CLEAR: Spatial-temporal Traffic Data Representation Learning for Traffic Prediction

James Jianqiao Yu, *Senior Member, IEEE*, Xinwei Fang, Shiyao Zhang, *Member, IEEE*, and Yuxin Ma, *Senior Member, IEEE*

Abstract—In the evolving field of urban development, precise traffic prediction is essential for optimizing traffic and mitigating congestion. While traditional graph learning-based models effectively exploit complex spatial-temporal correlations, their reliance on trivially generated graph structures or deeply intertwined adjacency learning without supervised loss significantly impedes their efficiency. This paper presents Contrastive Learning of spatial-tEmporal trAffic data Representations (CLEAR) framework, a comprehensive approach to spatial-temporal traffic data representation learning aimed at enhancing the accuracy of traffic predictions. Employing self-supervised contrastive learning, CLEAR strategically extracts discriminative embeddings from both traffic time-series and graph-structured data. The framework applies weak and strong data augmentations to facilitate subsequent exploitations of intrinsic spatial-temporal correlations that are critical for accurate prediction. Additionally, CLEAR incorporates advanced representation learning models that transmute these dynamics into compact, semantic-rich embeddings, thereby elevating downstream models' prediction accuracy. By integrating with existing traffic predictors, CLEAR boosts predicting performance and accelerates the training process by effectively decoupling adjacency learning from correlation learning. Comprehensive experiments validate that CLEAR can robustly enhance the capabilities of existing graph learning-based traffic predictors and provide superior traffic predictions with a straightforward representation decoder. This investigation highlights the potential of contrastive representation learning in developing robust traffic data representations for traffic prediction.

Index Terms—Traffic prediction, spatial-temporal data, contrastive learning, representation learning, self-supervised learning.

1 Introduction

In the ever-evolving landscape of urban development and mobility management, the analysis and prediction of traffic data play essential roles [1], [2]. Accurate traffic predictions allow city planners, traffic management systems, and navigation services to anticipate and mitigate traffic issues, thereby enhancing urban mobility [3]. By leveraging historical and real-time traffic data, predictive models can forecast traffic dynamics, enabling online monitoring of the transportation system and proactive measures to traffic management [4].

In recent years, graph learning-based traffic predictors have emerged as a dominant approach in the field of traffic data analysis [2]. These models utilize the natural graph structure of transportation networks, where intersections and road segments are represented as nodes and edges, respectively. The strength of graph-based models lies in their ability to capture the complex spatial-temporal dependencies between these nodes, facilitating more accurate and granular traffic predictions. By integrating techniques such as Graph Convolutional Network (GCN) [5], these models can effectively process and learn from the vast amounts of spatial-temporal data generated by traffic systems, thus significantly enhancing traffic prediction accuracy [6].

Despite their advantages, graph learning-based traffic predictors face significant challenges that impede their efficacy. The first major challenge arises from the conventional method of constructing graph node connectivity. Typically, these graphs are formed based on geographic distances or traffic connectivity, ignoring the contextual relationships between nodes or the temporal dynamics of traffic flow [7]. Static topology graphs do not reflect the real-time, dynamic nature of traffic, which can vary significantly due to various time-dependent factors such as peak-valley flows [8]. There exists research on capturing dynamic spatial correlations in spatiotemporal data by using a learnt 3-D tensor [9]. Nonetheless, the number of additional trainable parameters (e.g., the learnable adjacency matrix requires approx. $120\,\mathrm{M}$ parameters for 1000-node 1-hour lookback forecast) may overwhelm the model training process.

The second challenge pertains to the strong coupling of adjacency and data correlation learnings during model training process used in these models. Except for static adjacency matrices, there are other traffic predictors utilizing one or more learnable matrices to adaptively learn the adjacency relationships between nodes [10]. However, this method couples the learning of adjacency information with

This work is supported in part by National Natural Science Foundation of China (No. 62202217), in part by Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515012889), and in part by Guangdong Key Program (No. 2021QN02X794). Shiyao Zhang and Yuxin Ma are the corresponding authors.

James Jianqiao Yu is with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China, and with the Department of Computer Science, University of York, York YO10 5GH, United Kingdom (email: jqyu@ieee.org).

Xinwei Fang is with the Department of Computer Science, University of York, York YO10 5GH, United Kingdom (email: xinwei.fang@york.ac.uk).

Shiyao Zhang is with the School of Engineering, Great Bay University, Dongguan 523000, China, and with the Great Bay Institute for Advanced Study (GBIAS), Dongguan 523000, China (email: zhangshiyao@gbu.edu.cn).

Yuxin Ma is with the Department of Computer Science and Engineering, Southern University of Science and Technolgy, Shenzhen 518055, China (email: mayx@sustech.edu.cn).

that of the intrinsic data correlation, leading to a complex bilevel optimization problem. Such complexity increases the computational intensity of the training process, requesting notably more data and computation to achieve optimal performance according to the scaling law [11]. What makes the situation worse is that the former typically does not have supervised loss information to facilitate a guided search [12]. The complexity not only makes the training process more challenging but also increases the risk of overfitting or underfitting, thereby potentially reducing the model's overall effectiveness and adaptability to new data or environments [13].

To bridge the research gap, we propose a novel CLEAR framework (**C**ontrastive **L**earning of spatial-t**E**mporal tr**A**ffic data **R**epresentations) specifically designed to address the aforementioned challenges in traffic prediction. CLEAR utilizes the power of self-supervised contrastive learning [14]–[16] to extract discriminative embeddings from both traffic time-series and graph-structured data. The core principle of CLEAR revolves around capturing intrinsic spatial and temporal correlations inherent in traffic data, leveraging these learned representations for accurate traffic prediction, and facilitating seamless integration with existing graph learning-based traffic predictors. CLEAR employs a strategy of using weak and strong data augmentation techniques to facilitate the contrastive representation learning process, which enhances the model's ability to understand diverse traffic patterns and respond to dynamic changes in the network [17]. This approach allows CLEAR to dynamically generate and update adjacency information that reflects the real-time contextual relationships and connectivity changes over time, effectively addressing the first challenge.

Moreover, by learning robust representations that encode essential traffic information and substituting the input data and/or internal modules, CLEAR greatly simplifies the training process and difficulty of existing graph learning-based predictors. Principally, CLEAR decouples the adjacency learning from data correlation, thus mitigating the bilevel optimization complexity associated with these models. The training of integrated models becomes more straightforward as CLEAR eliminates the need for additional representation extraction steps during model training [18]. By incorporating these representations, existing predictors can also benefit from the rich semantic information captured through self-supervised contrastive learning, thereby enhancing their ability to model complex spatial and temporal dependencies within traffic data, directly tackling the second challenge.

The contributions of this paper are multifaceted:

- **Design of the CLEAR Framework**: We introduce CLEAR, a novel learning framework that utilizes contrastive learning to extract detailed and discriminative representations from spatial-temporal traffic data. This framework is designed to capture intrinsic correlations within traffic data to develop semantic-rich data representations for downstream applications, e.g., traffic prediction.
- **Data Augmentation Strategies**: We develop specialized data augmentation strategies that involve weak and strong manipulation techniques. These strategies are critical for enriching the training data and enabling the model to learn robust features from diverse traffic patterns, thereby improving the generalizability of the learned models.
- **Representation Learning Models**: We devises comprehensive representation learning models for both time-series and graph-structured traffic data. These models are capable of encoding crucial traffic dynamics into compact, information-rich embeddings, which are essential for accurately predicting traffic conditions.
- **Traffic Predictor Based on Representations**: We propose a simple-yet-effective traffic predictor that benefits from the representations learned by CLEAR. This predictor utilizes the embeddings to predict future traffic conditions, demonstrating how learned representations can achieve accurate traffic predictions.
- **Bootstrapping Graph Learning-Based Predictors**: We present novel bootstrapping techniques that integrate CLEAR with existing graph learning-based predictors. These techniques leverage the rich semantic embeddings from CLEAR to bootstrap existing traffic prediction models for performance improvements.

The remainder of this paper is structured as follows: Section 2 reviews related work in graph learning-based traffic predictors and contrastive representation learning. Section 3 outlines the preliminary definitions and problem formulation specific to this study. Section 4 details the CLEAR framework, including our proposed data augmentation strategies and representation learning models, and discusses the integration of CLEAR with existing graph learning-based predictors. Section 5 describes the experimental setup and presents a comprehensive evaluation of CLEAR's performance across various datasets. Section 6 concludes the paper, summarizing our contributions and suggesting avenues for future research.

2 Related Work

In this section, we briefly review the related work on graph learning-based traffic predictors and contrastive representation learning. The audience are referred to [6], [19]–[21] for more thorough discussions and analyses.

2.1 Graph learning-based Traffic Predictors

Graph deep learning has emerged as a transformative approach in traffic prediction, utilizing the rich spatial interconnections within traffic systems to significantly enhance prediction accuracy [6]. This evolution from traditional time-series models to complex graph-based approaches has catalyzed the development of more sophisticated predictive tools, capable of understanding the intricate dynamics of traffic flow that are critical for accurate real-time traffic prediction. Given the extensive body of related work in this field, this section focuses on a few pivotal models due to page limits.

One of the pioneering models in this domain is Graph WaveNet [10], which ingeniously combines graph neural networks with WaveNet's temporal convolutional approach. This model is designed to capture spatial dependencies through graph convolutions while utilizing dilated causal convolutions to handle temporal sequences efficiently. By integrating these two mechanisms, Graph WaveNet addresses both spatial and temporal aspects of traffic data, leading to improved accuracy in short-term traffic prediction. Its ability to model dynamic spatial structures without predefined adjacency matrices sets it apart, allowing it to adapt to various traffic scenarios and predict potential congestions with higher precision.

Another exemplary model is the Dynamic Graph Convolutional Recurrent Network (DGCRN) [22], which introduces an innovative approach to traffic prediction by addressing the dynamic nature of road network correlations. Unlike static models, DGCRN uses hyper-networks to capture dynamic characteristics from node attributes, updating the parameters of its filters at each time step to reflect ongoing changes. This allows for the creation of a dynamic graph that integrates with a pre-defined static graph, offering a more accurate representation of real-time traffic conditions.

Recently, the introduction of Transformers in traffic prediction like the Spatio-Temporal Adaptive Embedding Transformer (STAEformer) has brought new dimensions to this field [23]. The STAEformer leverages the Transformer architecture known for its effectiveness in natural language processing to the realm of traffic prediction. It utilizes a novel spatio-temporal adaptive embedding that enhances the model's ability to capture complex spatio-temporal relationships. STAEformer highlights the effectiveness of combining advanced embedding techniques with the self-attention mechanism of transformers, providing a powerful tool for traffic prediction that surpasses many traditional and graph-based methods.

Despite these advancements, the field of graph deep learning for traffic prediction faces several challenges. The (semi-)static nature of many graph constructions does not reflect the dynamic changes in real-world traffic conditions over time. Further, the computational intensity of these models, especially those based on transformers, poses difficulties for efficient model training on large-scale networks.

2.2 Contrastive Representation Learning

Contrastive representation learning develops an embedding space where similar instances are grouped together and dissimilar ones are separated. This approach is employed across various fields including natural language processing, computer vision, and time-series analysis. InfoNCE loss [24], SimCLR [15], and MoCo [25] are among the general and classical contrastive representation learning models that utilize pairs of positive and negative examples to refine this space, achieving robust results and setting the foundation for more sophisticated methods.

While contrastive representation learning for traffic data is relatively scarce, recent years have witnessed advancements on contrastive time-series representation learning. For example, the Time-Series representation learning framework via Temporal and Contextual Contrasting (TS-TCC) represents a significant advancement in exploiting unlabeled time-series data [26]. TS-TCC utilizes dual view augmentations to transform raw time-series into correlated views, employing a novel temporal contrasting module that challenges the model with a cross-view prediction task. Another innovative approach is the Temporal Neighborhood Coding (TNC) [27], which leverages the inherent local smoothness of time series to establish temporally stationary neighborhoods. Further, the STEP algorithm introduces a pre-training model to spatial-temporal graph neural networks [18], utilizing long-term historical data to enhance the contextual understanding essential for accurate multivariate time-series forecasting. Additionally, the Contrastive Seasonal-Trend representation learning framework (CoST) innovatively applies contrastive learning to disentangle seasonal and trend components of time series data [28]. By incorporating both time and frequency domain contrastive losses, CoST effectively learns discriminative features that significantly outperform traditional methods on multivariate time series prediction, underscoring its robustness across different neural architectures and regression models.

