

Semantic-Enhanced Representation Learning for Road Networks with Temporal Dynamics

Yile Chen, Xiucheng Li, Gao Cong, Zhifeng Bao, and Cheng Long

Abstract—The widespread adoption of mobile devices and positioning technology has resulted in the generation of massive urban data, offering great opportunities to improve analytical abilities for urban infrastructure components. In this study, we introduce a novel framework called *Toast* for learning general-purpose representations of road networks, along with its advanced counterpart *DyToast*, designed to enhance the integration of temporal dynamics to boost the performance of various time-sensitive downstream tasks. Specifically, we propose to encode two pivotal semantic characteristics intrinsic to road networks: traffic patterns and traveling semantics. To achieve this, we refine the skip-gram module by incorporating auxiliary objectives aimed at predicting the traffic context associated with a target road segment. Moreover, we leverage mobile trajectory data and design pre-training strategies based on Transformer to distill traveling semantics on road networks. *DyToast* further augments this framework by employing unified trigonometric functions characterized by their beneficial properties, enabling the capture of temporal evolution and dynamic nature of road networks more effectively. With these proposed techniques, we can obtain representations that encode multi-faceted aspects of knowledge within road networks, applicable across both road segment-based applications and trajectory-based applications. Extensive experiments on two real-world datasets across three tasks demonstrate that our proposed framework consistently outperforms the state-of-the-art baselines by a significant margin.

Index Terms—Road network representation learning, spatio-temporal data mining, self-supervised learning.

I. INTRODUCTION

WITH the advancement and penetration of mobile devices and positioning technology, a large amount of urban data has been collected to enhance the insights for urban sensing [1]–[4]. Associated with collected vehicle trajectories, road networks serve as fundamental yet indispensable infrastructure in urban spaces, and have been extensively utilized in various downstream analytical tasks, such as traffic forecasting [5], [6] and route inference [7]–[9]. Recent studies have increasingly focused on deriving effective representations that can capture the intrinsic characteristics of road networks. Such general-purpose representations have the potential to significantly enhance the effectiveness of these varied tasks.

Yile Chen, Gao Cong and Cheng Long are with College of Computing and Data Science, Nanyang Technological University, Singapore. E-mail: yile001@e.ntu.edu.sg, {gaocong, c.long}@ntu.edu.sg;

Xiucheng Li is with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Guangdong, China. E-mail: lixiucheng@hit.edu.cn;

Zhifeng Bao is with School of Computing Technologies, RMIT University, Melbourne, Victoria, Australia. E-mail: zhifeng.bao@rmit.edu.au;

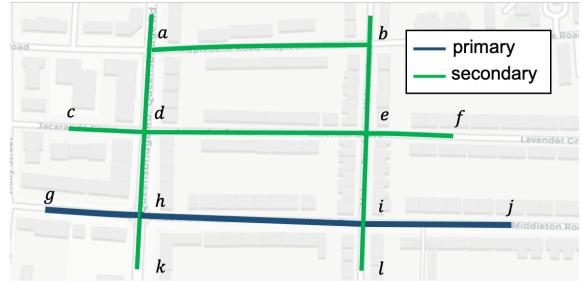


Fig. 1. Road network example. Blue line denotes primary roads and green lines denote secondary roads.

Given that road networks are essentially a graph, a natural question to ask is whether we can apply graph representation learning models to achieve this goal. Unfortunately, the application of such models is non-trivial due to two issues.

The first is the *discrepancies* with regard to the assumptions applied to common graphs and road networks. Most previous graph representation learning methods predominantly target citation graphs [10], [11] or social networks [12], [13], devising techniques grounded in certain well-established assumptions specific to these types of graphs. These assumptions, however, may not be applicable or valid for road networks. For example, a citation graph often exhibits network homophily, where interconnected nodes are more similar than distant nodes. However, this principle does not necessarily translate to road networks, where spatially adjacent road segments might not necessarily exhibit similar traffic patterns. In Fig. 1, road segments dh , gh , hi , hk are connected to each other, but primary roads typically have different *traffic patterns* (e.g., volume) compared to secondary roads since primary roads are travelled more frequently.

The second is the *feature uniformity* issue inherent in road networks. Features such as road type and lane number are often shared across spatially adjacent road segments. This characteristic is more evident in urban sub-regions, such as commercial areas and residential areas, where a large fraction of road networks have the same features within these areas. Such uniformity in road networks can dampen the performance of standard graph representation learning methods, especially graph neural networks (GNN) [14]. An example of this, illustrated in Fig. 1, shows that all road segments connected to the target road segment de possess the same features (road type). In such instances, GNN aggregation process renders these road segments indistinguishable when they present the same feature input [15].

While recent studies have adapted graph representation

learning to road network setting, they still have limitations in addressing the two issues. Methods in [16]–[18] aim to produce representations for road segments and intersections through multi-task learning. They propose to integrate additional classification objectives, such as identifying common attributes between two segments or intersections (e.g., same-way road or stop sign presence). However, these methods heavily rely on the homophily assumption, and thus cannot fully address the first issue. On the other hand, methods in [19]–[21] adapt GNN to road networks for learning road segment representations. However, they particularly suffer from the second issue in areas with uniform road features. Furthermore, these models focus on capturing certain aspects of road network characteristics (refer to Table I for details), and thus fall short in learning effective representations that contain multifaceted knowledge about road networks.

We argue that deriving effective road network representations requires capturing two types of semantic characteristics: namely *traffic patterns* and *traveling semantics*, in order to address the identified issues. Traffic patterns, encompassing factors like traffic volumes, serve as important indicators to enrich the knowledge beyond topological structure, thus overcoming the limitations posed by assumptions for common graphs. Meanwhile, traveling semantics, such as transition patterns, assist in distinguishing road segments that exhibit similar features. As illustrated in Fig. 1, transition patterns can reveal that the path $[c, d, e, f]$ is more frequently traveled compared to the detour path $[c, d, a, b, e, f]$ between segments c and f , thus highlighting the dependencies among road segments. These two types of characteristics represent the most fundamental aspects of road networks. Therefore, their proper encoding and integration are crucial for enriching multi-faced knowledge desired in downstream applications.

To this end, we propose a framework called **Toast** to learn general-purpose representations of road networks that can capture both traffic patterns and traveling semantics with dedicated modules. Apart from encoding the topological structure based on skip-gram training objective [18], [19], our method further enables this module to capture traffic patterns by incorporating an auxiliary traffic context prediction objective. By doing this, the module not only encodes the graph structure of road networks, but also distinguishes connected road segments in terms of traffic patterns, thereby addressing the *discrepancies* issue. To tackle the *feature uniformity* issue, we propose to leverage trajectory data to extract traveling semantics for indistinguishable road network fractions caused by uniform features. Inspired by the success of Transformer-based pre-trained models [22], we employ this architecture to capture transition patterns from trajectory data into representations. Considering the inadequacy of conventional training tasks for text modeling in road network contexts, we design two novel training tasks, *route recovery* and *trajectory discrimination*, tailored to effectively encode the traveling semantics. Both modules are based on self-supervised training paradigms where traffic patterns and traveling semantics are directly treated as training objectives without the need for additional task-specific labels. This ensures that the derived representations are versatile and effective in a range of downstream applications. Moreover,

Toast offers additional advantage of obtaining trajectory representations from the trajectory-enhanced Transformer module. Such capability further enhances the utility for trajectory-based tasks, such as travel time estimation and destination prediction.

The introduction of **Toast** [23] has triggered the development of several subsequent methods [24]–[27] to overcome the previously outlined issues. Specifically, some methods propose to incorporate contrastive learning techniques tailored for road networks, such as spatial-aware sampling [24] and multi-view contrasts between road segments and trajectories [25]. Others adapt GNN to tackle feature uniformity weakness by hypergraph construction [26] or transition pattern integration [27]. These methods have achieved encouraging results. However, they, along with other existing studies, are designed to learn static representations for road networks. In practice, numerous road network tasks are inherently dynamic: traffic speed on road segments varies over time, and travel times for the same route can differ significantly across different time frames. Therefore, there is a growing need to develop time-sensitive road network representations that not only embody better effectiveness but are also more readily applicable to these dynamic tasks.

To achieve this goal, we propose **DyToast**, an improved version of **Toast**, equipped with a unified temporal encoding technique that requires minimum model modifications to the original method. **DyToast** is designed to fuse temporal dynamics into representations by employing learnable trigonometric functions, which exhibit beneficial theoretical properties in road network contexts, into each module. First, apart from refining the original skip-gram objective with traffic patterns in **Toast**, we augment this module by supplementing the road network graph with transition frequencies for each time frame. This enhancement is complemented by the adoption of parameterization for temporal variations to adeptly capture the evolving patterns for a target road segment in relation to its surrounding road segments. Second, we address the challenge of modeling complex temporal correlations in trajectories with irregular time gaps between consecutive road segments. Traditional absolute or relative positional embeddings in Transformer are insufficient for modeling such irregularities. To resolve this, **DyToast** integrates the trigonometric function seamlessly into the self-attention mechanism, thereby effectively capturing such irregular and continuous properties. Through the proposed temporal encoding technique, **DyToast** stands out as a solution for capturing not only the dynamic evolution of road segments in relation to their surrounding environment but also the nuanced, higher-order dependencies inherent in trajectories with irregular time intervals.

