP contains three tailored components for spatiotemporal Kriging, we perform the following ablation experiments to provide insights into these components specifically. 1) without pretraining (w/o PT). The model is trained in an end-to-end way for Kriging without the SSL module. 2) without adaptive augmentation (w/o AA). We replace the adaptive augmentation module with a general view augmentation module, which simply masks all the selected nodes. 3) without the neighboring contrast (w/o NC). We remove the neighboring contrast module. 4) without the prototypical head (w/o PH). The prototypical head module is removed from the KCP. The experimental results on PeMS dataset are shown in Fig. 6. It can be concluded that (1) Compared to the model w/o PT, the introduce of SSL is beneficial to the model performance (about 2% $\sim$ 3% improvements). (2) The neighboring information aggregation is most crucial for the Kriging task, since the model without neighboring contrast (w/o NC) component causes the largest degradation. (3) The absence of the prototypical head component (w/o PH) demonstrates worse results than the proposed model, which indicates that the refining and recycling operations are still noteworthy. (4) The adaptive augmentation component is effective and helpful to the SSL model since its removal also hurts the model performance partly.

Visualization of Two Cases (Q4): (1) Misleading Neighbors and (2) Constructive Non-Neighbors. Fig. 5(a) shows: baselines such as IGNNK could get misled by the neighbors, e.g., IGNNK predicts an upward trend in the red box due to the misleading patterns from two quite dissimilar yet close neighbors #2 and #3. Fig. 5(b) shows: Our KCP makes the best of the features from different neighbors and non-neighbors at each time slot t. We also spatially visualize the kriging performance based on PeMS in Suppl. 3.6.

5 CONCLUSION

In this paper, we propose a self-supervised learning model for spatiotemporal Kriging. Rather than directly predicting the attributes of unobserved nodes, we achieve more robust Kriging by estimating the representations and then recovering them under the SSL framework. The tailored adaptive data augmentation and SSL modules can encourage data diversity and facilitate the model fully exploit helpful information, respectively. We not only enhance the neighboring aggregation ability of the GNN backbone by the neighboring contrast, but also emphasize the importance of refining the neighboring and recycling non-neighboring information by the prototypical head when constructing the SSL module.

References

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

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.

Bostan, P. (2017). Basic kriging methods in geostatistics. *Yuzuncu Yıl University Journal of Agricultural Sciences*, 27(1):10–20.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Chen, X., Chen, W., Chen, T., Yuan, Y., Gong, C., Chen, K., and Wang, Z. (2020). Self-pu: Self boosted and calibrated positive-unlabeled training. In *International Conference on Machine Learning*, pages 1510–1519. PMLR.

Chen, Y., Cheng, L., and Wu, Y.-C. (2023). Bayesian low-rank matrix completion with dual-graph embedding: Prior analysis and tuning-free inference. *Signal Processing*, 204:108826.

Ci, Y., Lin, C., Bai, L., and Ouyang, W. (2022). Fastmoco: Boost momentum-based contrastive learning with combinatorial patches. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 290–306. Springer.

Cuturi, M. (2013). Sinkhorn distances: lightspeed computation of optimal transport. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 2292–2300.

Dai, S., Wang, J., Huang, C., Yu, Y., and Dong, J. (2021). Temporal multi-view graph convolutional networks for citywide traffic volume inference. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1042–1047. IEEE.

Egressy, B. and Wattenhofer, R. (2022). Graph neural networks with precomputed node features. *arXiv preprint arXiv:2206.00637*.

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)*, pages 1327–1334. IEEE.

Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., and Liu, Y. (2019). Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 33, pages 3656–3663.

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

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035.

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

Han, X., Jiang, Z., Liu, N., and Hu, X. (2022b). G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, pages 8230–8248. PMLR.

Hassani, K. and Khasahmadi, A. H. (2020). Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126. PMLR.

Jain, P. and Oh, S. (2014). Provable tensor factorization with missing data. *Advances in Neural Information Processing Systems*, 27.

Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Jin, M., Zheng, Y., Li, Y.-F., Chen, S., Yang, B., and Pan, S. (2022). Multivariate time series forecasting with dynamic graph neural odes. *IEEE Transactions on Knowledge and Data Engineering*.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kiryo, R., Niu, G., Du Plessis, M. C., and Sugiyama, M. (2017). Positive-unlabeled learning with non-negative risk estimator. *Advances in Neural Information Processing Systems*, 30.

Klemmer, K., Safir, N. S., and Neill, D. B. (2023). Positional encoder graph neural networks for geographic data. In *International Conference on Artificial Intelligence and Statistics*, pages 1379–1389. PMLR.

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.

Kwong, K., Kavaler, R., Rajagopal, R., and Varaiya, P. (2009). Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors. *Transportation Research Part C: Emerging Technologies*, 17(6):586–606.

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, 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, Y., Yu, R., Shahabi, C., and Liu, Y. (2018a). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2018b). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*.

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, pages 4804–4810.

Li, Z., Xiong, G., Tian, Y., Lv, Y., Chen, Y., Hui, P., and Su, X. (2020b). A multi-stream feature fusion approach for traffic prediction. *IEEE transactions on intelligent transportation systems*, 23(2):1456–1466.

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*, pages 389–398.

Liu, C., Wen, L., Kang, Z., Luo, G., and Tian, L. (2021). Self-supervised consensus representation learning for attributed graph. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2654–2662.

Liu, Y., Jin, M., Pan, S., Zhou, C., Zheng, Y., Xia, F., and Yu, P. (2022). Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Lovász, L. (2012). *Large networks and graph limits*, volume 60. American Mathematical Soc.

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*, pages 1501–1510.

Meng, C., Yi, X., Su, L., Gao, J., and Zheng, Y. (2017). City-wide traffic volume inference with loop detector data and taxi trajectories. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10.

Rao, N., Yu, H.-F., Ravikumar, P. K., and Dhillon, I. S. (2015). Collaborative filtering with graph information: Consistency and scalable methods. *Advances in Neural Information Processing Systems*, 28.

Rasmussen, C. E., Williams, C. K., et al. (2006). *Gaussian processes for machine learning*, volume 1. Springer.

Strahl, J., Peltonen, J., Mamitsuka, H., and Kaski, S. (2020). Scalable probabilistic matrix factorization with graph-based priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5851–5858.

Tang, S., Zhu, F., Bai, L., Zhao, R., Wang, C., and Ouyang, W. (2022). Unifying visual contrastive learning for object recognition from a graph perspective. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 649–667. Springer.

Thakoor, S., Tallec, C., Azar, M. G., Azabou, M., Dyer, E. L., Munos, R., Veličković, P., and Valko, M. (2021). Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*.

Tonekaboni, S., Eytan, D., and Goldenberg, A. (2021). Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*.

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

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11).

Vartholomaios, A., Karlos, S., Kouloumpris, E., and Tsoumakas, G. (2021). Short-term renewable energy forecasting in Greece using prophet decomposition and tree-based ensembles. In *Database and Expert Systems Applications-DEXA 2021 Workshops: BIOKDD, IWCFS, MLKgraphs, AI-CARES, ProTime, AISys 2021, Virtual Event, September 27–30, 2021, Proceedings 32*, pages 227–238. Springer.

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, pages 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*.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 1907–1913.

Yang, J.-M., Peng, Z.-R., and Lin, L. (2021). Real-time spatiotemporal prediction and imputation of traffic status based on lstm and graph laplacian regularized matrix factorization. *Transportation Research Part C: Emerging Technologies*, 129:103228.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. (2020a). Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823.

You, Y., Chen, T., Wang, Z., and Shen, Y. (2020b). When does self-supervision help graph convolutional networks? In *International Conference on Machine Learning*, pages 10871–10880. PMLR.

Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640.