While the aforementioned representation learning approaches achieve satisfactory results in general time-series tasks, their capability of bootstrapping arbitrary graph learning-based traffic preditors generally remains unknown. This is among the primary objectives of CLEAR.

When confining the graph-structured data into grid-based presentations, self-supervised learning approaches have been applied to learn spatial-temporal representations. Notable past efforts include [29] and [30]. In the former, the proposed ST-SSL framework utilizes an integrated module with temporal and spatial convolutions to encode information across space and time. It employs adaptive augmentation on traffic flow data and incorporates two self-supervised learning auxiliary tasks to enhance the main traffic prediction task with spatial and temporal heterogeneity awareness. Further, the method in [30] introduces a contrastive self-supervision approach to predict fine-grained urban flows by leveraging correlated spatial and temporal patterns. It employs self-supervised tasks to extract high-level representations from flow data and utilizes a fine-tuning network combined with three pre-trained encoder networks for enhanced performance. While both methods achieved promising results in their respective tasks, they are limited to grid-based traffic data and require non-trivial effort to adapt to graph-structured traffic data.

3 Preliminaries

In this section, we first introduce the definitions and notations to be used in this paper. Then, we define the spatial-temporal traffic data representation learning and prediction problem.

3.1 Definitions

Definition 1 (Traffic Network)**.** In this work, the traffic network is conceptualized as a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of traffic data sensing locations, such as induction loops and surveillance cameras. The nodal connectivity, $\mathcal{E}$, is typically defined by geographical adjacency, which informs the creation of the adjacency matrix $\mathbf{A}$. We contend that this conventional approach to defining $\mathcal{E}$ and $\mathbf{A}$ fails to adequately capture the complexities of traffic dynamics. To address this, we advocate for the use of representation learning to develop semantically rich embeddings that enhance our construction of adjacency information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536009

Definition 2 (Spatial-temporal Traffic Data)**.** A set of spatial-temporal traffic data comprises multiple time-dependent variables captured from traffic sensors distributed across $\mathcal{V}$. This data is represented as $\mathbf{X} = \{x_{i,t}\}$ within a real number space $\mathbb{R}^{|\mathcal{V}|\times|\mathcal{T}|\times F}$, where $\mathcal{T}$ indicates the discrete time horizon of the examined traffic data, and F denotes the types of data observed, such as speed and flow. Notably, the traffic datasets used in subsequent experiments are univariate (i.e., $F = 1$); thus, we omit this dimension in subsequent discussions for clarity.

3.2 Problem Formulation

Definition 3 (Traffic Data Representation Learning)**.** The objective of representation learning in traffic data analysis is to identify low-dimensional representations for the time-series data of each traffic sensor and for the graph-structured data encompassing all sensors at any given time. Specifically, an embedding function $\omega : \mathbb{R}^T \to \mathbb{R}^D$ maps the past T traffic observations at node $i \in \mathcal{V}$ into a D-dimensional vector in a compact latent space, where D is significantly less than $|\mathcal{V}|\times|\mathcal{T}|$, optimizing for computational efficiency and model simplicity. Another embedding function $\vartheta : \mathbb{R}^{|\mathcal{V}|} \to \mathbb{R}^D$ translates all traffic observations across $\mathcal{V}$ at any arbitrary point in time into a similarly concise D-dimensional embedding. These two embedding functions are crafted to encapsulate the most informative and compact features of both traffic time-series and graph data.

Definition 4 (Traffic Prediction)**.** Traffic prediction aims to forecast H-dimensional future traffic conditions based on $\mathbf{X}$ by projecting the next L time steps of traffic data across all sensors, denoted by $\hat{\mathbf{X}} \in \mathbb{R}^{|\mathcal{V}|\times L\times H}$. Drawing parallels to $\mathbf{X}$, for the traffic datasets focused on speed and flow, predictions are typically univariate, thus $H = 1$ is maintained throughout subsequent experiments to simplify the notations.

Definition 5 (Traffic Predictor Bootstrapping)**.** Current graph-based traffic predictors leverage spatial-temporal correlations in diverse ways. Traffic predictor bootstrapping centers on utilizing representations derived from the well-trained embedding functions ω and ϑ. These representations replace certain input components and/or internal modules within the predictors, thereby enhancing prediction accuracy and streamlining model training.

4 Contrastive Learning of Spatial-Temporal Traffic Data Representations

In this section, we present CLEAR (**C**ontrastive **L**earning of spatial-t**E**mporal tr**A**ffic data **R**epresentations), a novel learning framework designed to extract rich representations of spatial and temporal traffic data. The design principle of CLEAR revolves around capturing intrinsic spatial and temporal correlations inherent in traffic data, leveraging these learned representations for accurate traffic prediction, and facilitating seamless integration with existing graph learning-based traffic predictors.

4.1 Overview

Fig. 1 presents an overview to the architecture of the proposed CLEAR framework. CLEAR employs self-supervised contrastive learning to extract discriminative embeddings from both traffic time-series and graph data, following the SimCLR contrastive training architecture [15]. By applying composition strategies of weak and strong data augmentation strategies, CLEAR learns robust representations that encode essential information regarding traffic flow, congestion patterns, and anomaly detection, enabling comprehensive analysis and modeling of transportation systems. For time-series representation, CLEAR utilizes a series of Transformer encoders to capture long-range dependencies and develop latent representations. Similarly, for spatial data, CLEAR employs a graph-embedded Transformer encoder architecture to process graph-structured data and extract spatial representations. Both are then fed into a traffic predicting decoder to generate predictions for future time steps.

Further, the CLEAR-learned representations can be leveraged to enhance or replace the spatial and temporal correlation learning components of existing graph-based predictors, thus seamlessly integrated with these predictors for performance improvements. By incorporating these representations, existing predictors can benefit from the rich semantic information captured through self-supervised contrastive learning, thereby enhancing their ability to model complex spatial and temporal dependencies within traffic data. The training of integrated models becomes more straightforward as CLEAR eliminates the need for additional representation extraction steps during model training. Note that when integrating CLEAR with existing predictors, only the Transformer-powered time-series and graph data encoders are adopted. The contrastive learning components, namely, the data augmentation strategies and the contrastive loss, are used to train the encoder parameters.

In the following sub-sections, we delve into the methodology of CLEAR, detailing the process of representation learning (Sections 4.2 and 4.3), its direct application in traffic prediction (Section 4.4), and its integration with existing traffic predictors equipped with various spatial-temporal correlation mining strategies (Section 4.5).

4.2 Traffic Time-series Representation Learning

Understanding the temporal dynamics within traffic data is essential for its comprehensive analysis and effective modeling. These temporal dynamics encapsulate crucial information regarding the evolution of traffic flow, congestion patterns, and anomaly detection, which are fundamental for devising efficient transportation strategies and intelligent traffic management systems. Additionally, each traffic sensor possesses a time-series, which, in the context of transportation graph $\mathcal{G}$, can be regarded as its nodal data feature. Node representations can be thereupon learnt from the corresponding time-series by adopting a time-series encoder $f^{\mathrm{T}}(\cdot)$, and subsequently used to explore the spatial correlation among nodes across the graph.

CLEAR utilizes self-supervised contrastive learning to construct the discriminative embedding space for traffic time-series. Contrastive learning operates on the principle of maximizing agreement between similar instances (positive

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536009

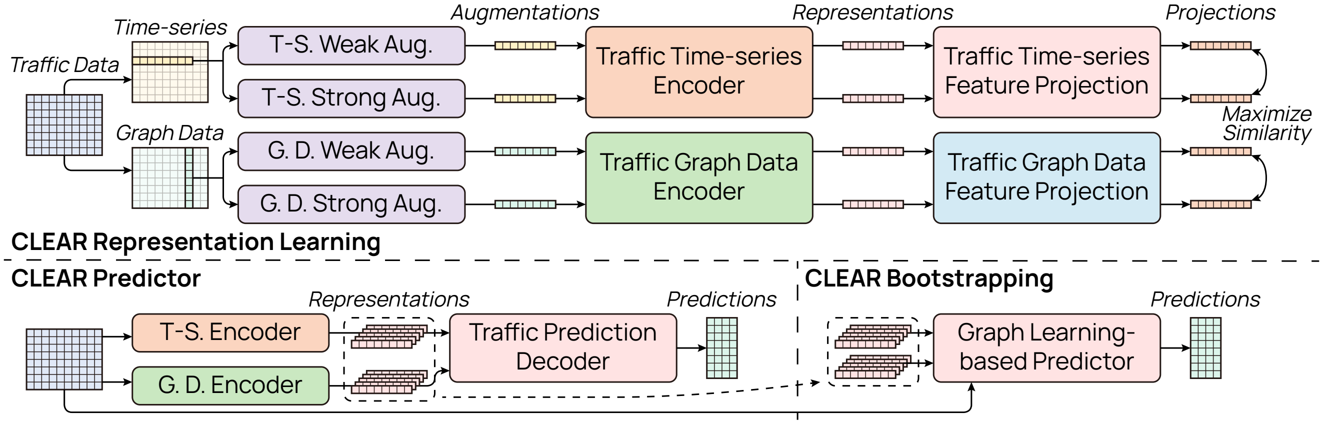


Fig. 1. Overall architecture of the proposed CLEAR framework. In the illustration, "T-S." and "G. D." denotes "time-series" and "graph data", respectively. "Feat. Proj." means "Feature Projection".

samples) while minimizing agreement between dissimilar ones (negative samples) within a latent space. At an arbitrary time t, each node $i \in \mathcal{V}$ has a historical traffic data feature $\mathbf{x}_{i,t} = \{x_{i,t}, x_{i,t-1}, \ldots, x_{i,t-T+1}\} \in \mathbb{R}^T$. We employ the approach of applying weak and strong augmentations to this original time-series and perform cross-view contrastive learning to learn robust nodal representations. Particularly, the weak augmentation is achieved by a moving-average-and-jitter strategy, where the moving average of length M is calculated on $\mathbf{x}_{i,t}$ and the result is further perturbed by Gaussian noise $\mathcal{N}(0, \sigma^2)$:

$$\tilde{\mathbf{x}}^{\mathrm{w}}_{i,t} = \mathrm{MA}(\mathbf{x}_{i,t}; M) + \mathcal{N}(0, \sigma^2), \tag{1}$$

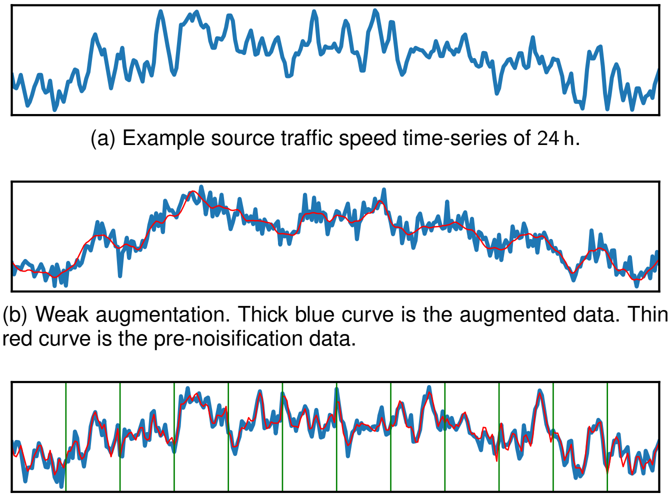
where $\mathrm{MA}(\cdot; M)$ is the moving-average function with padding. Further, the strong augmentation is done by a probablistic-swap-and-jitter strategy for a more significant perturbation to the time-series. We first divide the original time-series $\mathbf{x}_{i,t}$ into chunks $\mathbf{x}_{i,t}[c]$ of length C. Then, we randomly select half of all chunks to form a set $\mathcal{C}$ and apply a random permutation function $\pi(\cdot)$ over the set. The result is subsequently perturbed by Gaussian noise $\mathcal{N}(0, \sigma^2)$ to construct the strongly augmented time-series:

$$\tilde{\mathbf{x}}^{\mathrm{s}}_{i,t}[c] = \begin{cases} \mathbf{x}_{i,t}[\pi(c)] + \mathcal{N}(0, \sigma^2) & \text{if } c \in \mathcal{C} \\ \mathbf{x}_{i,t}[c] + \mathcal{N}(0, \sigma^2) & \text{otherwise} \end{cases}. \tag{2}$$

As empirically demonstrated in [15], applying compositions of data augmentation operations (principle of Eqs. (1) and (2)) is critical for effective representation learning. Both augmented time-series are then passed to the time-series encoder f^{T} to calculate their latent space embeddings.