To summarize, our contributions are as follows:

- We propose a method named **Toast**, which features with two modules: a traffic context-aware skip-gram module and a trajectory-enhanced Transformer module, to capture traffic patterns and traveling semantics within road networks. **Toast** can derive general-purpose representations for road networks, which are beneficial for both road segment-based and trajectory-based applications.
- Building upon **Toast**, we develop an enhanced version, **DyToast**, which incorporates the ability to capture tem-

TABLE I
COMPARISON OF ROAD NETWORK REPRESENTATION LEARNING METHODS IN TERMS OF MODEL ARCHITECTURE AND THEIR ABILITY TO CAPTURE FOUR TYPES OF ROAD NETWORK CHARACTERISTICS

Methods	Encoder	Topological structure	Traffic patterns	Traveling semantics	Temporal dynamics
node2vec [16]	Skip-gram	✓	✗	✗	✗
SRN2Vec [18]	MLP	✗	✓	✗	✗
HRNR [20]	GNN	✓	✗	✗	✗
RFN [21]	GNN+MLP	✓	✗	✗	✗
SARN [24]	GNN	✓	✗	✗	✗
JCLRNT [25]	Transformer	✓	✗	✓	✗
TrajRNE [27]	GNN+MLP	✓	✓	G	✗
HyperRoad [26]	GNN	✓	T	T	✗
Toast [23]	Skip-gram+Transformer	✓	✓	✓	✗
DyToast	Skip-gram+Transformer	✓	✓	✓	✓

poral dynamics. This is achieved through an innovative integration of learnable trigonometric functions, which align seamlessly with **Toast**. **DyToast** excels at encoding temporally nuanced knowledge in both road network and trajectory contexts, offering a more dynamic understanding of road network patterns.

- We conduct extensive experiments on three time-sensitive applications on road networks. The results show that **Toast** performs comparable to existing methods. Furthermore, with the integration of the proposed temporal encoding technique, **DyToast** demonstrates a significant performance improvement, consistently outperforming baseline methods by a substantial margin.

II. RELATED WORK

A. Representation Learning for Road Networks

Road networks serve as critical components in various intelligent transportation applications, such as traffic inference and forecasting [5], [6], road attribute prediction [28], and travel time estimation [29]. In these applications, road network representations are implicitly learned with supervision signals specific to the task at hand. To achieve more generic applicability, recent efforts have also focused on adapting graph representation learning techniques to road networks, aiming to derive general-purpose representations that can benefit a range of tasks.

Specifically, some studies employ random walk strategies based on the principles of classical Deepwalk [12] and node2vec [13]. They directly apply the method [16], or modify them to include geo-locality and geo-shape information through multi-task learning [18], [19]. Another line of research adapts GNN [14] to road networks. For example, RFN [17], [21] perform extends GNN to perform multi-view relational fusion by aggregating information at both road segment and intersection levels. HRNR [20] adopts a hierarchical GNN approach to model the bottom-up structure of road networks. To address the issues of these methods discussed in Section I, **Toast** proposes to further capture the knowledge of traffic patterns and traveling semantics by integrating a traffic context prediction objective and pioneering trajectory pre-training strategies. Following **Toast**, subsequent studies introduce various methods to model such essential knowledge. HyperRoad [26] implements GNN with hyperedge and hypergraph-based training objectives on hypergraphs

constructed from road networks. **TrajRNE** [27] leverages trajectory data to generate random walks and derive adjacency matrix for GNN, thus combining the studies from two technical branches. JCLRNT [25] and SARN [24] further augment **Toast** with contrastive learning techniques. However, existing methods do not adequately address the dynamic aspect of road network representations, thus resulting in sub-optimal performance for time-sensitive downstream applications. This gap underscores the need for further development in dynamic road network representations, as achieved in **DyToast**.

B. Trajectory Analysis and Modeling

Trajectories, representing the movement of vehicles within a city, are a crucial data source to provide supplementary insights for tasks related to road networks [30]. In particular, road networks explicitly impose structural constraints that govern the traversals of trajectories, forming the foundations for applications such as route planning [8], [31], anomaly detection [32] and destination prediction [33]. Conversely, trajectories provide rich knowledge of traveling semantics for road networks [34], which can effectively enhance tasks that may not necessarily involve trajectory data. For instance, in traffic flow prediction [35] and speed prediction [36], trajectories are employed to guide GNN aggregation processes. They are also utilized to extract transition features for region functionality modeling [37], [38]. In the topic of road network representation learning, **Toast** pioneers the integration of trajectories through pre-training strategies, and therefore produce representations for both road segments and trajectories that are applicable across diverse downstream applications. Following the path of **Toast**, JCLRNT [25] adopts a similar model architecture and applies contrastive learning techniques to also obtain both road segment and trajectory representations. In addition, **TrajRNE** [27] utilizes trajectories to derive the adjacency matrix for GNN for modeling higher-order road segment correlations. However, these methods only exploit the sequential aspect in trajectories while neglecting the temporal dimension. This leads to the limitations of not encoding detailed temporal dependencies in the routes, which are important in time-sensitive applications.

Table I presents a comprehensive comparison of existing studies in terms of their model architectures and their ability to capture four distinct characteristics of road networks. In the table, ‘G’ denotes that the method captures the traveling

semantics by constructing a global transition graph from trajectories rather than modeling individual trajectories, while ‘T’ indicates the use of techniques directly from *Toast* as a plugin module. Compared to other studies, *Toast* demonstrates the capability to simultaneously capture the network structure, traffic patterns and traveling semantics in a fine-grained manner with advanced pre-training strategies. Furthermore, *DyToast* addresses the limitations of all the previous methods by incorporating temporal dynamics for road segments and routes, thereby offering superior capabilities for time-sensitive applications in road networks.

III. PROBLEM FORMULATION AND OVERVIEW

In this section, we present required definitions in this paper, followed by articulating the problem statement. Next, we describe an overview of our proposed framework.

A. Problem Definitions

Definition 1. (Road Networks). Road networks are represented as a directed graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{C}_\mathcal{V})$. \mathcal{V} is a set of vertices, with each vertex v representing a road segment. \mathcal{E} is a set of edges, where each edge $e_{uv} \in \mathcal{E}$ represents a link connecting road segments u and v . $\mathcal{C}_\mathcal{V}$ is a set of features associated with road networks.

Definition 2. (Trajectory). A trajectory T is a sequence of sampled points $[p_i]_{i=1}^{|T|}$ from the underlying route of a moving object, and each point p_i corresponds to a coordinate of latitude and longitude.

Definition 3. (Route). A route $\mathbf{r} = [(r_i, t_i)]_{i=1}^n$ is time-ordered sequence consisting of n adjacent road segments within road networks G , where $r_i \in \mathcal{V}$ represents the i -th road segment in the route, and t_i represents the visit timestamp for r_i .

In our study, given road networks G , a trajectory T is first mapped to the road networks to get its underlying route \mathbf{r} by a map matching algorithm [39].

Problem Statement. Given road networks $G = (\mathcal{V}, \mathcal{E}, \mathcal{C}_\mathcal{V})$ and a trajectory dataset $\mathcal{D} = \{T^{(i)}\}_{i=1}^{|\mathcal{D}|}$, we aim to 1) learn vector representations $\{\mathbf{u}_v^t\}_{v \in \mathcal{V}}$ for road segments within the network, where t is the index of specific time frame (e.g., 8am-9am), and 2) derive the representation $\mathbf{u}_\mathbf{r}$ for any specified route \mathbf{r} on the road networks.

It is worth noting that our target is to learn generic representations for both road segments and trajectories rather than task-specific models. The obtained representations are versatile and can be easily applied to road various segment-based and trajectory-based downstream tasks.

B. Framework Overview

To obtain general-purpose representations for road networks, we propose *Toast* to tackle the two issues outlined in Section I (i.e., discrepancies and feature uniformity). Building upon this approach, we further introduce *DyToast*, which incorporates the modeling of temporal dynamics prevalent in

both road networks and trajectories. An overview of *DyToast* is presented in Fig. 2.

To tackle the first issue, we move beyond the conventional graph assumptions typically employed in existing studies [18], focusing on mitigating the discrepancies observed in road segments. To achieve this, we extend the skip-gram model [40], which is flexible in producing node representations based on a variety of structural assumptions for graphs, to capture traffic patterns (e.g., traffic volume). In addition to the original skip-gram objective of predicting the context neighbors of a target road segment, we introduce auxiliary tasks that predict traffic-related context features (e.g., road category) in a self-supervised manner. Such a multi-task learning paradigm allows the obtained representations to not only encode the graph structure but also differentiate among various traffic patterns that are indicated by these context features.

To tackle the second issue posed by the uniformity of features in various sub-regions, we learn from trajectories to extract traveling semantics on road networks. This includes identifying transition patterns and high-order dependencies between distant regions. To achieve this, we employ Transformer model [41] with two novel pre-training tasks for trajectory data tailored for road network contexts: *route recovery* and *trajectory discrimination*. Specifically, the *route recovery* task involves randomly masking a subsequence of road segments in a given route, and subsequently recovering the masked part based on the remaining segments of the route. The *trajectory discrimination* task aims to discriminate actual routes from the actual trajectories and those generated through random walks on road networks. These proposed techniques within *DyToast* enable encoding multi-faceted yet mutually enhanced characteristics of road networks into the representations. Moreover, our framework possesses the capability to produce representations for both individual road segments and trajectories.