Yu, Y., Tang, X., Yao, H., Yi, X., and Li, Z. (2019). Citywide traffic volume inference with surveillance camera records. *IEEE Transactions on Big Data*, 7(6):900–912.

Zhang, Z., Li, M., Lin, X., and Wang, Y. (2020). Network-wide traffic flow estimation with insufficient volume detection and crowdsourcing data. *Transportation Research Part C: Emerging Technologies*, 121:102870.

Zheng, C., Fan, X., Wang, C., Qi, J., Chen, C., and Chen, L. (2023). Increase: Inductive graph representation learning for spatio-temporal kriging. In *Proceedings of the ACM Web Conference 2023*, pages 673–683.

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*, pages 403–414. SIAM.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. (2021). Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [No]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials for "Non-Neighbors Also Matter to Kriging: A New Contrastive-Prototypical Learning"

6 MORE DETAILED RELATED WORK

6.1 Kriging and SSL methods

As mentioned in the main text, the fundamental intuition behind Kriging is to model the spatial correlation across observed points to estimate the attributes at unobserved locations (Krige, 1951; Goovaerts, 1998). The taxonomy of Kriging methods can be categorized as *transductive* and *inductive*. (1) Transductive models require all the nodes to be present during training, and it cannot learn the representation for the unseen nodes in a natural way: re-training the model with the new nodes is needed. Classic models such as matrix factorization, DeepWalk, and GCN are by default transductive. (2) Inductive models instead can directly handle the new nodes that are unseen during training. It can accommodate dynamic graphs and learn the representation of unseen nodes. Here we will introduce more details in Transductive Kriging.

Transductive Kriging Recently, there has been a significant focus on learning-based variants of Kriging to capture spatiotemporal patterns dynamically. (1) Several previous **GCN-based studies** construct spatiotemporal affinity matrices about the observed and unobserved nodes, and infer the city-wide traffic volume with the constraints of the spatiotemporal consistency (Meng et al., 2017; Yu et al., 2019; Dai et al., 2021). However, the predefined affinity matrices usually make the model transductive, and thus they are unable to effectively generalize to infer new nodes. (2) Another transductive stream emphasizes the **statistical approaches**, matrix/tensor factorization, for filling the unobserved attributes, thus the spatiotemporal data can be decomposed into the product of low-rank matrices (Jain and Oh, 2014). In this case, the node features are arranged under matrix/tensor forms, in which the unobserved nodes are embodied as completely-missing rows. The geographic structure is introduced as auxiliary information or priors (Zhou et al., 2012; Strahl et al., 2020; Chen et al., 2023). To estimate the network-wide traffic volume, Zhang et al. (2020) propose to incorporate floating car data into a geometric matrix completion model and add a divergence-based spatial smoothing index to measure the difficulty of estimation in each road segment. Lei et al. (2022) present a Bayesian kernelized matrix factorization model to capture the spatiotemporal dependencies among the data rows and columns, which is regularized by Gaussian process (Rasmussen et al., 2006) priors over the columns of factorized matrixes. Such methods are still transductive and lack real-time inference capabilities, since newly added nodes will increase the number of rows in a matrix, and thus re-factorization is inevitable.

Self-supervised learning on graphs In SSL, predictions or labels are generated from raw data and guide the model's learning by pretext tasks. Contrastive learning, as a sub-domain of SSL, aims to learn general representations by maximizing the mutual information (MI) between positive (similar) and negative (distinct) samples that are generated from data augmentation (Ci et al., 2022; Tang et al., 2022). The key lies in how to construct and define the positive/negative samples, which affects the representation quality largely. In the context of Kriging, the choice of positive/negative samples could mean which nodes will be used to infer the unseen node. Graph SSL (You et al., 2020b; Liu et al., 2022) offers potential solutions. The augmentation of graph SSL is usually either in node attributes or structural topology (Hassani and Khasahmadi, 2020; You et al., 2020a; Zhu et al., 2021).