As depicted in Fig. 3, encoder f^{T} of CLEAR utilizes the Transformer encoder architecture to capture long-range time-series dependencies from the augmented data. f^{T} starts with an input 1×1 convolution layer to first project the traffic time-series, typically univariate or in low dimensions, into a higher-dimensional context space, which is subsequently superimposed with a learnable positional encoding $\mathbf{p}$. As the semantics of time-series are generally more straightforward than languages for which Transformer was originally designed for, we apply three Transformer encoders over the projected input:

$$\mathbf{h}^{(0)}_{i,t} = \mathrm{Conv}^{1\times1}(\tilde{\mathbf{x}}_{i,t}) + \mathbf{p}, \tag{3a}$$



(a) Example source traffic speed time-series of 24 h.

(b) Weak augmentation. Thick blue curve is the augmented data. Thin red curve is the pre-noisification data.

(c) Strong augmentation. Thick blue curve is the augmented data. Thin red curve is the pre-noisification data. Thin green segments are the time-series chunks.

Fig. 2. An example of the source and augmented time-series.

$$\tilde{\mathbf{h}}^{(l)}_{i,t} = \mathrm{MHA}\left(\mathrm{BN}(\mathbf{h}^{(l-1)}_{i,t})\right) + \mathbf{h}^{(l-1)}_{i,t}, \quad 1 \le l \le 3, \tag{3b}$$

$$\mathbf{h}^{(l)}_{i,t} = \mathrm{MLP}\left(\mathrm{BN}(\tilde{\mathbf{h}}^{(l)}_{i,t})\right) + \tilde{\mathbf{h}}^{(l)}_{i,t}, \quad 1 \le l \le 3, \tag{3c}$$

$$\mathbf{h}_{i,t} = \mathrm{FC}\left(\mathrm{MeanPool}(\mathbf{h}^{(3)}_{i,t})\right), \tag{3d}$$

where $\mathrm{Conv}^{1\times1}(\cdot)$ is the 1×1 convolution, $\mathrm{MHA}(\cdot)$ is the multi-headed self-attention[1], $\mathrm{MLP}(\cdot)$ is the multi-layer perceptron with two fully-connected layers of 2048 neurons and a non-linear ReLU activation function, $\mathrm{BN}(\cdot)$ is the batch normalization operation, $\mathrm{FC}(\cdot)$ is the fully-connected layer, and $\mathrm{MeanPool}(\cdot)$ is the mean pooling operation, respectively. Notably, a learnable positional encoding is adopted over the canonical sinosoidal ones in favor of the former's capability in maintaining time-series' periodic feature and the better experimental performance to be demonstrated in Section 5.5. Additionally, pre-norm residual links are adopted to produce more stable gradient values during model training [32]. The final $\mathbf{h}_{i,t} \in \mathbb{R}^D$ is the learnt D-

[1] Following [31], we adopt eight heads in each multi-headed self-attention calculation.

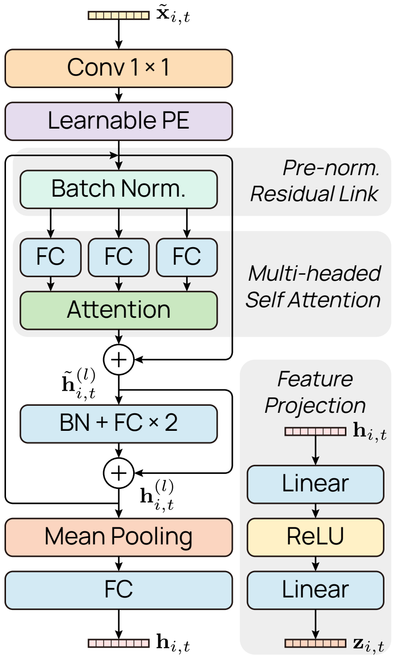


Fig. 3. CLEAR time-series encoder.

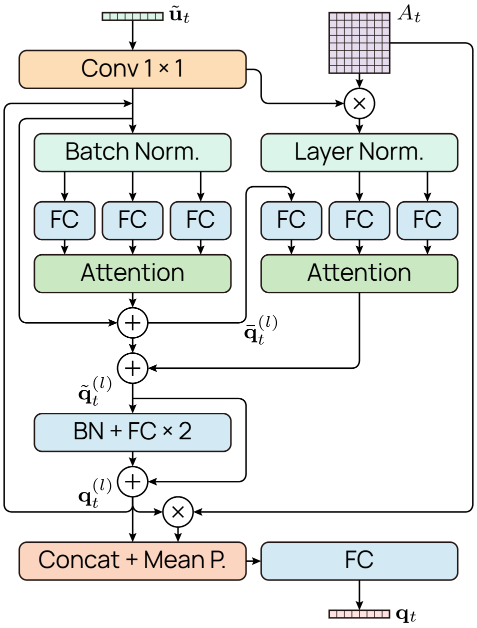


Fig. 4. CLEAR graph data encoder. The feature projection shares the same structure as that in Fig. 3 with the linear transformations replaced by graph convolutions

dimensional representation of the input augmented time-series $\tilde{\mathbf{x}}_{i,t} \in \{\tilde{\mathbf{x}}^{\mathrm{w}}_{i,t}, \tilde{\mathbf{x}}^{\mathrm{s}}_{i,t}\}$.

Given an arbitrary t, there are in total $|\mathcal{V}|$ number of $\mathbf{x}_{i,t}$ samples in the dataset, leading to $2 \times |\mathcal{V}|$ augmented time-series, weakly or strongly done. We use $f^{\mathrm{T}}(\tilde{\mathbf{x}}^{\mathrm{w}}_{i,t})$ and $f^{\mathrm{T}}(\tilde{\mathbf{x}}^{\mathrm{s}}_{i,t})$ as a pair of positive samples, and the other augmented time-series as negative samples. In order to further preserve the semantic information in representations developed from f^{T}, we concatenate the encoder with a concluding time-series feature projection head $g^{\mathrm{T}}(\cdot)$ before training the model with contrastive loss, which can be accordingly formulated as follows:

$$\ell^{\mathrm{T}} = \sum_{i\in\mathcal{V}}\sum_{t\in\mathcal{T}}\Bigg(-\log\frac{\exp\big(\mathrm{sim}(\mathbf{z}^{\mathrm{w}}_{i,t},\mathbf{z}^{\mathrm{s}}_{i,t})/\tau\big)}{\sum_{k\in\mathcal{V}}^{k\neq i}\exp\big(\mathrm{sim}(\mathbf{z}^{\mathrm{w}}_{i,t},\mathbf{z}^{\mathrm{s}}_{k,t})/\tau\big)} - \log\frac{\exp\big(\mathrm{sim}(\mathbf{z}^{\mathrm{s}}_{i,t},\mathbf{z}^{\mathrm{w}}_{i,t})/\tau\big)}{\sum_{k\in\mathcal{V}}^{k\neq i}\exp\big(\mathrm{sim}(\mathbf{z}^{\mathrm{s}}_{i,t},\mathbf{z}^{\mathrm{w}}_{k,t})/\tau\big)}\Bigg), \quad (4)$$

where $\mathrm{sim}(\mathbf{a},\mathbf{b}) = \mathbf{a}^{\mathsf{T}}\mathbf{b}/\|\mathbf{a}\|\cdot\|\mathbf{b}\|$ is the cosine similarity calculation, $\mathbf{z}_{i,t} = g^{\mathrm{T}}(\mathbf{h}_{i,t}) = \mathbf{W}^{(2)}\,\mathrm{ReLU}(\mathbf{W}^{(1)}\mathbf{h}_{i,t})$, and τ denotes a temperature parameter. Principally, each weakly augmented time-series is contrastively tested against all strongly augmented time-series, and vice versa. As the contrastive loss in Eq. (4) forces the model to learn data transformation-agnostic projections $\mathbf{z}_{i,t}$, information useful for downstream traffic tasks may be removed in the process [15]. The perceptron feature projection head $g^{\mathrm{T}}(\cdot)$ is introduced to isolate this transformation-invariant training step and maintain more context in $\mathbf{h}_{i,t}$. This hypothesis is empirically verified in Section 5.5 by testing the performance of including $\mathbf{h}_{i,t}$ in Eq. (4), stand-alone or jointly.

4.3 Traffic Graph Data Representation Learning

The aforementioned time-series encoder $f^{\mathrm{T}}(\cdot)$ is not sufficient in identifying the temporal correlation within traffic data. Rather, it essentially embeds time-series corresponding to traffic nodes in transportation networks, and the result indeed illustrates the inter-nodal dependency, i.e., spatial correlation. The missed is a graph data encoder $f^{\mathrm{G}}(\cdot)$ that takes the traffic data at an arbitrary time to compute their embeddings, so that multiple time instances can develop their correlation with these numerical representations. Further, such embeddings intrinsically capture the spatial dependency among traffic nodes in the condensed representation without confined by explicit geographical adjacency information, thereby dynamically exploits the spatial correlation for traffic prediction. In CLEAR, we employ a contrastive learning-based representation learning paradigm to establish the graph data encoder, aiming at projecting traffic graph data into a semantic-rich embedding space.

Following the same contrastive sample augmentation principle of spatial data representation learning, traffic data feature $\mathbf{u}_t = \{x_{i,t} \mid \forall i \in \mathcal{V}\} \in \mathbb{R}^{|\mathcal{V}|}$ at an arbitrary time t is augmented by a periodic-average-and-jitter strategy as the weak one and a one-hop-average-and-jitter strategy as the strong one. Particularly, the former first calculates the average traffic data at t and that of the same time-of-day and day-of-week in the last week (denoted by $t - 1\mathrm{wk}$), with a further Gaussian noise perturbation:

$$\tilde{\mathbf{u}}^{\mathrm{w}}_t = (\mathbf{u}_t + \mathbf{u}_{t-1\mathrm{wk}})/2 + \mathcal{N}(0,\sigma^2). \quad (5)$$

The latter aggregates the values of one-hop neighboring nodes for any arbitrary node, and use the Gaussian noise-perturbed average value as the strongly augmented data:

$$\tilde{\mathbf{u}}^{\mathrm{s}}_t = \{u^{\mathrm{s}}_{i,t} \mid \forall i \in \mathcal{V}\}, \quad (6a)$$

$$u^{\mathrm{s}}_{i,t} = \sum_{j\in\mathcal{V}^1(i;A_t)} x_{j,t}/\big|\mathcal{V}^1_t(i;A_t)\big| + \mathcal{N}(0,\sigma^2), \quad (6b)$$

where $\mathcal{V}_t^1(\cdot; A_t)$ is the self-containing one-hop neighboring function of an input node on the graph according to the adjacency defined by $A_t \in \mathbb{B}^{|\mathcal{V}|\times|\mathcal{V}|}$. When augmenting $\mathbf{u}_t$, we propose to employ the previous spatial data representations after feature projection $\mathbf{z}_{i,t}$ to construct preliminary nodal adjacency as follows

$$a_{t,ij} = \mathrm{sim}(g^{\mathrm{T}}(f^{\mathrm{T}}(\mathbf{x}_{i,t})), g^{\mathrm{T}}(f^{\mathrm{T}}(\mathbf{x}_{j,t}))), \tag{7a}$$

$$A_t[i,j] = \begin{cases} 1 & \text{if } a_{t,ij} \text{ is among top-}\lambda \text{ in } \{a_{t,ik} \mid \forall k \in \mathcal{V}\} \\ 0 & \text{otherwise} \end{cases}, \tag{7b}$$

instead of using geographical adjacency as defined in $\mathcal{E}$. Hypothetically, the projected representation embeds more semantic information about traffic dynamics than the road network connectivity and facilitates better data augmentation on $\mathbf{u}_t$. We present empirical studies in Section 5.5 supporting this claim.