Apart from the foundational capabilities conforming to the static characteristics of road networks, we have further augmented it by integrating temporal dynamics into its modules by using unified trigonometric functions. Specifically, we propose to construct a dynamic graph based on transition frequencies at each time frame, and then incorporate trigonometric-based, time-aware functions into skip-gram objective, thus enabling the capture of evolving patterns inherent in these graphs over time. Furthermore, apart from modeling only sequential information in trajectories, we also integrate the function into the self-attention mechanism in trajectory pre-training tasks, allowing the modeling of fine-grained temporal correlations. The modified architecture is adept at handling continuous timestamps with irregular intervals. These innovations can substantially improve the effectiveness of produced representations in time-dependent applications.

IV. METHODOLOGY

We elaborate our *DyToast* framework in this section. We start with preliminaries of the skip-gram model, and then discuss the extended skip-gram model augmented with an auxiliary traffic context prediction objective. Next, we describe the Transformer module and the two trajectory-enhanced pre-

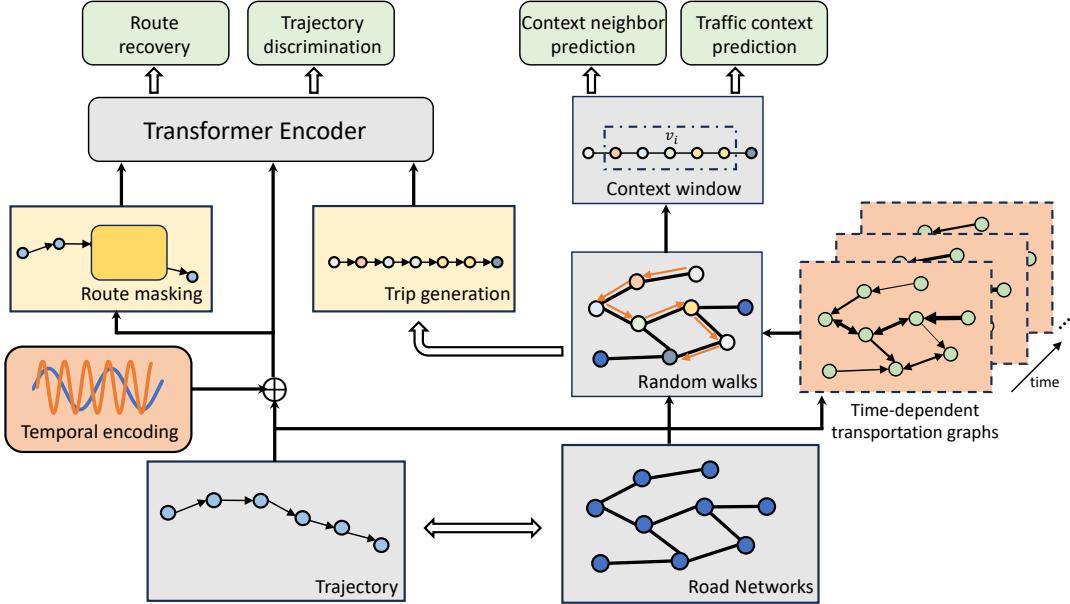


Fig. 2. The framework overview of our proposed DyToast. Components marked in red are designed to integrate temporal dynamics.

training tasks. Finally, we present the temporal encoding techniques integrated into these two modules.

A. Preliminaries: Skip-gram Model

The skip-gram model was originally introduced in word2vec [40] to learn embeddings for words. It has been widely adopted in graph representation learning methods later by viewing nodes in a graph as words in a document. This approach involves generating a set of random walks \mathcal{S} on a graph, with each random walk being treated as a sentence. The core objective of the model is to maximize the likelihood of observing the neighborhood nodes within a context window given a target node, which equals to minimizing the following loss function:

$$\begin{aligned} \mathcal{L}_{SG} &= - \sum_{v_i \in \mathcal{S}} \sum_{v_j \in \mathcal{N}(v_i)} \log p(v_j | v_i) \\ \log p(v_j | v_i) &= \log \frac{\exp(f(v_i)^\top h(v_j))}{\sum_{v'_j \in \mathcal{V}} \exp(f(v_i)^\top h(v'_j))} \end{aligned} \quad (1)$$

where $f, g : \mathbb{N} \rightarrow \mathbb{R}^d$ are the embedding functions for target nodes and context nodes respectively, $\mathcal{N}(v_i)$ is the context neighbors of node v_i , and \mathcal{S} is a random walk sequence from the set \mathcal{S} . For computational efficiency, we adopt negative sampling [40] to optimize the objective in practice, then the objective in Eq. 1 can be reformulated as follows:

$$\log(\sigma(f(v_i)^\top h(v_j))) + \sum_{\hat{v}_j \in \mathcal{V}} \log(\sigma(-f(v_i)^\top h(\hat{v}_j))) \quad (2)$$

where $\sigma()$ is the sigmoid activation function, and \mathcal{V} is the distribution of the vocabulary for negative sampling. By model training, the final node representations could capture various structural properties (e.g., homophily) via various random walk sampling strategies [13], [42].

B. Auxiliary Traffic Context Prediction Objective

Toast is designed to not only encode the structural assumptions of common graphs, but also to incorporate traffic patterns into representations. To achieve this, we propose to extend the skip-gram model by introducing auxiliary traffic context prediction tasks. For instance, road segments often have associated traffic context features, such as speed limits and road types, which we treat as auxiliary context information that indicates the traffic patterns of their respective road segments. Based on this, given a target road segment and its context neighbors, our key idea is to first determine the traffic context of the target node, and then to further predict the context neighbors. To perform traffic context prediction for a target road segment, we begin by applying binarization to the selected features that are indicative of traffic patterns. For example, if we select road type c_n from the set of traffic context features $\{c_n\}_{n=1}^N \in \mathcal{C}_V$, where c_n has $|c_n|$ possible categories, this feature is transformed into a $|c_n|$ -dimensional label vector where each dimension is 0 or 1, representing the presence of a specific category within the context of the target road segment. Formally, given a target road segment v_i and its corresponding N types of binarized traffic context features $\{c_n^i\}_{n=1}^N$, our goal is to minimize the binary cross entropy loss for any given context feature c_n :

$$\begin{aligned} \mathcal{L}_{c_n} &= - \sum_{v_i \in \mathcal{S}} \sum_{j=1}^{|c_n|} c_{nj}^i \log \sigma(f(v_i)^\top g(c_{nj})) + \\ &\quad (1 - c_{nj}^i) \cdot \log(1 - \sigma(f(v_i)^\top g(c_{nj}))) \end{aligned} \quad (3)$$

where c_{nj}^i is the j -th entry of the n -th binarized feature c_n for node v_i , $f(v_i)$ is the target embedding for node v_i , $g(c_{nj})$ is the feature embedding for c_{nj} that is shared across road segments, and $\sigma()$ denotes the sigmoid function.

Then, road segment representations are optimized to produce accurate predictions on both traffic context and context

neighbors. This strategy is more appropriate for road network settings than only predicting context neighbors. Moreover, the prediction tasks are structured hierarchically, such that the traffic context is utilized to enhance the prediction of context neighbors. In other words, when predicting the context neighbors, instead of only conditioning on the target road segment v_i as in Eq. 1, we refine this objective to incorporate traffic context as an additional conditioning factor:

$$\begin{aligned}\mathcal{L}_{SG'} &= - \sum_{v_i \in \mathbf{s}} \sum_{v_j \in \mathcal{N}(v_i)} \log p(v_j | v_i, \xi(v_i)) \\ &= - \sum_{v_i \in \mathbf{s}} \sum_{v_j \in \mathcal{N}(v_i)} \log \frac{\exp(\tilde{f}(v_i)^\top \tilde{h}(v_j))}{\sum_{v'_j \in \mathcal{V}} \exp(\tilde{f}(v_i)^\top \tilde{h}(v'_j))} \quad (4)\end{aligned}$$

where $\xi(v_i) \stackrel{\text{def.}}{=} [\sigma(f(u_i)^\top g(c_{nj})]_{j=1, n=1}^{|c_n|, N}$ is the n -th predicted traffic context of road segment v_i . $\tilde{f}(v_i)$ is the traffic-enhanced target embedding for v_i , namely, the concatenation of $f(u_i)$ and all the predicted traffic context $\xi(v_i)$, and $\tilde{h}(v_j)$ is the corresponding context embedding for node v_j . Similarly, we apply the negative sampling strategy as in Eq. 2 in practice.

The final objective function is a weighted sum of the modified skip-gram loss and the loss of all auxiliary traffic context prediction tasks. Formally, it is defined as

$$\mathcal{L} = \mathcal{L}_{SG'} + \sum_{n=1}^N \delta_n \mathcal{L}_{c_n} \quad (5)$$

where δ_n is the weight of the n -th auxiliary task. Compared to the original objective in Eq. 1, we incorporate a broader spectrum of semantic information, particularly traffic patterns, into the representations through our meticulously designed auxiliary tasks. Furthermore, the prediction of context neighbors is also refined with the inclusion of traffic context knowledge. As a result, this multi-task learning paradigm would produce more robust and effective road network representations.

C. Transformer and Pre-training Tasks

To tackle the feature uniformity issue suffered by existing methods, we employ a Transformer model with two novel pre-training tasks specifically designed to extract transition patterns and high-order dependencies on road networks. The effectiveness of Transformer pre-training in modeling text sequences has been extensively validated, particularly in learning semantically rich word representations for numerous downstream tasks [22], [43]. Given the sequential nature of trajectory data, we propose to leverage this model to learn representations for road networks. Now we proceed to describe the model details in a bottom-up manner.