When specifying positive/negative, similar to the Kriging norm, neighbors are usually considered as positive samples (Mao et al., 2022) and non-neighbors as negative samples. However, scholars also realized the problem with the negative sample definition. Grill et al. (2020) and Thakoor et al. (2021) stated that it is rather difficult to contrast a realistic but semantically dissimilar augmented sample; thus, a contrast loss without negative samples was proposed. Kiryo et al. (2017); Chen et al. (2020); Tonekaboni et al. (2021) raised the concern of *negative sampling bias*, where blindly drawing samples from the distribution outside of the positive samples may result in

negative samples that turn out quite similar to the reference; thus, "Positive-Unlabeled" learning is proposed: outside of the positive region, it is an unlabeled region with a weighted combination of w positive and $1-w$ negative.

We improve the dilemma of positive/negative even further, considering both positive and negative are not strictly cut off, both with exceptions. We propose "Contrastive-Prototypical" learning for Kriging, where the contrastive module coarsely defines positive/negative as neighboring/non-neighboring and the prototypical module refines the positive and recycles the negative. Besides, an adaptive augmentation is also proposed to choose the feature mask and node mask in a probabilistic manner, encouraging higher diversity.

6.2 Methods that consider non-neighbors

Outside the scope of Kriging, there are methods that also consider the nodes that are not physical neighbors. They mainly introduce more semantic graphs such as Point of Interest (POI) (Li et al., 2020a), transition (Mao et al., 2022), and connectivity (Geng et al., 2019) (which are the most popularly defined graphs), besides the topological graph. These semantic graphs are either incorporated as additional graph Laplacian penalties in statistical models such as matrix factorization (Yang et al., 2021) and tensor decomposition (Li et al., 2020a), or constructed as multi-view graphs in GCN-based models (Geng et al., 2019). However, these explicitly defined graphs require domain knowledge, which might not be wholesome to explain why two nodes are similar. Moreover, these methods are transductive. INCREASE (Zheng et al., 2023) is the first inductive method that explicitly defined the three graphs, i.e., spatial proximity, function similarity, and transition probability for Kriging. It shows that, with more graphs considered, the performance could consistently increase. However, the questions are: are three graphs enough and accurate to describe how different nodes utilize each other's information? If the nodes are correlated with ten different patterns, do ten graphs need to be constructed? How to construct graphs that are outside the scope of human domain knowledge? Therefore, we propose to select similar or dissimilar nodes and show nodes utilize each others' information in a learning base: no explicit definitions are required, and most likely, the patterns that are not under the awareness of domain knowledge could also be captured.