With the two augmented graph data samples ($\tilde{\mathbf{u}}_t^{\mathrm{w}}$ and $\tilde{\mathbf{u}}_t^{\mathrm{s}}$ with adjacency A_t), we further adopt a variant of Transformer architecture to process the graph-structured data, depicted in Fig. 4. Three encoder layers are adopted after an input 1×1 convolution layer for feature projection:

$$\mathbf{q}_t^{(0)} = \mathrm{Conv}^{1\times 1}(\tilde{\mathbf{u}}_t), \tag{8a}$$

$$\dot{\mathbf{q}}_t^{(l)} = \mathrm{BN}(\mathbf{q}_t^{(l-1)}), \quad \ddot{\mathbf{q}}_t^{(l)} = \mathrm{BN}(A_t\mathbf{q}_t^{(l-1)}), \tag{8b}$$

$$\bar{\mathbf{q}}_t^{(l)} = \mathrm{MHA}(\dot{\mathbf{q}}_t^{(l)}) + \dot{\mathbf{q}}_t^{(l)}, \quad 1 \le l \le 3, \tag{8c}$$

$$\tilde{\mathbf{q}}_t^{(l)} = \mathrm{MHA}(\bar{\mathbf{q}}_t^{(l)}, \ddot{\mathbf{q}}_t^{(l)}, \ddot{\mathbf{q}}_t^{(l)}) + \bar{\mathbf{q}}_t^{(l)}, \quad 1 \le l \le 3, \tag{8d}$$

$$\mathbf{q}_t^{(l)} = \mathrm{MLP}\left(\mathrm{BN}(\tilde{\mathbf{q}}_t^{(l)})\right) + \tilde{\mathbf{q}}_t^{(l)}, \quad 1 \le l \le 3, \tag{8e}$$

$$\mathbf{q}_t = \mathrm{FC}\left(\mathrm{MeanPool}(\mathbf{q}_t^{(3)} \| A_t\mathbf{q}_t^{(3)})\right), \tag{8f}$$

where we abuse the notation of $\mathrm{MHA}(\cdot,\cdot,\cdot)$ to use the first input as the query of multi-head self-attention and the latter two as the key and value [31], respectively. Comparing Eq. (8) with the standard Transformer encoder in Eq. (3), we highlight the introduction of an additional inter-layer attention in Eq. (8d). The model incorporates both intra-level attention Eq. (8c) and inter-level attention Eq. (8d) mechanisms to effectively capture dependencies and relationships within and across different levels of abstraction in the input graph-structured. Intra-level attention facilitates the exchange of information between nodes within the same level of the graph hierarchy, enabling nodes to update their embeddings based on their local neighborhoods. On the other hand, inter-level attention allows nodes to exchange information with neighboring nodes across different levels of abstraction, facilitating the understanding of the global graph structure. After the stacking encoders, the resulting output $\mathbf{q}_t^{(3)}$ is passed to a two-layer graph pooling MLP to generate the embedding. The model thereby efficiently processes and learns complex relationships within the input data, ultimately enabling it to generate rich representations. The final $\mathbf{q}_t \in \mathbb{R}^D$ is the learnt representation of the input augmented graph data $\tilde{\mathbf{u}}_t \in \{\tilde{\mathbf{u}}_t^{\mathrm{w}}, \tilde{\mathbf{u}}_t^{\mathrm{s}}\}$.

The contrastive loss of $f^{\mathrm{G}}(\cdot)$ follows a similar principle of $f^{\mathrm{T}}(\cdot)$. Given a traffic dataset, there are $|\mathcal{T}|$ graph data samples $\mathcal{U} = \{\mathbf{u}_t \mid t \in \mathcal{T}\}$. The loss is accordingly defined as (c.f. Eq. (4)):

$$\ell^{\mathrm{G}} = \sum_{t\in\mathcal{T}} \Bigg(-\log \frac{\exp\left(\mathrm{sim}(\mathbf{v}_t^{\mathrm{w}}, \mathbf{v}_t^{\mathrm{s}})/\tau\right)}{\sum_{r\in\mathcal{U}}^{r\neq t} \exp\left(\mathrm{sim}(\mathbf{v}_t^{\mathrm{w}}, \mathbf{v}_r^{\mathrm{s}})/\tau\right)} - \log \frac{\exp\left(\mathrm{sim}(\mathbf{v}_t^{\mathrm{s}}, \mathbf{v}_t^{\mathrm{w}})/\tau\right)}{\sum_{r\in\mathcal{U}}^{r\neq t} \exp\left(\mathrm{sim}(\mathbf{v}_t^{\mathrm{s}}, \mathbf{v}_r^{\mathrm{w}})/\tau\right)} \Bigg), \tag{9}$$

where $\mathbf{v}_t = g^{\mathrm{G}}(\mathbf{q}_t)$ is the graph feature projection head with two sequential graph convolution operations and a nonlinear ReLU in-between. By this design, CLEAR is capable of adaptively capturing the dynamic spatial correlations within traffic graph slices independently across time steps. This flexibility allows the spatial correlation to be different over time and relaxes the reliance on a fixed or predefined spatial structure. Furthermore, the framework's ability to autonomously learn spatial relationships from data aligns with the concept of adaptability seen in advanced traffic prediction models like Graph WaveNet.

4.4 Traffic Prediction with Traffic Data Representations

In the previous sub-sections, we introduced two self-supervised representation learning models for generating task-agnostic traffic data embeddings. These representations can be utilized in downstream traffic data mining tasks, where traffic prediction is among the most prominent ones. The objective of the task is to develop a prediction function $f\colon \mathbb{R}^{|\mathcal{V}|\times T} \to \mathbb{R}^{|\mathcal{V}|\times L}$ that takes the historical traffic data as input and develop those for the next L time instances.

Given the original traffic data $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{T}|}$ and $\mathbf{x}_{i,t} \subset \mathbf{X}, \mathbf{q}_t \subset \mathbf{X}$, CLEAR extracts the latent time-series and graph data representations at current time t with data encoders

$$\mathbf{h}_{i,t} = f^{\mathrm{T}}(\mathbf{x}_{i,t}), \quad \mathbf{q}_t = f^{\mathrm{G}}(\mathbf{u}_t), \tag{10}$$

respectively. Subsequently, a traffic predicting decoder $f^{\mathrm{F}}(\cdot)$ develops the predicted values. We follow the principle of Graph Attention neTwork (GAT) [33] and devise to use the nodal relationship defined by $\mathbf{h}_{i,t}$ to aggregate traffic data representations by graph convolution defined as follows[2]:

$$\boldsymbol{\hbar}_i = \sum_{j\in\mathcal{V}^1(i;A_t)} \mathbf{h}_{i,t}^{\mathrm{T}}\mathbf{h}_{j,t}\mathbf{W}(\mathbf{h}_{i,t}\|\mathbf{q}_t), \tag{11}$$

where $\mathbf{W} \in \mathbb{R}^{D'\times 2D}$ is a trainable weight matrix. Following the graph convolution, $f^{\mathrm{F}}(\cdot)$ adopts two convolutional layers with kernel sizes of $1 \times D'$ and 1×1 and a ReLU activation in-between, and the final output channel number is L. After decoding, the output matrix $\hat{\mathbf{X}} \in \mathbb{R}^{|\mathcal{V}|\times L}$ corresponds to the L-step traffic prediction of each node in $\mathcal{V}$. The decoder is trained with mean absolute error between the predicted $\hat{\mathbf{X}}$ and the ground truth:

$$\ell^{\mathrm{F}} = \sum_{i\in\mathcal{V}}\sum_{t\in\mathcal{T}}\sum_{\Delta=1}^{L} |\hat{x}_{i,t+\Delta} - x_{i,t+\Delta}|. \tag{12}$$

[2] The presented GAT-based model is among many possible choices for the predictor. Simpler structures like MLP or more complex ones like Transformer can also be adopted.

4.5 Bootstrapping Existing Graph Learning-based Predictors

In addition to empowering downstream traffic data mining tasks, the learned representations from CLEAR possess versatility beyond standalone prediction models. These embeddings can also integrate with existing graph learning-based traffic predictors, replacing or enhancing their spatial and temporal correlation learning components. The merit lies on a recognition that the integration of graph structure learning and data correlation mining often presents a tightly coupled scenario, where the learning task becomes a complex bilevel optimization problem due to the absence of supervised loss information for graph structure learning. This interplay underscores the importance of devising strategies that effectively leverage learned representations to complement or bootstrap existing graph-based predictors, decoupling the adjacency matrix learning from the spatial-temporal data prediction task. The former is outsourced and preprocessed by the prepositional representation learning, so that the predictors can focus on training the forecast model without concerning on another learning task.

Current graph learning-based predictors can be broadly categorized into four patterns based on how spatial-temporal correlation is defined or mined. By examining these patterns, we can effectively illustrate how the representations learned by CLEAR can be integrated with and enhance existing predictors. Each category represents a distinct strategy for capturing and exploiting spatial-temporal correlations within traffic data, offering unique opportunities for integration with learned representations.

Category 1 (C1): Static adjacency matrix based on geo-adjacency. Using a statically defined adjacency matrix based on nodal distance for $\mathcal{G}$ is arguably the most straightforward approach for integrating domain knowledge on the spatial correlation. Example usages are presented in [34], [35]. For this pattern, we can directly replace the original adjacency matrix, commonly denoted by $\mathbf{A}$, with A_t as defined in Eq. (7) to score an intuitive-yet-effective performance improvement:

$$\mathbf{A} \leftarrow A_t, \tag{13}$$

where the current timestamp of prediction is adopted as the t in A_t.

C2: Adaptive adjacency matrix based on representation learning. Another commonly adopted strategy is to employ two learnable node embedding matrices (e.g., $\mathbf{E}_1$ and $\mathbf{E}_2$ as in [10]), whose softmax-ed multiplication $\mathrm{softmax}(\mathrm{ReLU}(\mathbf{E}_1\mathbf{E}_2^\mathsf{T}))$ is considered as the spatial dependency weights $\tilde{\mathbf{A}}_{\text{adp}}$ between pairs of nodes. For this pattern, we can also intuitively replace the learnable weights by the time-series representations $\mathbf{h}_{i,t}$ learnt by CLEAR:

$$\tilde{\mathbf{A}}_{\text{adp}} \leftarrow \mathrm{softmax}(\mathrm{ReLU}(\mathbf{H}_t\mathbf{H}_t^\mathsf{T})), \tag{14}$$

where matrix $\mathbf{H}_t \in \mathbb{R}^{|\mathcal{V}|\times D}$ is constructed by stacking $\mathbf{h}_{i,t}, \forall i \in \mathcal{V}$ as rows.

C3: Spatial and temporal attention. Besides using an adjacency matrix to capture the spatial data correlation, attention mechanism is another approach to represent the data dependency within either the spatial domain or the temporal domain, or both. Typically, the attention values are still derived from trainable nodal or temporal representations, e.g., in [36], [37], which can be substituted by CLEAR-generated ones:

$$\boldsymbol{\alpha} = \{\alpha_{i,j}\} = \mathrm{softmax}(\mathrm{ReLU}((\mathbf{H}_t\mathbf{W}_1)(\mathbf{H}_t\mathbf{W}_2)^\mathsf{T})), \tag{15a}$$

$$\boldsymbol{\beta} = \{\beta_{t,\tau}\} = \mathrm{softmax}(\mathrm{ReLU}((\mathbf{Q}\mathbf{W}_3)(\mathbf{Q}\mathbf{W}_4)^\mathsf{T})), \tag{15b}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the attention matrices between nodes and between timestamps, respectively, and matrix $\mathbf{Q} \in \mathbb{R}^{|\mathcal{T}|\times D}$ is constructed by stacking all $\mathbf{q}_t$ of input timestamps as rows.