1) Model Architecture: Input Embedding Layer. The road segment representations obtained from the first module serve as the input embeddings in Transformer. To preserve the order information in trajectories, learnable positional embeddings are integrated into the input representations as follows:

$$\mathbf{x}_i = \mathbf{u}_i + \mathbf{p}_i \quad (6)$$

where \mathbf{u}_i and \mathbf{p}_i are road segment representation and positional embedding for the i -th road segment, respectively.

Multi-head Self-attention. Self-attention mechanism allows the model to selectively focus on correlated parts of the input sequence. We follow the scaled inner-product form of self-attention, which can be described as mapping the representations of the input sequence to output representations [41]. Formally, it is defined as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} \right) \mathbf{V} \quad (7)$$

where \mathbf{Q} , \mathbf{K} and \mathbf{V} are the query, key, and value matrix respectively derived from a linear projection on the representations of the input trajectory, and d_k is the vector dimension, which is set to be the same for all the three matrices.

In our work, we adopt multi-head self-attention to model trajectory sequences. Specifically, the representations of input trajectory are projected into h sets of different queries, keys, and values to perform the self-attention mechanism, which has been shown to achieve better performance. Given the input representations $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{N \times d_{in}}$ with length N where \mathbf{x}_i is the representation of the i -th road segment in the trajectory after the embedding encoding layer, the output representations $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N] \in \mathbb{R}^{M \times d_{out}}$ are produced as follows:

$$\begin{aligned}\mathbf{Z} &= \text{MH-Attn}(\mathbf{X}) = [\text{head}_1, \dots, \text{head}_h] \cdot \mathbf{W}^O \\ \text{head}_i &= \text{Attention} \left(\mathbf{X}\mathbf{W}_i^Q, \mathbf{X}\mathbf{W}_i^K, \mathbf{X}\mathbf{W}_i^V \right) \quad (8)\end{aligned}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d_{in} \times d_{in}/h}$, $\mathbf{W}^O \in \mathbb{R}^{d_{in} \times d_{out}}$ are self-attention parameters.

Position-wise Feed-forward Network. After multi-head self-attention component, the output representations \mathbf{Z} are sent to a fully connected feed-forward network as follows:

$$\text{FFN}(\mathbf{Z}) = \Phi(\mathbf{Z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (9)$$

where $\Phi()$ is the ReLU activation function, \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{b}_1 and \mathbf{b}_2 are parameters of the feed-forward network.

Model Stacking. It is usually beneficial to learn more complex transition patterns in trajectory data by stacking multiple layers. In particular, each layer is composed of two sub-layers, namely multi-head self-attention and position-wise feed forward network, connected by residual connection and layer normalization as follows:

$$\begin{aligned}\mathbf{Z}' &= \text{LayerNorm}(\mathbf{X} + \text{MH-Attn}(\mathbf{X})) \\ \mathbf{Z}' &= \text{LayerNorm}(\mathbf{Z}' + \text{FFN}(\mathbf{Z}')) \quad (10)\end{aligned}$$

where LayerNorm denotes layer normalization and \mathbf{Z}' denotes the final output representations which are passed as the input to the subsequent layer of Transformer.

2) Model Pre-Training: Despite the capabilities of Transformer model, a critical concern is how to ensure that the derived representations adequately encode traveling semantics within road networks. The model's effectiveness largely depends on the design of loss functions that are adequately tailored to the specific domain (e.g., language [22], image [44], video [45]). To this end, it is important to devise appropriate pre-training tasks that demand the comprehension of traveling semantics on road networks.

Common pre-training tasks employed in Transformer models include masked language modeling [22], next token prediction [43], and other tasks such as next sentence prediction [22] and sentence order prediction [46]. While these tasks can generate effective text representations, they cannot achieve our target under road network settings. For example, consider the masked language modeling task, where each word in a sequence is randomly masked at a certain probability (e.g., 15%), and the model is then tasked with predicting these masked words. However, this task is less effective when applied to trajectories within road network contexts. This is because two consecutive road segments in a trajectory must be connected in road networks. When a road segment is masked within such a sequence, it can often be trivially inferred from the knowledge of the graph structure and its given adjacent road segments, as i represents the only segment that makes the sequence a valid route. Since the graph structure is well captured by the skip-gram objective, this task may not contribute additional valuable information for the representations. Moreover, in the next token prediction task, the focus is primarily on forward prediction within a sequence, which does not contribute to a comprehensive understanding of the entire trajectory. In addition, sentence-level pre-training tasks do not naturally align with the goals of road network representation learning.

To this end, we propose two novel pre-training tasks for trajectory data within road networks: *route recovery* and *trajectory discrimination*. These tasks are tailored to effectively encode the traveling semantics into representations.

Route Recovery. Different from the masked language model task where every independent word is randomly masked, we mask a continuous sequence of road segments within a trajectory to make it into a partially observed route. In particular, given a route, we randomly mask 40% of the consecutive road segments in the sequence. This task prevents the trivial recovery of the masked road segments based solely on the awareness of the graph structure. Instead, it requires the representations to capture more complex transition patterns and accurately identify the most likely options for the masked segments. The model is trained by the cross entropy loss between masked road segments and the predicted ones.

Trajectory Discrimination. This task is designed to enhance the model's ability to distinguish real trips from generated ones. Real trips are sampled from our trajectory databases, while fake trips are generated through random walks on road networks. We train the model to minimize the prediction error for these two types of trips. The purpose of this task is two-fold. First, it provides an alternative way for the model to capture transition patterns. After training, the model is able to identify fake trips by recognizing sub-sequences that do not follow the normal transition patterns. Second, this task offers a holistic perspective on traveling semantics across road networks. Trajectories naturally span various regions of the network, and by accurately identifying actual trips, especially those that occur frequently between distant regions, the model can effectively capture high-order dependencies and correlations between distant road segments.

D. Encoding Temporal Dynamics

1) Dynamic Extension for Traffic-enhanced Skip-gram:

Time-dependent Transportation Graph Construction. To fuse the enhanced skip-gram model with dynamicity, we propose to construct time-dependent transportation graphs. These graphs are designed to characterize not only the static structural information of road networks, as captured in the previous module, but also the dynamic transition information. Specifically, time is split into discrete time frames, each representing a specific period (e.g., 8am-9am). For each time frame t , we augment the road network graph G to form $G^{t_i} = (\mathcal{V}, \mathcal{E}_t, \mathcal{C}_V)$, which reflects the holistic transportation condition at each time frame t . Here, \mathcal{V} and \mathcal{C}_V remain as the road segments and their corresponding features, respectively, while each edge $e_{ij}^t = (v_i, v_j, w_{ij}^t) \in \mathcal{E}_t$ denotes an adjacent link between road segments i and j with an associated weight w_{ij}^t , defined as:

$$w_{ij}^t = \gamma \times e_{ij} + \text{count}^t(v_i \rightarrow v_j) \quad (11)$$

where $e_{ij} \in \{0, 1\}$ indicates the presence of a structural edge on road networks, $\text{count}(v_i \rightarrow v_j)$ denotes the frequencies of transitions between road segment i and j within time frame t , and γ is the hyperparameter to balance the influence of these two terms. By constructing a collection of transportation graphs, we gain a holistic view of vehicle movements for each time frame, thus supplementing the static traffic context, typically indicated by features such as speed limits and road types, with dynamic traffic patterns.

Trigonometric Parameterization. Equipped with the time-dependent transportation graphs, we perform the strategy as in Eq. 2 while integrating a novel temporal encoding technique, which modifies the target embedding function f to additionally condition on the time variable t :

$$\log(\sigma(f(v_i, t)^\top h(v_j))) + \sum_{\hat{v}_j \in \hat{\mathcal{V}}} \log(\sigma(-f(v_i, t)^\top h(\hat{v}_j))) \quad (12)$$

Here, the proximity degree calculated by $f(v_i, t)^\top h(v_j)$ is essential to capture the evolving patterns inherent in the transportation graphs over time. To achieve this, we employ sinusoidal function: $\psi(t) : \mathbb{R} \rightarrow \mathbb{R}^d$ to model road segment representations specific to time frame t , defined as:

$$\psi(t) = [\cos(\mathbf{w}_t \odot t) || \sin(\mathbf{w}_t \odot t)] \in \mathbb{R}^d, \quad (13)$$

where $\mathbf{w}_t \in \mathbb{R}^{d/2}$ is a learnable parameter to control the frequencies, \odot indicates the broadcast multiplication between vector and scalar, and $||$ denotes vector concatenation. Then the proximity degree is expressed as:

$$\begin{aligned} f(v_i, t)^\top h(v_j) &= \mathbf{u}_i^{t\top} \mathbf{v}_j = (\mathbf{u}_i + \psi(t))^\top \mathbf{v}_j \\ &= \mathbf{u}_i^\top \mathbf{v}_j + \sum_{k=1}^{d/2} v_{j,k} \cos(w_{t,k} t) + v_{j,k+\frac{d}{2}} \sin(w_{t,k} t), \end{aligned} \quad (14)$$

This formulation belongs to the trigonometric polynomial $a_0 + \sum_{k=1}^K a_k \cos(kx) + b_k \sin(kx)$ where $a_0, \dots, a_K,$

$b_1, \dots, b_K \in \mathbb{R}$. Such a function, with suitably selected coefficients, can approximate any periodic continuous functions defined over an arbitrarily closed interval [47], [48]. This capacity aligns well with the typical characteristics of road networks, which often exhibit fluctuating and periodic patterns across days that can be learned by the proposed technique. Consequently, this approach significantly enhances the effectiveness of the representations in capturing temporal aspects.