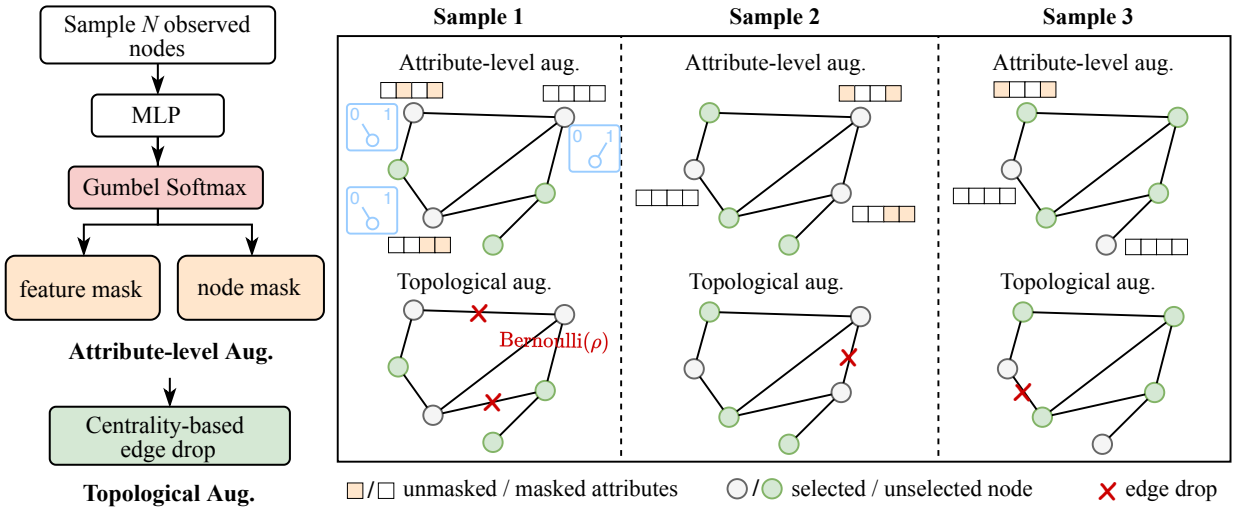
7 COMPUTATIONAL COMPLEXITY ANALYSIS

The proposed KCP includes three main modules, each contributing to the overall computational complexity. The complexities of these modules are as follows:
(1) GNN (i.e., GraphSAGE) is an off-the-shelf component and its complexity is $\mathcal{O}(\sum_{i=1}^{L-1} NE_iE_{i+1})$, where L is the number of GNN layers, N indicates the number of nodes in the graph, and E_i means the dimension of the i-th layer representations (embeddings). (2) The neighboring contrast module is designed to supervise target nodes using representations from their neighbors, which involves aggregating features from K neighbors for each node in the graph. Consequently, the computational complexity of this module is expressed as $\mathcal{O}(\sum_{i=1}^{L_n}(N+K)E_i)$, where L_n indicates the number of aggregation layers (defaulted to 1) and K participates in the attention-based readout. (3) The prototype head incorporates a linear projection and the Sinkhorn algorithm. Since the implementation of the Sinkhorn involves alternately iterative normalization of the rows and columns for the matrix, during the model training, it contributes to a computational complexity denoted as $\mathcal{O}(\eta HN)$, where η represents the number of iterations and H is the number of prototypes. Associated with the $\mathcal{O}(NEH)$ complexity of the linear projection, the overall computational complexity of the prototype head is expressed as $\mathcal{O}(\eta HN + NEH)$. To sum up, the total computational complexity of the KCP model is the summarization of the three modules, that is, $\mathcal{O}(\sum_{i=1}^{L-1} NE_iE_{i+1} + \sum_{i=1}^{L_n}(N+K)E_i + \eta HN + NEH)$.

8 IMPLEMENTATION DETAILS

8.1 The illustration for data augmentation

As shown in Fig. 7, we further diagram the proposed data augmentation module. Specifically, among the observed nodes, we first randomly select N nodes while keeping the remaining nodes unchanged. Then, attribute-level and topological augmentations are sequentially applied to each selected node (shown in Fig. 7(a)). In Fig. 7(b), we showcase the internal modifications on three samples with augmentations.



(a) The adaptive data augmentation pipeline

(b) Illustration of obtaining 3 samples after attribute-level and topological augmentations with $N = 3$. In the training stage, $N_o = 6$.

Figure 7: The diagram of adaptive data augmentation module.

8.2 Proof of Theorem 3.1

For the topologically augmented graphon W', there is $W' = (\mathbf{1} - \Phi) \odot W$. Recall that the definition of homomorphism density t is $t(\mathcal{G}, W) = \int_{[0,1]^{V(\mathcal{G})}} \prod_{i,j \in E(\mathcal{G})} W(x_i, x_j) \prod_{i \in V(\mathcal{G})} dx_i$. And the homomorphism density of W' is