C4: Spatial and temporal representation learning. There are also graph learning-based traffic predictors that comprehensively utilize learnt spatial and temporal representations for prediction, [23], [38] for examples. For such models, we may directly utilize $\mathbf{h}_{i,t}$ as the representation for node i, $\mathbf{q}_t$ as the representation for time t, and $\mathbf{h}_{i,t}\|\mathbf{q}_t$ if the representation for node i at time t is required, respectively. One thing that may sounds counter-intuitive is that we uses "time-series representation" $\mathbf{h}_{i,t}$ for node embedding and vice versa, yet the principle is grounded on the fact that each node i corresponds to a $\mathbf{h}_{i,t}$ at an arbitrary time t, making it effectively representing the semantic context of a node, i.e., node representation. The same applies for graph data representation $\mathbf{q}_t$, which encapsulates context of all nodes in the graph at time t into a dense numerical representation.

While the aforementioned bootstrapping methods apply to a wide range of graph learning-based traffic predictors, it is essential to acknowledge that these patterns may not exhaustively capture all strategies employed in the literature. However, their underlying principles provide a foundation upon which both existing and future graph learning-based traffic predictors can be built. Each pattern offers unique opportunities for leveraging learned representations to enhance predictor performance, demonstrating the adaptability and versatility of the CLEAR framework in traffic data analysis and prediction.

5 Experiments

In this section, we present a series of comprehensive experiments on four real-world datasets to show the effectiveness of CLEAR on traffic prediction and bootstrapping other graph learning-based predictors. We first introduce the experimental configurations, including the datasets, baseline methods, performance evaluation metrics, and implementation details of CLEAR. Subsequently, we answer the following research questions (RQs) by demonstrating and discussing the simulation results:

- **RQ1**: Can the traffic data representations learnt by CLEAR provide outstanding traffic predicting accuracy compared to state-of-the-art baselines?
- **RQ2**: Can CLEAR bootstrap existing graph learning-based traffic predictors?
- **RQ3**: How does representations by CLEAR perform in multi-step traffic prediction?
- **RQ4**: How does CLEAR improve model training efficiency?
- **RQ5**: How do the implementation details of CLEAR affect bootstrapping performance?

TABLE 1
Statistical Information of Datasets

Dataset	# Nodes	# Samples	Interval	Year	Span
Beijing	3126	21 600	5 min	2022	2 months
NI-SH	1830	47 495	5 min	2019	5 months
PeMS04	307	16 992	5 min	2018	2 months
METR-LA	207	34 272	5 min	2012	4 months

5.1 Experimental Setup

5.1.1 Datasets

In this work, we conduct experiments on four small-to-large-scale traffic prediction datasets:

- **Beijing** is a traffic speed dataset collected by [39] from the major roads in Beijing, China. The dataset contains speed values of 3126 sensors from May 12, 2022 to July 25, 2022. Traffic speed is recorded every five minutes.
- **NI-SH** is a traffic speed dataset collected by NavInfo from selected roads in Shanghai, China. The dataset contains speed values of 1830 sensors from January 2, 2019 to June 15, 2019. Traffic speed is recorded every five minutes.
- **PeMS04** is a traffic flow dataset collected by California Transportation Agencies Performance Measurement System in the Bay Area of United States. The dataset contains flow volumes of 307 sensors from January 1, 2018 to February 28, 2018. Traffic volume is recorded every five minutes.
- **METR-LA** is a traffic speed dataset collected from the loop-detectors on the Los Angeles County road network of United States. The dataset contains flow volumes of 207 sensors from March 1, 2012 to June 27, 2012. Traffic volume is recorded every five minutes.

All four datasets are collected by different agencies. Their statistical information is summarized in Table 1. For a fair comparison, we adopt the chronological $7/1/2$ split to generate the training, validation, and testing data. Bayesian Gaussian CANDECOMP/PPARAFAC tensor decomposition model [40] is employed to interpolate the missing values in all datasets. Z-score normalization is adopted to improve the model training stability.

5.1.2 Baselines

We select a wealth of state-of-the-art traffic prediction baseline methods in the following experiments:

- **Historical Average (HA)** predicts future traffic based on the historical average traffic volume at each spatial-temporal location.
- **Vector Auto-Regression (VAR)** models capture the dependencies between multiple time series variables, making predictions based on their own lagged values and the lagged values of other variables in the system.
- **Support Vector Regressor (SVR)** utilizes training data to estimate a regression function that generalizes well to unseen data points.
- **Autoregressive Integrated Moving Average (ARIMA)** is widely used for time-series prediction by modeling the relationship between an observation and a number of lagged observations and error terms.
- **ASTGCN** [35] proposes an attribute-augmented spatiotemporal graph convolutional network to enhance spatio-temporal accuracy in predicting traffic by integrating external factors as dynamic and static attributes into the model through an attribute-augmented unit.
- **STSGCN** [34] captures localized spatial-temporal correlations through synchronous modeling mechanisms and accommodating heterogeneities with multiple modules for different time periods.
- **STGODE** [41] captures spatial-temporal dynamics through tensor-based ordinary differential equations, facilitating deeper networks and synchronous utilization of spatial-temporal features, while employing a well-designed temporal dilated convolution structure.
- **Graph WaveNet** [10] introduces a graph neural network architecture designed for spatial-temporal graph modeling, effectively capturing hidden spatial dependencies and handling long sequences with a stacked dilated 1D convolution component.
- **AGCRN** [42] proposes adaptive modules, including a node adaptive parameter learning one and data adaptive graph generation one,to enhance graph convolutional network capabilities for capturing fine-grained spatial and temporal correlations in traffic series automatically.
- **GMAN** [37] utilizes an encoder-decoder architecture with spatio-temporal attention blocks to model the impact of spatio-temporal factors on traffic conditions.
- **STWave+** [36] mitigates distribution shift with a disentangle-fusion framework, employing a dual-channel spatio-temporal network to model trends and events separately, and incorporating self-supervised learning and multi-scale graph wavelet positional encoding for efficient dynamic spatial correlation modeling.
- **STAEFormer** [43] introduces a spatio-temporal adaptive embedding, enhancing vanilla transformers for superior performance in traffic prediction by effectively capturing intrinsic spatio-temporal relations and chronological information in traffic time series.
- **GMSDR** [38] introduces a multi-step dependency relation scheme in recurrent neural networks, seamlessly integrating with graph-based neural networks for spatial-temporal prediction.
- **DGCRN** [22] utilizes hyper-networks to extract dynamic characteristics from node attributes, generating dynamic graphs at each time step and integrating them with predefined static graphs, with an efficient training strategy.

Among the baselines, HA, VAR, SVR, and ARIMA are statistical methods, while all others are state-of-the-art graph learning-based traffic predictors.

5.1.3 Performance Metrics

We evaluate the performance of CLEAR and all baseline methods by three widely-adopted metrics in traffic prediction [10], [23], [36], [37], namely, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

5.1.4 Implementation Details

We set the prediction horizon L to 12. The length of time-series in generating their corresponding representation T

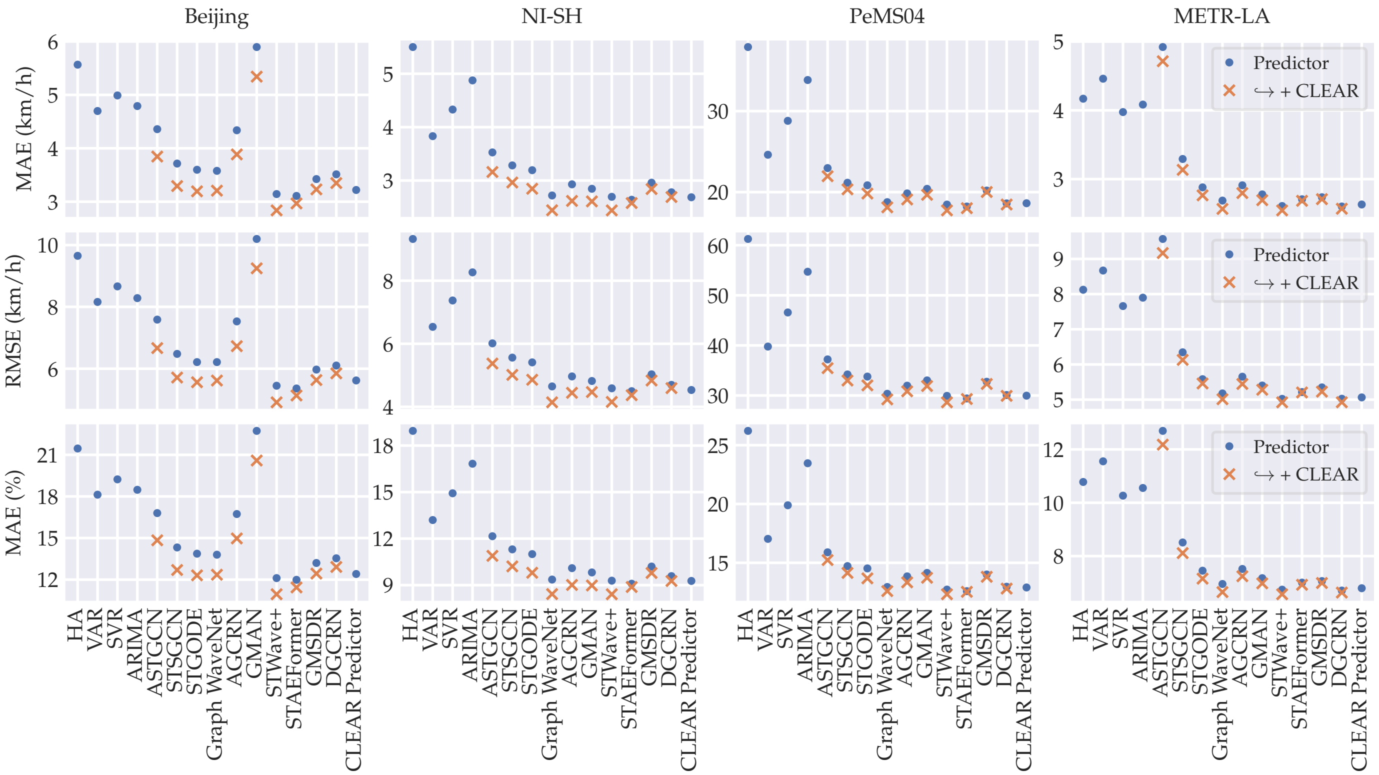


Fig. 5. Performance comparison of baselines, CLEAR-bootstrapped baselines, and the CLEAR Predictor.