2) *Temporal Feature Integration in Transformer*: The original Transformer applied in our previous study is limited to only encoding sequence ordering information through pre-defined or learned positional embeddings [41], [49]. To further effectively model temporal correlations encoded in road segments within trajectories, an intuition approach is to employ discrete embeddings to represent specific time intervals, with each embedding associated with a particular time bin (e.g., 30 seconds). However, this approach has significant drawbacks. It becomes challenging to select appropriate intervals to obtain discrete embeddings, particularly when dealing with irregular time intervals between consecutive road segments. In addition, the method also fails to model the fine-grained correlations for two different intervals that fall in the same time bin.

In light of this, we adopt the sinusoidal function as in Eq. 13 to model continuous visit timestamp t_i at road segment r_i within Transformer. Specifically, the self-attention mechanism computes dot-product between two encoded temporal features $\psi(t_i)$ and $\psi(t_j)$ as:

$$\psi(t_i) \cdot \psi(t_j) = \mathcal{K}(t_i, t_j) = \mathbf{1}^\top \cos(\mathbf{w}_t \odot (t_i - t_j)) \quad (15)$$

This means that such a temporal encoding function can be viewed as a translation-invariant kernel (i.e., $\mathcal{K}(x, y) = \mathcal{K}(x + c, y + c)$), which offers several advantages. First, the translation invariant property allows the model to focus on the interval gaps between timestamps, which are more informative for denoting travel time on road segments than the absolute values of timestamps. Second, it enables direct modeling of correlations on continuous timestamps without manually selecting the intervals for discrete embeddings and thus reducing the information loss. Then the input representations into the Transformer are modified as follows:

$$\hat{\mathbf{x}}_i = \mathbf{x}_i + \psi(t_i) + \mathbf{e}_t \quad (16)$$

where \mathbf{x}_i is the representation for road segment i within a trajectory as in Eq. 6, and \mathbf{e}_t is the time frame representation which encodes coarse-grained temporal information corresponding to the start time of the trajectory. To facilitate the modeling of temporal correlations, we initialize the parameter from a normal distribution $\mathbf{w}_t \sim \mathcal{N}(0, \sigma^{-2})$. By doing this, Eq. 15 approximates the Gaussian kernel (i.e., $\psi(x) \cdot \psi(y) \approx \exp(-\|x - y\|^2/\sigma^2)$) over its temporal differences [50]. This introduces a useful inductive bias of L_2 distances as the starting point in the model, enhancing its ability to capture temporal dynamics within trajectories.

3) *Computational Complexity*: The computational cost of DyToast can be decomposed into two parts: the traffic context-aware skip-gram module and the Transformer encoder module.

During the entire training phase, the complexity of the traffic context aware skip-gram module is $\mathcal{O}(\mathcal{N} \cdot |\mathcal{S}| \cdot |\mathcal{V}| \cdot (k + |c_n| + 1))$, where \mathcal{N} is the size of context neighbors, $|\mathcal{S}|$ is the length of random walks, $|\mathcal{V}|$ is the number of road segments, k is the number of negative samples, and $|c_n|$ is the number of auxiliary task labels. For the Transformer encoder, the complexity is the same with the Transformer architecture with $\mathcal{O}(|\mathcal{D}||\mathcal{S}|^2d + |\mathcal{D}||\mathcal{S}|d^2)$, where $|\mathcal{D}|$ denotes the number of trajectories used for training, and d is the dimension of the Transformer hidden states. In the inference phase, our goal is to derive representations for road segments and routes. Dynamic road segment representations can be obtained via look-up operations for road segments and specific time frames, followed by the summation of these vectors, with a complexity of $\mathcal{O}(d)$. To derive a route representation, the representations for each road segment are organized into a sequence and processed by the Transformer encoder, resulting in a final route representation with a complexity of $\mathcal{O}(|\mathcal{S}|^2d + |\mathcal{S}|d^2)$.

4) *Remarks*: The adoption of the sinusoidal function in our framework offers a unified solution to significantly enhance the temporal dynamics encoding within the proposed two modules. On the one hand, it complements the skip-gram objective by capturing fluctuating and periodic patterns inherent in time-dependent transportation graphs. This capability contributes to a holistic understanding of the entire road network, thereby providing a macroscopic perspective of temporal evolution. On the other hand, this function serves as an effective way of encoding visit timestamps within trajectories. Its seamless integration into the Transformer's self-attention mechanism enables the model to perform continuous and translation-invariant modeling of fine-grained temporal correlations. In this way, it facilitates the capability of understanding microscopic higher-order dependencies for road segments from each individual trajectory.

As a result, the representations produced by DyToast – both in terms of road segment representations and trajectory representations derived from Transformer outputs – are enriched with multi-faceted characteristics enhanced by the inclusion of temporal dynamics. The model's ability to capture dynamic traffic patterns and traveling semantics ensures that these representations are highly effective for time-sensitive downstream applications.

V. EXPERIMENTS

In this section, we compare our proposed framework against other methods applied in road network representation learning. We perform extensive experiments on two real-world datasets, and across three time-sensitive tasks to test the effectiveness of the learned representations for both road segment-based and trajectory-based applications.

A. Datasets

We utilize two datasets comprising road networks and trajectory data from two cities, *Chengdu* and *Xi'an*. The road networks are obtained from OpenStreetMap [51], while the trajectory data was obtained from the ride-hailing company DiDi, spanning the month of November 2016. To verify the

effectiveness of leveraging trajectory data in road network representation learning, we filter out the road segments not covered by trajectory data. We further apply map matching algorithm [39] to convert the GPS records from trajectories into sequences of road segments. The statistics of the datasets are presented in Table II.

TABLE II
STATISTICS OF THE DATASETS

Dataset	#Road Segments	#Edges	#Trajectories
Chengdu	6,125	15,933	5,266,120
Xi'an	5,146	12,804	2,533,359

B. Compared Methods

To evaluate the performance of our proposed Toast and DyToast, we conduct extensive comparisons with 9 baseline methods, described as follow.

Conventional graph learning methods:

- **node2vec** [13]: It employs biased random walks on road networks to explore neighborhood of each road segment, capturing both local and high-order structural knowledge within road networks.
- **GCN** [52]: It is the implementation of graph convolutional network for road networks. The model is trained to reconstruct the original road network structure.
- **GAT** [53]: It is the implementation of graph attention network that applies attention mechanism in the aggregation operation. The model is trained to reconstruct the original road network structure.

Standard road network representation learning methods:

- **SRN2Vec** [18]: It adopts multi-task learning to make road segments with similar properties, such as spatial closeness and road shapes, close in the representation space.
- **HRHR** [20]: It employs a hierarchical GNN to model different semantic levels in road networks. It utilizes two reconstruction tasks to learn the inter-relationships between the three layers of the road network hierarchy.
- **RFN** [21]: It utilizes a multi-view GNN to learn representations from both node-relational and edge-relational perspectives of road network graphs.
- **SARN** [24]: It adapts graph contrastive learning techniques to road networks by integrating spatial proximity and distance-based negative sampling in its data augmentation stage.

Trajectory-enriched representation learning methods:

- **JCLRNT** [25]: It aims to derive both road segment and trajectory representations through contrastive learning, including the contrast between road-road, road-trajectory and trajectory-trajectory interactions.
- **TrajRNE** [27]: It leverages trajectory data to construct adjacency matrix for GNN aggregation, and incorporates road context information using techniques similar to SRN2Vec.

C. Experimental Settings

1) **Downstream Tasks:** Given the capability of DyToast to produce both road segment and trajectory representations, we adopt the experimental setups and tasks commonly used in prior studies [17], [20], [25], [27]. These studies utilize learned representations as inputs for training simple models across various downstream tasks, following a typical paradigm for leveraging general-purpose entity representations in the geospatial domain (e.g., POIs, road networks, and regions). This approach effectively evaluates the transferability of representations to diverse downstream tasks [54]. Specifically, we select three tasks from previous studies for evaluation: road traffic inference, travel time estimation and destination prediction [17], [20], [25], [27]. These tasks are evaluated by utilizing static representations produced by models, which typically ignore the temporal effects. As these tasks are inherently time-sensitive, they provide a meaningful benchmark for evaluating the model's ability to capture temporal dynamics, aligning closely with the scope of this paper.

Road speed inference. Our target is to infer dynamic traffic speeds on all road segments in scenarios where only partial traffic speed observations are available. Specifically, at each time frame, and aim to infer the average traffic speed for road segments that have missing values, using the model trained on traffic speeds from other road segments. For evaluation, we split a day into one-hour time frames, and extract the speed information at each time frame using the aggregated records across the days from the dataset to avoid data sparsity. Then we randomly mask out 20% of the traffic speed data at each time frame. We apply 5-fold cross validation to evaluate the performance of all the compared methods. In these methods, the learned road segment representations are utilized as input features into a two-layer fully-connected neural network, which functions as a regressor for this task.

Travel time estimation. Our target is to estimate the travel time of trajectories that start at varying time frames. Specifically, for methods that do not inherently generate trajectory representations, we produce the trajectory representations by employing a two-layer Transformer to process the representations of road segments as inputs. In contrast, for trajectory-enriched methods that are equipped to produce trajectory representations, we exclude timestamps in the inputs to avoid data leakage. Subsequently, the derived or directly produced trajectory representations are fed into a linear layer to get the prediction of travel time for all the methods. We define each time frame as one hour, and use 80% trajectory data for pre-training tasks when applicable, while the remaining 20% is further partitioned into 4:1 ratio for the task-specific training and evaluation stages.