$$
\begin{aligned}
t(\mathcal{G}, W') &= \int_{[0,1]^{|V(\mathcal{G})|}} \prod_{i,j \in E(\mathcal{G})} W'(x_i, x_j) \prod_{i \in V(\mathcal{G})} dx_i \\
&= \int_{[0,1]^{|V(\mathcal{G})|}} \prod_{i,j \in E(\mathcal{G})} ((\mathbf{1} - \Phi) \odot W)(x_i, x_j) \prod_{i \in V(\mathcal{G})} dx_i \\
&= \int_{[0,1]^{|V(\mathcal{G})|}} \prod_{i,j \in E(\mathcal{G})} (1 - \Phi_{i,j}) W(x_i, x_j) \prod_{i \in V(\mathcal{G})} dx_i \\
&= \prod_{i,j \in E(\mathcal{G})} (1 - \Phi_{i,j}) \int_{[0,1]^{|V(\mathcal{G})|}} \prod_{i,j \in E(\mathcal{G})} W(x_i, x_j) \prod_{i \in V(\mathcal{G})} dx_i \\
&= \prod_{i,j \in E(\mathcal{G})} (1 - \Phi_{i,j}) t(\mathcal{G}, W)
\end{aligned}
\tag{9}
$$

Since $\Phi_{i,j}$ is determined by sampling under the edge drop probability ρ and irrelevant to dx, it can be excluded from the integration. Following triangle inequality in G-mixup Han et al. (2022b), $|t(\mathcal{G}, W') - t(\mathcal{G}, W)| \leqslant e(\mathcal{G}) \|W - W'\|_{\square}$. Let $\lambda = \prod_{i,j \in E(\mathcal{G})} (1 - \Phi_{i,j})$, and then the upper bound is derived as

$$
\left| t(\mathcal{G}, W') - t(\mathcal{G}, W) \right| = |\lambda t(\mathcal{G}, W) - t(\mathcal{G}, W)| \leqslant e(\mathcal{G}) \|(1 - \lambda) W\|_{\square} = (1 - \lambda) e(\mathcal{G}) \|W\|_{\square} \tag{10}
$$

8.3 Dataset description

For a broader performance evaluation of the proposed model, we conduct Kriging experiments on three publicly available time series datasets with geographic properties, which are described in Table 3.

1) PeMS: The dataset aggregates a 5-minute traffic flow across 325 traffic stations, collected from the Caltrans Performance Measurement System over a 2-month period (January 1st, 2017 to March 1st, 2017). 2) NREL: It records 5-minute solar power output from 137 photovoltaic plants in Alabama in 2006, which is extracted from (Wu et al., 2021a). 3) Wind: This dataset records onshore renewable energy generation for Greece, which contains hourly wind speed aggregation on 18 installations over 4 years (2017-2020) (Vartholomaios et al., 2021).

Table 3: Dataset description

	Attributes	Nodes	Resolution
PeMS	Traffic flow	325	5 min
NREL	Solar power	137	5 min
Wind	Wind speed	18	hourly

Gaussian kernel-based inverse distance is employed to calculate the adjacent matrix $\mathbf{A}$ following (Li et al., 2018b), where road network distance is for the PeMS dataset and Euclidean distance is for NREL and Wind datasets. For nodes v_i and v_j, the related entry $\mathbf{A}_{ij}$ in $\mathbf{A}$ is calculated by

$$\mathbf{A}_{ij} = \exp\left(-\left(\frac{\text{dist}\left(v_i, v_j\right)}{\sigma}\right)^2\right), \tag{11}$$

where σ is the standard deviation of the distance.