TABLE 2
Numerical Results of Best-Performing Baselines, CLEAR-bootstrapped baselines, and the CLEAR Predictor

Method	Beijing			NI-SH			PeMS04			METR-LA		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Graph WaveNet	3.578	6.213	13.798%	2.720	4.648	9.360%	18.759	30.341	12.943%	2.688	5.176	6.954%
↪ + CLEAR	3.207	5.617	12.353%	2.443	4.144	8.424%	18.118	29.219	12.594%	2.567	5.012	6.650%
STWave+	3.143	5.449	12.110%	2.695	4.588	9.288%	18.457	29.925	12.722%	2.609	5.022	6.740%
↪ + CLEAR	2.837	4.912	10.945%	2.440	4.156	8.413%	17.740	28.636	12.327%	2.545	4.923	6.569%
STAEFormer	3.108	5.360	11.985%	2.640	4.498	9.091%	18.215	29.410	12.589%	2.707	5.215	7.004%
↪ + CLEAR	2.968	5.133	11.437%	2.578	4.370	8.880%	18.015	29.295	12.532%	2.685	5.204	6.923%
DGCRN	3.513	6.102	13.539%	2.780	4.697	9.581%	18.606	30.092	12.965%	2.602	5.025	6.706%
↪ + CLEAR	3.349	5.849	12.914%	2.691	4.594	9.267%	18.466	29.926	12.806%	2.569	4.925	6.631%
CLEAR Predictor	3.220	5.620	12.407%	2.687	4.534	9.271%	18.624	29.965	12.903%	2.631	5.063	6.794%

is set to the number of samples in one day, i.e., 288 at a $5\,\mathrm{min}$ sampling interval. The dimensionality D of learnt traffic data representations ($\mathbf{h}_{i,t}$ and $\mathbf{q}_t$) is 256. The moving-average horizon M in Eq. (1) is set to 12. The probablistic-swap length C in Eq. (2) is set to $2\,\mathrm{h}$. The nodal adjacency threshold λ in Eq. (7) is empirically set to the total number of edges in each dataset's geographical adjacency matrix. The number of attention heads in Transformer encoders is set to 8. We adopted the Adam optimizer [44] with an initial learning rate of 5×10^{-4} and weight decay of 10^{-4}. We applied a mini-batch size of 128. CLEAR and baseline methods are implemented with Python and Pytorch. All experiments are conducted on the Viking cluster provided by the University of York with NVIDIA H100 GPUs.

5.2 Traffic Prediction Performance (RQ1, RQ2)

In this sub-section, we present empirical results on employing the proposed CLEAR for traffic prediction and bootstrapping graph learning-based traffic predictors. Among the baseline approaches, ASTGCN, STSGCN, and STGODE are C1 predictors c.f. Section 4.5, Graph WaveNet and AGCRN are C2 predictors, GMAN and STWave+ are C3 predictors, STAEFormer, GMSDR, and DGCRN are C4 predictors. We test all baselines, their CLEAR-bootstrapped variants (Section 4.5), and the CLEAR predictor (Section 4.4) on all four datasets. The traffic predicting accuracy statistics are presented in Fig. 5 and Table 2. Each graph learning-based baselines has the results from both its original model (denoted by its name) and the corresponding CLEAR-bootstrapped variant, denoted by the following line with the tag "↪ + CLEAR". Fig. 5 presents the comparison of all

TABLE 3
Relative MAE Improvement / Degradation of Adopting CLEAR to Bootstrap Baselines

Method	Cat.	Beijing	NI-SH	PeMS04	METR-LA
ASTGCN	C1	+11.770%	+10.401%	+4.375%	+4.196%
STSGCN	C1	+11.338%	+9.759%	+3.850%	+4.801%
STGODE	C1	+11.214%	+10.887%	+4.905%	+4.026%
Graph WaveNet	C2	+10.368%	+10.185%	+3.419%	+4.506%
AGCRN	C2	+10.426%	+10.617%	+3.787%	+3.877%
GMAN	C3	+9.405%	+8.296%	+3.609%	+3.021%
STWave+	C3	+9.739%	+9.436%	+3.888%	+2.440%
STAEFormer	C4	+4.527%	+2.323%	+1.102%	+0.840%
GMSDR	C4	+5.647%	+4.020%	+0.871%	+1.029%
DGCRN	C4	+4.660%	+3.191%	+0.755%	+1.251%
C1 Average		+11.441%	+10.349%	+4.377%	+4.341%
C2 Average		+10.397%	+10.401%	+3.603%	+4.192%
C3 Average		+9.572%	+8.866%	+3.749%	+2.730%
C4 Average		+4.944%	+3.178%	+0.910%	+1.040%
Overall Average		+8.909%	+7.911%	+3.056%	+2.999%

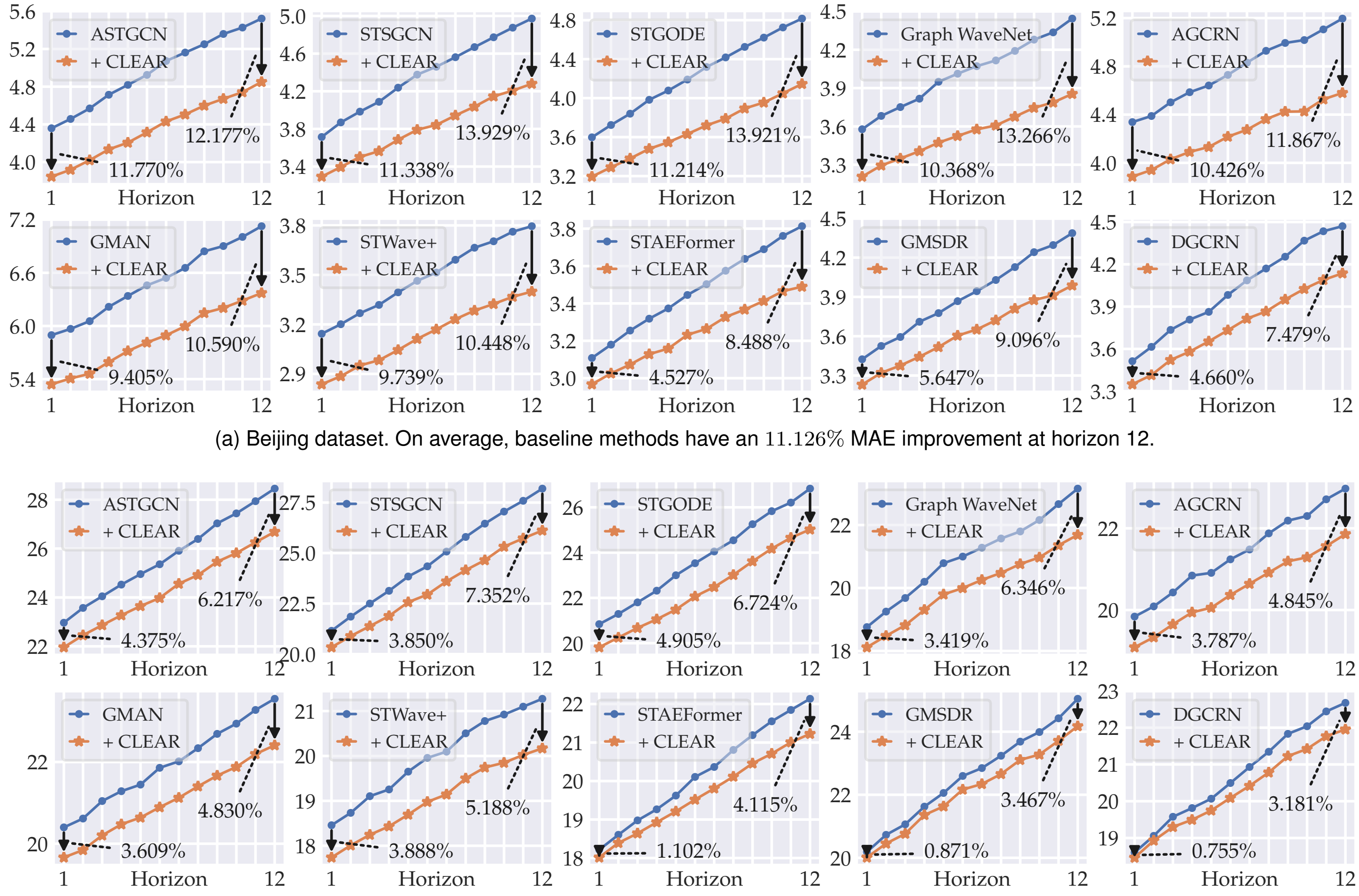
baselines, and their CLEAR-bootstrapped variants, and the CLEAR predictor. Note that HA, VAR, SVR, and ARIMA do not have CLEAR-bootstrapped variants, as they are not graph learning-based predictors. CLEAR predictor cannot be further bootstrapped. Additionally, Table 2 provides the numerical results of the best-performing approaches for a more precise comparison. In the table, top-3 performance from all baselines and CLEAR predictor is underlined.

We first use Fig. 5 and Table 2 to answer RQ1, i.e., does the CLEAR predictor work well in traffic prediction when compared with state of the arts. The simulation results indicate that the CLEAR predictor, even with a quite simplistic convolutional representation decoder, can achieve the 3rd best prediction performance in Beijing dataset, the 2nd in NI-SH, 4th in PeMS04, and 3rd in METR-LA regarding MAE. While STAEFormer develops better predicting results in the first three datasets over CLEAR predictor, the MAE performance gap is not overwhelming: $0.111\,\mathrm{km/h}$ on Beijing, $0.047\,\mathrm{km/h}$ on NI-SH, and $0.408\,\mathrm{vh}$ on PeMS04. Indeed, the best performing baseline on METR-LA, DGCRN, outperforms the CLEAR predictor by $0.029\,\mathrm{km/h}$. We credit the outstanding plain performance of STWave+ and STAEFormer to their effective exploitation of traffic representations, which primarily relies on discrete wavelet transform and Transformer. In this context, the GAT-CNN-CNN structure adopted by CLEAR Predictor c.f. Section 4.4 is far from as with efficacy, resulting in the performance gap. We further tested offline with a straightforward linear projection decoder head in place of the GAT-CNN-CNN structure on Beijing dataset: MAE performance minusculely drops to 3.238 at an approximately 0.56% relative degradation. Consequently, our answer to RQ1 is, though not the best-performing method, the CLEAR predictor can provide outstanding and highly competitive predictions compared with the state-of-the-art baselines.

What makes CLEAR truly stand out is its superior bootstrapping capability. To make the statistics more comprehensible, we summarize the relative MAE improvement/degradation of adopting CLEAR to bootstrap graph learning-based predictors in Table 3. The table indicates that the traffic data representations generated by CLEAR can almost consistently improve the predicting accuracy of all graph learning-based predictor baselines. The improvement is particularly notable on datasets with large and complex transportation networks (Beijing and NI-SH) and baselines with relatively straightforward spatial-temporal correlation mining strategies. Particularly, C1, C2, and C3 predictors experience an average 10.171% improvement on MAE. These results demonstrates the superiority of CLEAR representations on bootstraping graph learning-based traffic predictors for better spatial-temporal data correlation mining on large graphs. The other datasets, namely, PeMS04 and METR-LA have much smaller graphs and respectively noncomplex spatial-temporal correlation. Therefore, existing predictors are more likely to exhaustively exploit such correlation for traffic prediction. Nonetheless, CLEAR can still improve the predicting accuracy by approximately 3.832% for the first three categories. This can be attributed to the more semantic-rich time-variant adjacency matrix (C1), more robust representation-based adjacency matrix (C2), and better semantic-empowered attention scores (C3) developed by CLEAR.

Moving forward to C4-type predictors, we figured that the 2.518% improvements, while still statistically significant with node-wise Wilcoxon signed-rank tests, are not as remarkable as the first two categories. This observation is grounded on the nature of these baselines, which also explicitly extract spatial and temporal traffic data representations by learnable parameters. As a result, well-trained baseline models, in principle, can learn semantic-rich embeddings for the downstream prediction task. In the meantime, the incorporation of CLEAR relieves the training difficulty by decoupling the graph structure learning (i.e., representation learning) from the main data correlation mining during model optimization. We may expect, and get verified in Section 5.4, that the computation burden and training time of respective models can be non-trivially reduced. Datasets with large graphs but less samples can benefit more from CLEAR bootstrapping, as the spatial-temporal correlation mining is more challenging. Alongside being a computationally efficient representation learning substitute for C4 predictors, CLEAR can still obtain better prediction results due to the lighter training difficulty. Therefore, we conclude that CLEAR can bootstrap existing graph learning-based traffic predictors with notable accuracy improvements (RQ2).