Destination prediction. Our target is to predict the destination road segment of trajectories that start at varying time frames. Specifically, we utilize the initial 50% of road segments as partial trajectories to produce their corresponding representations. Then these representations are fed into a linear layer to classify the destination road segment. The strategies for deriving trajectory representations, as well as the data

TABLE III
PERFORMANCE OF THE COMPARED METHODS ON THE CHENGDU DATASET.

Task	Road Speed Inference (m/s)		Travel Time Estimation (s)		Destination Prediction (%)	
	Metric	MAE	RMSE	MAE	RMSE	Acc@5
node2vec	9.69 ± 0.13	12.92 ± 0.11	98.77 ± 1.87	142.08 ± 2.02	50.37 ± 0.21	59.50 ± 0.18
GCN	9.15 ± 0.20	12.27 ± 0.18	97.73 ± 2.36	141.22 ± 2.75	49.51 ± 0.32	58.53 ± 0.28
GAT	9.33 ± 0.23	12.56 ± 0.26	96.78 ± 2.17	140.33 ± 2.89	49.12 ± 0.22	57.90 ± 0.19
SRN2Vec	8.64 ± 0.16	11.71 ± 0.19	96.44 ± 1.72	139.38 ± 1.67	55.38 ± 0.24	64.42 ± 0.23
HNRN	8.70 ± 0.13	11.82 ± 0.12	93.72 ± 1.48	136.99 ± 1.63	54.60 ± 0.45	63.69 ± 0.43
RFN	8.99 ± 0.35	12.25 ± 0.41	93.27 ± 3.18	136.29 ± 3.02	47.44 ± 0.72	56.09 ± 0.99
SARN	8.96 ± 0.18	12.20 ± 0.18	94.80 ± 1.32	137.28 ± 1.44	52.57 ± 0.20	61.79 ± 0.22
TrajRNE	9.03 ± 0.20	12.27 ± 0.24	93.51 ± 1.53	136.76 ± 1.60	56.24 ± 0.35	65.44 ± 0.48
JCLRNT	7.89 ± 0.22	10.98 ± 0.27	88.18 ± 1.79	129.02 ± 1.93	60.52 ± 0.33	69.77 ± 0.36
Toast	8.74 ± 0.15	11.89 ± 0.16	82.86 ± 1.46	119.49 ± 1.37	58.79 ± 0.25	67.01 ± 0.26
DyToast	7.01 ± 0.17	10.14 ± 0.21	76.19 ± 1.60	111.83 ± 1.84	63.28 ± 0.25	72.39 ± 0.25

TABLE IV
PERFORMANCE OF THE COMPARED METHODS ON THE XI'AN DATASET.

Task	Road Speed Inference (m/s)		Travel Time Estimation (s)		Destination Prediction (%)	
	Metric	MAE	RMSE	MAE	RMSE	Acc@5
node2vec	8.42 ± 0.11	10.85 ± 0.13	123.59 ± 1.73	190.92 ± 2.01	47.67 ± 0.19	57.11 ± 0.18
GCN	7.85 ± 0.17	10.20 ± 0.17	120.69 ± 1.96	187.06 ± 1.87	43.73 ± 0.28	52.94 ± 0.27
GAT	8.04 ± 0.20	10.48 ± 0.22	118.49 ± 2.38	186.03 ± 2.60	44.76 ± 0.13	54.48 ± 0.14
SRN2Vec	7.49 ± 0.09	9.88 ± 0.11	115.63 ± 1.52	182.06 ± 1.53	52.21 ± 0.19	61.38 ± 0.17
HNRN	7.75 ± 0.14	10.17 ± 0.18	114.49 ± 1.44	180.80 ± 1.71	51.85 ± 0.39	60.93 ± 0.43
RFN	7.93 ± 0.26	10.41 ± 0.28	116.51 ± 2.87	182.27 ± 2.66	46.57 ± 0.76	56.02 ± 0.89
SARN	7.67 ± 0.16	10.05 ± 0.15	112.49 ± 1.30	178.36 ± 1.29	48.82 ± 0.17	58.06 ± 0.20
TrajRNE	7.75 ± 0.17	10.10 ± 0.19	115.33 ± 1.48	181.70 ± 1.53	52.69 ± 0.38	62.57 ± 0.40
JCLRNT	6.97 ± 0.20	9.32 ± 0.25	108.92 ± 1.83	172.44 ± 1.98	55.88 ± 0.27	66.35 ± 0.25
Toast	7.60 ± 0.12	9.99 ± 0.13	108.05 ± 1.57	171.12 ± 1.62	55.22 ± 0.26	65.99 ± 0.28
DyToast	6.53 ± 0.15	8.99 ± 0.17	92.84 ± 1.56	146.04 ± 1.72	58.95 ± 0.25	68.77 ± 0.26

partitioning settings, are consistent with those outlined in travel time estimation task.

For the tasks of road speed inference and travel time estimation, we use mean absolute error (MAE) and root mean square error (RMSE) as evaluation metrics. For the task of destination prediction, we use Top- N accuracy (Acc@ N) as metrics to evaluate the proportion of instances where the actual destination road segment appears in top- N predictions ranked by highest probabilities. We run each method 10 times and report the results along with their standard deviations.

2) *Parameter Settings*: To select the model parameters, we began with a grid search for a subset of parameters considering the huge search space. Specifically, we explored the embedding size from the candidate set {32, 64, 96, **128**, 256} and the number of layers from {1, 2, 3, 4}, while keeping the mask ratio fixed at 0.25 and the auxiliary loss weight fixed at 2. After that, we refined other parameters around the optimal configurations identified in the grid search (embedding size 128 and 2 layers) by testing additional combinations of mask ratio for Transformer pre-training and auxiliary loss weight for traffic context prediction task from {0.1, 0.25, **0.4**, 0.55, 0.7} and {0.1, 0.5, **1**, 2, 5} respectively. The final parameters selected for the model are highlighted in bold. To ensure fair comparison, we set the dimension of the representations for both road segments and trajectories to be 128 for all the

compared methods, and select road type as the prediction objective for auxiliary traffic context prediction task. We apply 30 training epochs for both modules iteratively in our experiments. For the baseline methods, we adhere to the default configurations for their model architectures as described in their respective papers.

D. Performance Comparison

The results of all the methods across the three tasks on the Chengdu and Xi'an dataset are presented in Table III and Table IV, respectively. Then we have several observations.

First, methods such as node2vec, GCN and GAT, which are not specifically designed for road networks, yield the worst results among the baselines. This highlights the importance of developing approaches to tackle the distinctive characteristics of road networks. Second, methods like SRN2Vec, HNRN, RFN and SARN, which focus on capturing road-specific features and spatial information, demonstrates improved performance in the road speed inference task compared to those generic graph representation learning methods. However, they are less effective in trajectory-based tasks due to a lack of capability in modeling high-order dependencies among road segments, which are usually reflected in trajectory data. Furthermore, methods that leverage trajectory data for extracting high-order dependencies, including TrajRNE, JCLRNT,

and Toast, although not showing further improvement in road speed inference task, exhibit enhanced performance in trajectory-based tasks. Fourth, among all the baselines, Toast achieves the best performance in travel time estimation, due to its tailored sequence modeling for trajectory pre-training. On the other hand, JCLRNT, building upon Toast by integrating contrastive learning objectives, facilitates enriched interactions between road segments and trajectories, thus enhancing the effectiveness for these two data modalities. Therefore, it achieves the superior performance in both road speed inference and destination prediction. Lastly, the baseline methods generally fall short in capturing the dynamic aspects of road network representation learning. In contrast, DyToast introduces a unified temporal encoding strategy, adeptly adapting to the temporal dynamics inherent in road networks. As a result, our proposed DyToast outperforms all the compared methods with substantial margin across these time-sensitive tasks.

E. Model Analysis

1) *Ablation Study*: We conduct an ablation study by removing different components to investigate their contributions to the performance. Specifically, we test the effectiveness of components that enrich temporal dynamics, given that other components have been previously evaluated in the Toast study [23]. For this purpose, we compare DyToast with the following variants:

- **DyToast-G**: it removes the construction of time-dependent transportation graphs, relying instead on the static road network structure while still applying the proposed techniques for encoding temporal dynamics.
- **DyToast-S**: it removes the temporal encoding mechanism within the traffic-enhanced skip-gram module.
- **DyToast-T**: it removes the temporal feature integration module within Transformer, omitting the capture of fine-grained temporal correlations in trajectory data.
- **DyToast-ST**: it is a combination of the variants of DyToast-S and DyToast-T by removing both components in skip-gram and Transformer modules.

The results for these model variants on the tasks of traffic speed inference and travel time estimation are shown in Fig. 3. Based on the results, we can observe that excluding different components from the framework leads to a decrease in performance across both road segment-based and trajectory-based tasks. This highlights the importance of integrating temporal dynamics on road networks from various perspectives to enhance the model performance. Notably, the removal of temporal encoding within the pre-trained Transformer module (-T) indicates a more pronounced impact on model performance than modifications to the traffic-enhanced skip-gram module (-G and -S), demonstrating the benefits of capturing fine-grained temporal correlations in trajectory data. Furthermore, the contributions of these components are complementary, as the removal of multiple modules (i.e. -ST) result in the most significant performance degradation. Overall, the ablation study validates the effectiveness of our proposed techniques to enrich the knowledge on temporal dimension into road network representation learning.