8.4 Baselines

We compare seven advanced Kriging models involving statistical models and deep learning-based methods. They are 1) KNN-IDW: A KNN model which incorporates inverse distance weighted interpolation for Kriging. The weighted average of the K nearest observed nodes to the unobserved node is taken as the estimation, and the weight is associated with the entry in the adjacent matrix. We set $K = 5$ in this paper. 2) XGBoost (Chen and Guestrin, 2016): We train an extreme gradient boosting model, in which the geolocations are regarded as the input, and the values of the attributes are the output. 3) OKriging (Bostan, 2017): Ordinary Kriging. It estimates the unknown nodes with a known variogram under the Gaussian process, which is a typical spatial interpolation model. 4) GPMF (Strahl et al., 2020): It is a graph-based prior probabilistic matrix factorization, in which the graph structure is used as the side information to achieve the matrix factorization. 5) BGRL (Thakoor et al., 2021): It is a self-supervised graph representation learning method that learns by predicting alternative augmentations of the input. It uses only simple augmentations without explicit negative samples. The model is also tailored to the Kriging task under the pretraining and then finetuning SSL paradigm. 6) IGNNK (Wu et al., 2021a): It constructs dynamic subgraphs by random sampling and uses diffusion graph convolutional network (Li et al., 2018b) for the spatiotemporal Kriging. 7) KCN (Appleby et al., 2020): A graph convolutional network for Kriging, which makes direct use of K nearest neighboring observations for graph message passing. We also set $K = 5$.

For the neural networks, we split the time series attributes in each node by slide window with size 24 for input.

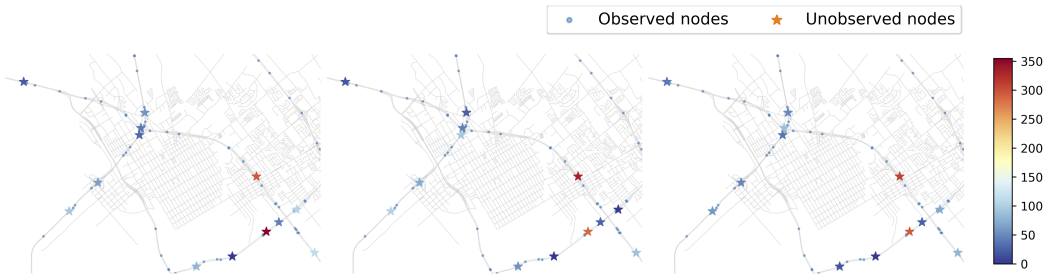
8.5 Evaluation metrics

To quantitatively characterize the performance of Kriging models, mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) are adopted to evaluate the estimations $\hat{\mathbf{Y}}$ and ground truth $\mathbf{Y}$ on unobserved nodes. The first reflects the average error between $\hat{\mathbf{Y}}$ and $\mathbf{Y}$. The second provides higher weights to larger errors, which is more sensitive to outliers. The last one indicates the average percentage deviation of the $\hat{\mathbf{Y}}$ from $\mathbf{Y}$. Their definitions are

$$\text{MAE} = \frac{1}{N_u \times T} \sum_{j=1}^{N_u} \left(\sum_{t=1}^{T} \left| y_j^t - \hat{y}_j^t \right| \right), \tag{12}$$

$$\text{RMSE} = \left[\frac{1}{N_u \times T} \sum_{j=1}^{N_u} \left(\sum_{t=1}^{T} \left(y_j^t - \hat{y}_j^t \right)^2 \right) \right]^{1/2}, \tag{13}$$

$$\text{MAPE} = \frac{1}{N_u \times T} \sum_{j=1}^{N_u} \left(\sum_{t=1}^{T} \frac{\left| y_j^t - \hat{y}_j^t \right|}{y_j^t} \right), \tag{14}$$



(a) IGNNK (b) KCN (c) KCP (ours)

Figure 8: The visualization of Kriging errors at a particular time point on the PeMS. The stars represent the unknown nodes and the dot points mean the observed nodes.

where N_u denotes the number of unobserved nodes. T is the time horizon to perform Kriging. y_j^t and $\hat{y}_j^t$ are scalars at the time step t on node v_j, which indicate ground truth and the estimation, respectively.

8.6 Spatial visualization of Kriging results

Based on PeMS, we further provide intuitive visualization about the absolute Kriging errors across the geographic locations, which are calculated by $|estimation - ground\ truth|$ and illustrated in Fig. 8. For demonstration, we choose the traffic flow of the nodes at a particular time and show the Kriging errors from the graph-based baselines and our method, respectively. It can be seen that the errors in our model are the smallest, which also indicates its superiority.