Lastly, we adopt the current art of contrastive representation learning frameworks, namely, TS-TCC, TNC, and CoST, to bootstrap the best-performing graph learning-based predictors (STWave+ and STAEFormer). The publicly available source code of respective frameworks is utilized to generate representations for traffic time-series, and the same bootstrapping strategies in Section 4.5 is applied. Simulation results on Beijing dataset are presented in Table 4, indicating that CLEAR outperforms all contrastive representation learning frameworks in enhancing the predicting accuracy of graph learning-based predictors. We further visualize the representations learnt by CLEAR on the Beijing dataset by



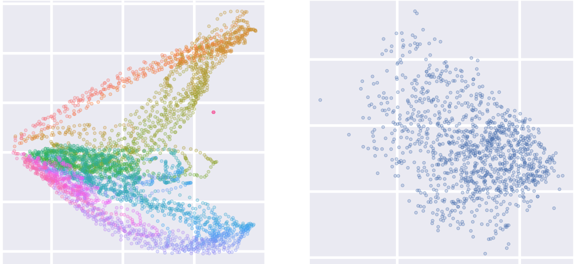
(a) Beijing dataset. On average, baseline methods have an 11.126% MAE improvement at horizon 12.

(b) PeMS04 dataset. On average, baseline methods have a 5.226% MAE improvement at horizon 12.

Fig. 6. Multi-step MAE performance of graph learning-based predictor baselines and their respective CLEAR-bootstrapped variants on the Beijing and PeMS04 datasets.

TABLE 4
Relative Performance Improvement of Contrastive Representation Frameworks on Beijing Dataset

Method		MAE	RMSE	MAPE
STWave+	+ CLEAR	+9.739%	+9.850%	+9.616%
	+ TS-TCC	+3.621%	+3.383%	+3.589%
	+ TNC	+1.664%	+1.229%	+1.638%
	+ CoST	+1.445%	+1.550%	+1.433%
STWave+	+ CLEAR	+4.477%	+4.157%	+4.528%
	+ TS-TCC	−0.290%	−0.166%	−0.261%
	+ TNC	+0.513%	+0.488%	+0.388%
	+ CoST	−0.294%	−0.539%	−0.247%



(a) Graph data representations. Analogous colors are time-of-day-adjacent.

(b) Time-series data representations. Each point refers to a sensor in $\mathcal{V}$.

Fig. 7. Visualization of CLEAR representations by principal component analysis.

principal component analysis in Fig. 7. The graph data representations in Fig. 7(a) are color-coded by the time-of-day, and each point in Fig. 7(b) refers to the traffic dynamics of a sensor. The scatter plots shows that the contrastive representation learning effectively groups similar traffic dynamics and repels opposite ones. We credit the performance of CLEAR to its ability to effectively capture both temporal and spatial dependencies within traffic data, which are essential for accurate traffic prediction but the latter is missing in compared frameworks.

5.3 Multi-Step Prediction Performance (RQ3)

Multi-step traffic prediction also plays a pivotal role in the efficacy of graph learning-based traffic predictors, as it enables insights into traffic trends and patterns over extended periods and allows for better downstream services. Therefore, investigating how representations by CLEAR perform in multi-step traffic prediction scenarios is essential for comprehensively evaluating their effectiveness and applicability in real-world traffic prediction tasks.

In this sub-section, we extend the prediction horizon to 12 steps, i.e., $1\,\mathrm{h}$, on the Beijing and PeMS04 datasets for their distinct network sizes, with all graph learning-based predictor baselines and their respective CLEAR-bootstrapped variants. The simulation results are depicted in Fig. 6, where the horizontal axes denote the predicting horizon from one to twelve and the vertical axes are the MAE values in $\mathrm{km/h}$ (Beijing) and vh, respectively. From the plots, it can be observed that the performance improvement of CLEAR over all baselins are further enlarged with the expanding predicting horizon. On the larger-scale Beijing dataset, the four types of predictors embrace 13.342%, 12.566%, 10.519%, and 8.354% MAE improvements at the 12th predicting horizon, leading to an overall average 11.126% performance boost c.f. 8.909% at the first step. The advancement persists on the smaller PeMS04 dataset with the corresponding type improvements at the 12th horizon to be 6.764%, 5.595%, 5.009%, and 3.588%, overall an 5.226% up from the first step 3.056%. The substantial gains achieved by CLEAR across multiple prediction steps underscore its effectiveness in capturing and leveraging long-term temporal and spatial dependencies within traffic data, affirming its potential to significantly enhance the capabilities of graph learning-based traffic predictors in real-world applications (RQ3).

5.4 Model Training Efficiency (RQ4)

Efficiency in model training is a crucial aspect in the development and deployment of machine learning models, particularly in the context of large-scale traffic prediction tasks. With the ever-increasing volume and complexity of traffic data, the computational resources required for training graph learning-based traffic predictors can become a bottleneck, hindering the scalability and practicality of these models. Therefore, investigating how CLEAR improves model training efficiency is essential for assessing its feasibility and effectiveness in real-world applications.

In this sub-section, we delve into a comprehensive analysis of the size, computational complexity, and training time of both baseline models and their CLEAR-bootstrapped variants. Particularly, we calculate the number of trainable parameters, floating-point operations (FLOPs) for each forward pass calculation on the large-scale Beijing dataset, and measure the relative model training time reduced without and with CLEAR bootstrapping strategies. The empirical results are presented in Table 5. In this table, we also present the relative parameter and FLOPs reduction by introducing CLEAR as well as the relative training time changed besides the raw data. Note that for the relative training time changed column, two values are presented where the former employs available CLEAR representations and the latter includes the CLEAR model training time over the respective base model's.

The simulation result table clearly indicate that CLEAR can effectively reduce the model size (# parameters) and complexity (# FLOPs) for C2, C3, and C4 baselines, thereby improves the model training efficiency. This can be credited to the decoupling of graph structure learning and the main data correlation mining, where the former is pre-achieved by CLEAR. Further, the semantic-rich adjacency matrices employed in C1-type baselines, while do not reduce the model footprint, help the model converge faster with efficacy. When we take the CLEAR model training time into account, Graph WaveNet experience a non-negligible increase in the total training time. Nonetheless, considering its performance gain ($+10.368\%$ c.f. Table 1) and relative short base model training time (approx. $2.6\,\mathrm{h}$), we consider CLEAR still an effective bootstrapping method for Graph WaveNet.

5.5 Ablation Study (RQ5)

In the design of CLEAR, we adopt a few structural designs to improve model capacity and facilitate better performance. Particularly, we employ a learnable positional encoding (Learn PE) for time-series in Eq. (3) substituting the more straightforward sinusoidal encoding, pre-norm residual connections (Pre-N Res.) in Eqs. (3) and (8) instead of the post-norm connection proposed in [31], and feature projection heads (Proj. Head) $g^{\mathrm{T}}(\cdot)$ and $g^{\mathrm{G}}(\cdot)$ following [15]. Finally, we adopt the contrastive learning paradigm to train the representation learning model.

To verify their efficacy in the model performance, we construct a series of CLEAR ablations and test their performance on the Beijing dataset with both the Graph WaveNet base model and the CLEAR predictor. We create a series of CLEAR variants with tags “A” to “G” to denote the ablation of the Learn PE, Pre-N Res. and Proj. Head designs. Additionally, we remove the contrastive learning parts in CLEAR, namely, data augmentation and contrastive loss (replaced by reconstruction loss), to create the ablated model “~~CL~~EAR” with a variant “~~CL~~EAR-PH” that further removes the projection head layers. Note that Beijing dataset is selected for its large-scale traffic network size. Table 6 presents the experimental results. Comparing the performance metrics among different rows, an easy conclusion can be made that all the designs contribute to better CLEAR performance in terms of both traffic prediction and predictor bootstraping, and the contrastive learning paradigm plays a critical part in generating semantic-rich representations. The results generally accord with previous studies on the respective designs, namely, [15], [32], [45]. One minor discrepancy lies in the performance boost by projection heads, which lead to over 10% improvement rather than the approximately 3% in Table 6. We hypothesize that the information loss induced by the contrastive loss, conjectured in [15], is less significant on time-series and graph data than images as originally investigated. Despite this trivial difference, the simulation results verify and agree with the literature that the feature projection heads indeed introduces more robust and semantic-rich data representations for downstream tasks, i.e., traffic prediction in this context.

5.6 Discussion on Limitations

The proposed CLEAR framework demonstrates significant advancements in spatial-temporal traffic data representation learning but is not without limitations. The complexity of the framework, particularly its reliance on dual-branch contrastive learning and Transformer-based encoders, introduces computational overhead that might limit its scalability in resource-constrained environments. Additionally, CLEAR depends on complete input data for representation

TABLE 5
Model Size, Complexity, and Training Time Reduction on Beijing Dataset

Method	Cat.	Original		+ CLEAR		% Reduced		% Training Time Changed	
		# FLOPs	# Params.	# FLOPs	# Params.	# FLOPs	# Params.	Excl. CLEAR†	Incl. CLEAR†
ASTGCN	C1	2.386 T	39.349 M	2.386 T	39.349 M	0.000%	0.000%	−11.921%	−9.386%
STSGCN	C1	961.327 G	89.042 M	961.327 T	89.042 M	0.000%	0.000%	−9.016%	−3.674%
STGODE	C1	193.832 G	813.684 K	193.832 G	813.684 K	0.000%	0.000%	−14.711%	+1.283%
Graph WaveNet	C2	202.407 G	367.748 K	202.387 G	305.228 K	0.010%	17.001%	−33.751%	+11.387%
AGCRN	C2	243.445 G	777.000 K	242.585 G	745.740 K	0.353%	4.023%	−54.944%	−52.363%
GMAN	C3	222.298 G	513.795 K	132.811 G	509.187 K	40.256%	0.895%	−2.076%	−1.501%
STWave+	C3	70.438 G	881.598 K	61.142 G	881.598 K	13.203%	0.000%	−45.610%	−30.242%
STAEFormer	C4	77.174 G	4.061 M	77.174 G	1.060 M	0.000%	73.892%	−6.135%	−1.938%
GMSDR	C4	12.126 G	6.669 M	11.188 G	1.617 M	7.711%	75.712%	−43.292%	−34.570%
DGCRN	C4	1.061 T	432.961 K	1.061 T	182.881 K	0.000%	57.761%	−3.602%	+0.397%

† Excluding / Including the one-time CLEAR training time.

TABLE 6
Ablation Performance of CLEAR on Beijing Dataset

Model	Learn PE	Pre-N Res.	Proj. Head	Graph WaveNet + CLEAR			CLEAR Predictor		
				MAE	RMSE	MAPE	MAE	RMSE	MAPE
CLEAR	✓	✓	✓	3.207	5.617	12.353%	3.220	5.620	12.407%
CLEAR-A	✓	✓	-	3.401	5.900	13.111%	3.447	6.025	13.291%
CLEAR-B	✓	-	✓	3.317	5.763	12.785%	3.379	5.834	13.023%
CLEAR-C	✓	-	-	3.379	5.879	13.022%	3.492	6.053	13.455%
CLEAR-D	-	✓	✓	3.310	5.741	12.765%	3.302	5.763	12.726%
CLEAR-E	-	✓	-	3.446	6.056	13.294%	3.431	5.944	13.208%
CLEAR-F	-	-	✓	3.390	5.896	13.074%	3.317	5.745	12.776%
CLEAR-G	-	-	-	3.410	5.900	13.138%	3.431	5.938	13.225%
~~CL~~EAR	✓	✓	✓	3.561	6.198	13.733%	4.752	8.227	18.299%
~~CL~~EAR-PH	✓	✓	-	3.612	6.249	13.930%	4.845	8.405	18.687%

extraction, which could restrict its applicability to datasets with missing or noisy observations: further data imputation approaches are required as pre-processors and they may undermine the representation learning performance. Further, while CLEAR decouples adjacency matrix learning from spatial-temporal correlation extraction, this approach may not outperform the ideal case of tightly coupled learning in scenarios with abundant computational resources, training data, and carefully tailored models. Though the GAT-based predictor within CLEAR performs competitively, its relatively simple architecture does not outperform certain state-of-the-art predictors, emphasizing CLEAR's role as a representation learning framework over a standalone predictor.

Despite these limitations, CLEAR's ability to enhance the performance of existing graph-based predictors highlights its practicality and potential for impact. By decoupling adjacency matrix learning from spatial-temporal correlation extraction, CLEAR reduces the training complexity of integrated models and offers a flexible, task-agnostic representation learning paradigm. Future work could address these limitations by exploring data imputation mechanisms, simplifying the architecture, and adapting CLEAR for a broader range of traffic analytics task decoders.