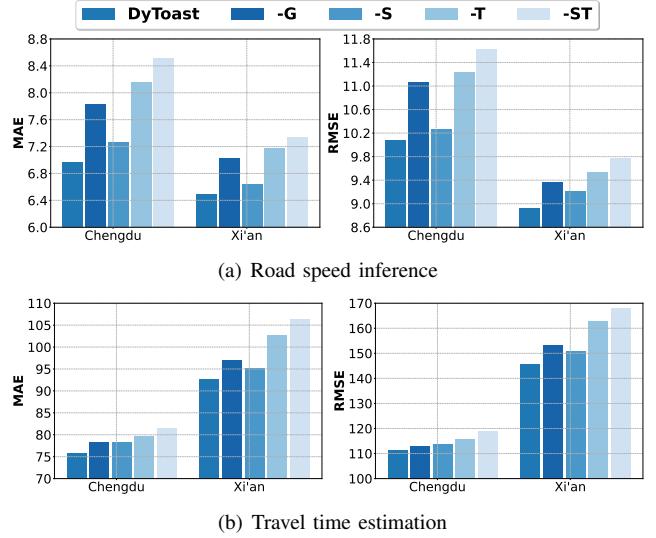


Fig. 3. Ablation study of four model variants without modules for encoding temporal dynamics.

TABLE V
COMPARISON FOR TEMPORAL ENCODING TECHNIQUES ON TRAFFIC SPEED INFERENCE TASK.

Dataset	Chengdu		Xi'an	
	Metric	MAE	RMSE	MAE
T-Emb	7.66 ± 0.23	10.74 ± 0.28	7.03 ± 0.20	9.41 ± 0.27
Road-Emb	7.48 ± 0.18	10.38 ± 0.20	6.70 ± 0.17	9.13 ± 0.19
T-Attention	7.56 ± 0.34	10.50 ± 0.39	6.73 ± 0.35	9.15 ± 0.41
CPE	8.02 ± 0.37	11.20 ± 0.42	7.23 ± 0.41	9.58 ± 0.48
DyToast	7.01 ± 0.17	10.14 ± 0.21	6.53 ± 0.15	8.99 ± 0.17

2) *Temporal Encoding Techniques*: We further examine the effectiveness of our temporal encoding techniques in DyToast compared to other temporal encoding methods utilized for trajectory data. Specifically, we replace the techniques as described in Section IV-D with four alternative techniques for incorporating time information, while keeping all other components consistent. The details of these techniques are listed as follows:

- **T-Emb**: it partitions the time into discrete 1-hour intervals and represents each interval with a distinct embedding. These embeddings are applied to all road segments, and subsequently utilized as inputs to the traffic-enhanced skip-gram module.
- **Road-Emb**: unlike T-Emb, it assigns discrete embeddings based on 1-hour time intervals to each road segment independently. In other words, every road segment possesses its own set of time embeddings.
- **T-Attention** [55]: it employs a time interval-aware self-attention mechanism by transforming these intervals into bias terms in attention score calculation in Transformer.
- **CPE** [56]: it converts time intervals into kernels, which are applied within convolution operations to incorporate fine-grained temporal information. The results after convolution are utilized as inputs in Transformer.

The results of the traffic speed inference task against these compared methods are shown in Table V, and similar results

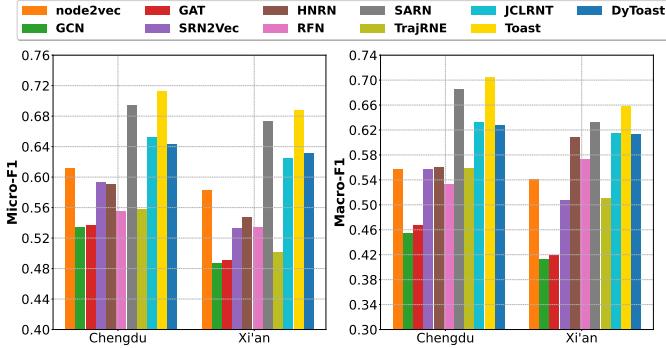


Fig. 4. Results of compared methods on road label classification task.

can be found on other tasks. From the results, we can make the following observations. First, CPE shows the worst performance, attributed to the misalignment between its original application context in GPS trajectories and the map-matched trajectories in our scenario. Second, despite the simplicity of discrete time embeddings (T-Emb and Road-Emb), they achieve performance improvements for the time-sensitive task. This indicates the benefits of integrating temporal information. While Road-Emb outperforms T-Emb, it also significantly increases the parameter numbers compared to all other methods. Third, T-Attention generally produces the best results among baselines, validating the effectiveness of enhancing the self-attention mechanism with additional temporal knowledge in trajectory data. Last, DyToast further advances the self-attention mechanism by incorporating sinusoidal function, exhibiting superior properties such as translation invariance, strong inductive bias and the capacity for continuous temporal modeling. Moreover, the learned parameters do not increase with the number of frames increases. These features facilitate a better capture of temporal dynamics, thereby surpassing these baseline methods in performance.

3) *Performance on Static Task:* While temporal dynamics are effectively captured in DyToast, it is worth investigating how the associated module impacts the performance on static tasks. To this end, we follow the experimental settings in [23] and utilize the learned road segment representations as inputs for the road type classification task. Specially, we employ a logistic regression classifier and five-fold cross validation to evaluate the performance of all the compared methods. For evaluation metrics, we use Micro-F1 and Macro-F1 scores, as adopted in prior studies [23], [26], [27]. For DyToast, we calculate the average road segment representations across time frames to serve as input for the classification task.

The results of the road type classification task are presented in Figure 4, and several observations can be drawn. First, methods that incorporate trajectory data with sequential models, namely JCLRNT and our proposed two methods, generally achieve effective performance. This is because traveling semantics in trajectory sequence inherently encode correlations in representations for road segments with similar static features. For example, trajectories tend to traverse coherent roads (e.g., same road type) rather than switching frequently unless necessary. Second, SARN achieves the best result among GNN-based methods by a large margin. This can be attributed

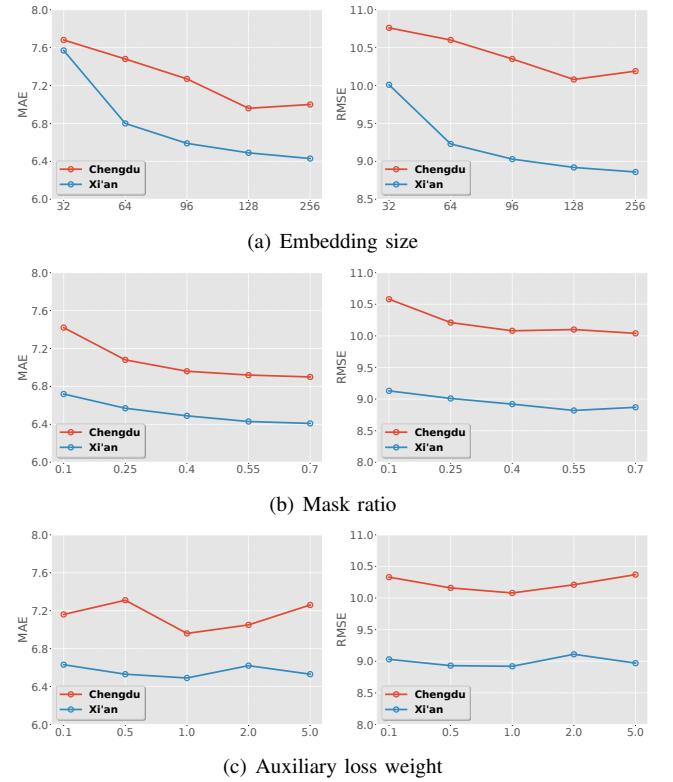


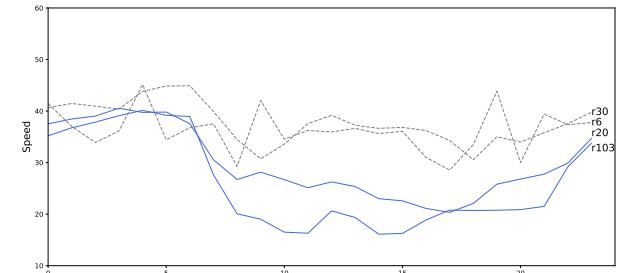
Fig. 5. Impact of hyperparameters on model performance.

to its contrastive learning paradigm, which explicitly generates similar representations for road segments that share similar spatial characteristics (e.g., shape), rather than relying solely connectivity as in other methods, such as HNRR and GAT. Third, while **Toast** outperforms other compared methods, the inclusion of temporal dynamics does not necessarily enhance the performance of the static task, where temporal information is not critical. Instead, encoding temporal dynamics may introduce additional noise, thereby negatively affecting the model's effectiveness on the static task. As a result, this represents a key limitation of our proposed method, which is discussed further at the end of this section.

4) *Impact of Hyperparameters:* We study the impacts of various hyper-parameters on the model performance, including embedding size, mask ratio in Transformer pre-training, and the weight of auxiliary traffic context prediction objective. The results for the traffic speed inference task are presented in Fig. 5. We omit the results for other tasks since the patterns are found to be similar.

Impact of embedding size. As illustrated in Fig. 5(a), an increase in embedding size generally leads to improved performance. However, when the embedding size exceeds 128, the improvement becomes negligible, or even degrade on the Chengdu dataset probably due to over-fitting issues. Thus, an embedding size of 128 is set as the default value.