6 Conclusions

This paper introduces the CLEAR (Contrastive Learning of spatial-tEmporal trAffic data Representations) framework that leverages the power of self-supervised contrastive learning to extract meaningful embeddings from both traffic time-series and graph-structured data, facilitating more accurate traffic predictions. By employing both weak and strong data augmentation techniques, CLEAR enhances its robustness and generalization in generating semantic-rich data representations within diverse traffic datasets. Its specialized models capture critical temporal and spatial dependencies, allowing seamless integration with existing traffic prediction models, increasing their accuracy and reducing model training complexity without significant modifications.

Experimental evaluations across four real-world datasets validate CLEAR's superior ability to predict future traffic and bootstrap graph learning-based predictors. Multi-step prediction tests and ablation studies confirm its robustness and the strategic efficacy of its design, and a thorough look into the model training efficiency indicate its efficiency in decoupling the graph adjacency learning and data correlation learning processes during training. These experiments substantiate the framework's utility in real-world applica-

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536009

tions, making it a promising approach to be integrated in traffic prediction methods.

Looking ahead, the CLEAR framework opens several avenues for further research. Future work could explore the extension of this framework to other types of downstream intelligent transportation tasks. As the CLEAR framework is designed to be modular and flexible, it can be adapted to other traffic-related tasks with tailor-made representation decoders, somehow similar to the Traffic Prediction Decoder in Fig. 1. Additionally, further refinement of the data augmentation and representation learning techniques could yield even more robust models capable of handling increasingly complex datasets.

References

[1] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, Nov. 2022.

[2] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1544–1561, Apr. 2022.

[3] J. Liu, T. Li, S. Ji, P. Xie, S. Du, F. Teng, and J. Zhang, "Urban flow pattern mining based on multi-source heterogeneous data fusion and knowledge graph embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 2133–2146, Feb. 2023.

[4] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: A probabilistic model fusing multi-source data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1310–1323, Jul. 2018.

[5] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proc. International Conference on Learning Representations*, ser. ICLR '17, Toulon, France, Apr. 2017, pp. 1–14.

[6] S. Rahmani, A. Baghbani, N. Bouguila, and Z. Patterson, "Graph neural networks for intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8846–8885, Aug. 2023.

[7] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting," in *Proc. International Conference on Learning Representations*, ser. ICLR '18, Vancouver, Canada, Apr. 2018, pp. 1–16.

[8] J. J. Q. Yu, "Graph construction for traffic prediction: A data-driven approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 015–15 027, Sep. 2022.

[9] L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, and H. Xiong, "Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting," in *Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Singapore, Aug. 2021, pp. 547–555.

[10] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. International Joint Conference on Artificial Intelligence*, ser. IJCAI '19, Macao, China, Aug. 2019, pp. 1907–1913.

[11] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma, "Explaining neural scaling laws," *Proceedings of the National Academy of Sciences*, vol. 121, no. 27, Jun. 2024.

[12] L. Franceschi, M. Niepert, M. Pontil, and X. He, "Learning discrete structures for graph neural networks," in *Proc. International Conference on Machine Learning*, ser. ICML '19, Long Beach, CA, Jul. 2019, pp. 1–20.

[13] H. Lin, Y. Fan, J. Zhang, and B. Bai, "REST: Reciprocal Framework for Spatiotemporal-coupled Predictions," in *Proc. Web Conference*, ser. WWW '21, Ljubljana, Slovenia, Apr. 2021, pp. 3136–3145.

[14] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR '05, San Diego, CA, Jun. 2005, pp. 539–546.

[15] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. International Conference on Machine Learning*, ser. ICML '20, vol. 119, Virtual, Jul. 2020, pp. 1597–1607.

[16] Y. Gong, T. He, M. Chen, B. Wang, L. Nie, and Y. Yin, "Spatio-temporal enhanced contrastive and contextual learning for weather forecasting," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–16, 2024, early access.

[17] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" in *Proc. Advances in Neural Information Processing Systems*, ser. NeurIPS '20, Vancouver, Canada, Dec. 2020, pp. 6827–6839.

[18] Z. Shao, Z. Zhang, F. Wang, and Y. Xu, "Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting," in *Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22, Washington, DC, Aug. 2022, pp. 1567–1577.

[19] X. Luo, C. Zhu, D. Zhang, and Q. Li, "STG4Traffic: A survey and benchmark of spatial-temporal graph neural networks for traffic prediction," arXiv: 2307.00495, Jul. 2023.

[20] X. Ren, W. Wei, L. Xia, and C. Huang, "A comprehensive survey on self-supervised learning for recommendation," arXiv: 2404.03354, Apr. 2024.

[21] T. Uelwer, J. Robine, S. S. Wagner, M. Höftmann, E. Upschulte, S. Konietzny, M. Behrendt, and S. Harmeling, "A survey on self-supervised representation learning," arXiv: 2308.11455, Aug. 2023.

[22] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 1, pp. 9:1–9:21, Feb. 2023.

[23] M. Liu, H. Huang, H. Feng, L. Sun, B. Du, and Y. Fu, "PriSTI: A conditional diffusion framework for spatiotemporal imputation," in *Proc. IEEE International Conference on Data Engineering*, ser. ICDE '23, Anaheim, CA, Apr. 2023, pp. 1927–1939.

[24] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," arXiv: 1807.03748, Jul. 2018.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ser. CVPR '20, Seattle, WA, Jun. 2020, pp. 9726–9735.

[26] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," in *Proc. International Joint Conference on Artificial Intelligence*, ser. IJCAI '21, Virtual, Aug. 2021, pp. 2352–2359.

[27] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding," in *Proc. International Conference on Learning Representations*, ser. ICLR '21, Virtual, May 2021, pp. 1–17.

[28] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting," in *Proc. International Conference on Learning Representations*, ser. ICLR '22, Virtual, Apr. 2022, pp. 1–17.

[29] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, and Y. Zheng, "Spatio-temporal self-supervised learning for traffic flow prediction," in *Proc. AAAI Conference on Artificial Intelligence*, ser. AAAI '23, vol. 37, Washington, DC, Feb. 2023, pp. 4356–4364.

[30] H. Qu, Y. Gong, M. Chen, J. Zhang, Y. Zheng, and Y. Yin, "Forecasting fine-grained urban flows via spatio-temporal contrastive self-supervision," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8008–8023, Aug. 2023.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, ser. NeurIPS '17, Long Beach, CA, Dec. 2017, pp. 6000–6010.

[32] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T.-Y. Liu, "On layer normalization in the Transformer architecture," in *Proc. International Conference on Machine Learning*, ser. ICML '20, vol. 119, Virtual, Jul. 2020, pp. 10 524–10 533.

[33] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," in *Proc. International Conference on Learning Representations*, ser. ICLR '18, Vancouver, Canada, Apr. 2018, pp. 1–12.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3536009

[34] C. Song, Y. Lin, S. Guo, and H. Wan, “Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting,” in *Proc. AAAI Conference on Artificial Intelligence*, ser. AAAI ’20, vol. 34, New York, NY, Apr. 2020, pp. 914–921.

[35] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, and H. Li, “AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting,” *IEEE Access*, vol. 9, pp. 35 973–35 983, 2021.

[36] Y. Fang, Y. Qin, H. Luo, F. Zhao, B. Xu, L. Zeng, and C. Wang, “When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks,” in *Proc. IEEE International Conference on Data Engineering*, ser. ICDE ’23, Anaheim, CA, Apr. 2023, pp. 517–529.

[37] C. Zheng, X. Fan, C. Wang, and J. Qi, “GMAN: A graph multi-attention network for traffic prediction,” in *Proc. AAAI Conference on Artificial Intelligence*, ser. AAAI ’20, vol. 34, New York, NY, Apr. 2020, pp. 1234–1241.

[38] N. Liu, X. Wang, D. Bo, C. Shi, and J. Pei, “Revisiting graph contrastive learning from the perspective of graph spectrum,” in *Proc. Advances in Neural Information Processing Systems*, ser. NeurIPS ’22, New Orleans, LA, May 2022, pp. 1–12.

[39] Z. Cai, R. Jiang, X. Yang, Z. Wang, D. Guo, H. H. Kobayashi, X. Song, and R. Shibasaki, “MemDA: Forecasting urban time series with memory-based drift adaptation,” in *Proc. ACM International Conference on Information and Knowledge Management*, ser. CIKM ’23, Birmingham, UK, Oct. 2023, pp. 193–202.

[40] X. Chen, Z. He, and L. Sun, “A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation,” *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 73–84, Jan. 2019.

[41] Z. Fang, Q. Long, G. Song, and K. Xie, “Spatial-temporal graph ODE networks for traffic flow forecasting,” in *Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD ’21, Singapore, Aug. 2021, pp. 364–373.

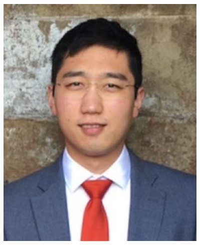
[42] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, “Adaptive graph convolutional recurrent network for traffic forecasting,” in *Proc. Advances in Neural Information Processing Systems*, ser. NeurIPS ’20, Vancouver, Canada, Dec. 2020, pp. 17 804–17 815.

[43] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, “Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting,” in *Proc. ACM International Conference on Information and Knowledge Management*, ser. CIKM ’23, Birmingham, UK, Oct. 2023, pp. 4125–4129.

[44] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proc. International Conference on Learning Representations*, ser. ICLR ’15, San Diego, CA, May 2015, pp. 1–15.

[45] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, “A Transformer-based framework for multivariate time series representation learning,” in *Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD ’21, Singapore, Aug. 2021, pp. 2114–2124.

James Jianqiao Yu (S’11–M’15–SM’20) is a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. He received the B.Eng. and Ph.D. degree in Electrical and Electronic Engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2011 and 2015, respectively. He was a post-doctoral fellow at the University of Hong Kong from 2015 to 2018. He held professorship/lectureship at Southern University of Science and Technology, China and University of York, United Kingdom from 2018 to 2024. His general research interests are in data mining, multi-modal learning, intelligent transportation systems, and embodied artificial intelligence. His work is now mainly on spatial-temporal data mining, multi-modal foundation model, and forecasting and logistics of future transportation systems. He has published over 100 academic papers in top international journals and conferences, and representative papers have been selected as ESI highly cited papers. He was the World’s Top 2% Scientists since 2020 and of career by Stanford University. He is an Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and IET SMART CITIES. He is a Senior Member of IEEE.



Xinwei Fang is a Lecturer with the Department of Computer Science at the University of York, United Kingdom. His research focuses on the design and development of trustworthy autonomous systems by understanding, detecting, and mitigating uncertainties that may arise at various stages of these systems through methods such as machine learning, model checking, and statistical analysis.

Shiyao Zhang (S’18–M’20) received the B.S. degree (Hons.) in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA, in 2014, the M. S. degree in Electrical Engineering from University of Southern California, Los Angeles, CA, USA, in 2016, and the Ph.D. degree from the University of Hong Kong, Hong Kong SAR, China, in 2020. He was a Post-Doctoral Research Fellow with the Academy for Advanced Interdisciplinary Studies, Southern University of Science and Technology from 2020 to 2022, and a Research Assistant Professor with the Research Institute for Trustworthy Autonomous Systems, Southern University of Science and Technology from 2022 to 2024. He is currently an Assistant Professor with the School of Engineering, Great Bay University. His research interests include intelligent transportation systems, autonomous driving, embodied AI, and transportation electrification.

Yuxin Ma is a tenure-track Associate Professor in the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), China. He received B.Eng. and Ph.D. from Zhejiang University, China. Before joining SUSTech, he worked as a Postdoctoral Research Associate in VADER Lab, CIDSE, Arizona State University. His primary research interests are in the areas of visualization and visual analytics, focusing on explainable AI, high-dimensional data, and spatiotemporal data.