Impact of mask ratio. As illustrated in Fig. 5(b), increasing the mask ratio during the Transformer pre-training with trajectory data improves the performance. However, the benefits of increasing the mask ratio becomes saturated after the mask



(a) Dynamic speeds for four sampled road segments

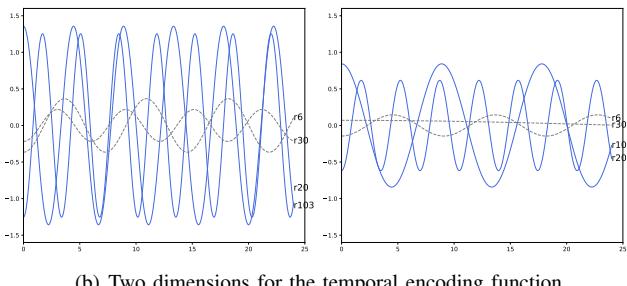


Fig. 6. Illustration of two dimensions for the learned temporal encoding functions.

ratio of 0.4, while accompanied by more computational cost. Therefore, a mask ratio of 0.4 is chosen as the default setting.

Impact of auxiliary loss weight. As illustrated in Fig. 5(c), selecting either excessively low or high weights for the traffic context prediction objective exhibits negative effect on performance. An optimal value is achieved at a weight of 1.0, which is adopted in our experiments.

5) *Analysis on Temporal Encoding Function:* In addition to the quantitative results presented in Section V-E2, we further analyze the effects of the learned trigonometric function used to encode the temporal dynamics in the proposed method. For this analysis, we select four road segments with two distinct dynamic speed characteristics. As shown in Fig. 6(a), r_{30} and r_6 represent road segments which exhibit relatively stable speeds throughout the day, and r_{20} and r_{103} represent road segments which exhibit higher fluctuations, characterized by lower speeds during the daytime and higher speeds at night.

The learned trigonometric functions for two dimensions are depicted in Fig. 6(b). For r_6 and r_{30} , the functions demonstrate minimal fluctuations or remain nearly constant, indicating that these road segments experience limited variation over time for the given dimensions. In contrast, r_{20} and r_{103} display more pronounced amplitude variations, reflecting the greater temporal dynamics associated with these road segments. Moreover, the frequencies of the learned functions are lower for r_6 and r_{30} compared to r_{20} and r_{103} , aligning with the intuition that functions with longer periodicity correspond to more moderate temporal dynamics. This analysis demonstrates the ability of the proposed method to effectively capture and differentiate varying temporal patterns in road segments.

6) *Case Study:* We study how the evolution of road segment representations over a day by visualizing r_6 and r_{103} , which exhibit relatively stable speeds and higher fluctuations, respectively (Fig. 6(a)) using t-SNE [57]. The results are presented

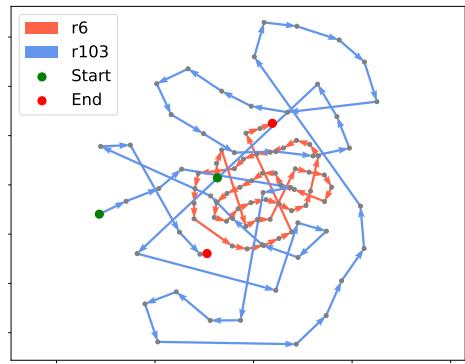


Fig. 7. Visualization of two sampled dynamic road segment representations.

in Fig. 7.

We can observe that the representations of r_6 form a compact and localized cluster, suggesting relatively stable and consistent dynamics that align with its speed characteristics with minimal speed fluctuations. In contrast, the representations of r_{103} span a broader area with a more distributed pattern, reflecting the higher variability and dynamic changes inherent to this segment. Moreover, despite the observed fluctuations of the road segment representations, the end representations for both road segments are positioned close to their start representations. This demonstrates the clear periodicity in traffic speed patterns (i.e., daily traffic patterns tend to repeat). This visualization exhibits strong interpretability of our method, which effectively captures and distinguishes the temporal dynamics of road segments.

Discussions. Through extensive experiments, we can identify several insights into the strengths and limitations of our proposed methods. **Toast** and **DyToast** introduce innovative techniques to jointly capture traffic patterns and traveling semantics on road networks, addressing the gaps in existing studies. By combining context-aware skip-gram models and Transformer-based trajectory modeling, the methods effectively capture the unique characteristics of road networks. Moreover, we propose to further integrate temporal dynamics in **DyToast** through unified trigonometric functions, achieving superior performance across time-sensitive tasks such as road speed inference, travel time estimation, and destination prediction (Section V-D). Within the model, different components demonstrate their meaningful contributions to the overall performance. This reflects the well-considered design of each module for encoding temporal dynamics (Section V-E1). Furthermore, the proposed temporal encoding technique based on unified trigonometric functions not only accommodates varying time frame lengths without increasing space complexity, but also outperforms alternative temporal encoding approaches found in related works (Section V-E2). Apart from these strengths, **DyToast** shows relative insensitivity to hyperparameter settings, facilitating robust performance without extensive tuning (Section V-E4). In addition, it demonstrates good properties in modeling temporal dynamics for evolving traffic patterns with interpretability, as illustrated in the analysis of temporal encoding function and case study (Section V-E5 & Section V-E6).

Despite these significant strengths, the proposed methods still possess limitations, which present opportunities for further improvement. First, the reliance on extensive trajectory data for modeling trajectory sequences and constructing transition graphs poses challenges for data collection, particularly in underdeveloped cities with limited urban sensing infrastructure. Enhancing the methods to perform effectively in data-scarcity settings is an important direction for future research. Second, while encoding temporal dynamics enhances performance on time-sensitive tasks, it reduces effectiveness on static tasks, such as road label classification, which does not rely on temporal information (Section V-E3). Therefore, we need to utilize **Toast** and **DyToast** separately for static and time-sensitive tasks for superior performance. Addressing such a limitation also represents a promising direction for future research.

VI. CONCLUSION

In this paper, we propose a novel framework, **Toast**, along with its advanced version **DyToast**, designed to enhance the integration of temporal dynamics for effective road network representation learning. The methods are designed to learn generic representations of both road segments trajectories, supporting a wide range of downstream applications, particularly those sensitive to temporal variations. Specifically, our framework is featured with two modules: a traffic-enhanced skip-gram module to incorporate traffic contexts into the learning process, and a trajectory-enhanced Transformer module to extract the travelling semantics encoded in trajectory data. These modules are further augmented by a unified approach based on trigonometric functions, enabling the capture of temporal dynamics from both time-dependent transportation graphs and trajectory data with fine-grained time interval knowledge. Our experiments demonstrate that the proposed framework consistently outperforms the state-of-the-art road network representation methods on three different tasks within dynamic settings.

ACKNOWLEDGMENTS

This research is supported in part by the Ministry of Education, Singapore, under its Academic Research Fund (Tier-2 grant MOE-T2EP20223-0004, Tier 2 grant MOE-T2EP20221-0013 and Tier 1 grant (RG20/24)). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

REFERENCES

- [1] J. Si, J. Yang, Y. Xiang, H. Wang, L. Li, R. Zhang, B. Tu, and X. Chen, “Trajbert: Bert-based trajectory recovery with spatial-temporal refinement for implicit sparse trajectories,” *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 4849–4860, 2024.
- [2] M. Wang, H. Jiang, P. Zhao, J. Li, J. Liu, G. Min, and S. Dustdar, “Ropriv: Road network-aware privacy-preserving framework in spatial crowdsourcing,” *IEEE Trans. Mob. Comput.*, vol. 23, no. 3, pp. 2351–2366, 2024.
- [3] M. Lv, D. Zeng, L. Chen, T. Chen, T. Zhu, and S. Ji, “Private cell-id trajectory prediction using multi-graph embedding and encoder-decoder network,” *IEEE Trans. Mob. Comput.*, vol. 21, no. 8, pp. 2967–2977, 2022.
- [4] H. Wang, Y. Li, C. Gao, G. Wang, X. Tao, and D. Jin, “Anonymization and de-anonymization of mobility trajectories: Dissecting the gaps between theory and practice,” *IEEE Trans. Mob. Comput.*, vol. 20, no. 3, pp. 796–815, 2021.
- [5] 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*, 2021.
- [6] J. Hu, C. Guo, B. Yang, and C. S. Jensen, “Stochastic weight completion for road networks using graph convolutional networks,” in *ICDE*, 2019, pp. 1274–1285.
- [7] Z. Cao, D. Zhao, H. Song, H. Yuan, Q. Wang, H. Ma, J. Tong, and C. Tan, “F\$^{\{3\}}\\$3vetrac: Enabling fine-grained, fully-road-covered, and fully-individual- penetrative vehicle trajectory recovery,” *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 4975–4991, 2024.
- [8] J. Wang, N. Wu, and W. X. Zhao, “Personalized route recommendation with neural network enhanced search algorithm,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5910–5924, 2022.
- [9] L. Ding, D. Zhao, Z. Wang, G. Wang, C. Tan, L. Fan, and H. Ma, “Learning to help emergency vehicles arrive faster: A cooperative vehicle-road scheduling approach,” *IEEE Trans. Mob. Comput.*, vol. 22, no. 10, pp. 5949–5962, 2023.
- [10] L. Du, X. Shi, Q. Fu, X. Ma, H. Liu, S. Han, and D. Zhang, “GBK-GNN: gated bi-kernel graph neural networks for modeling both homophily and heterophily,” in *The ACM Web Conference 2022*, 2022, pp. 1550–1558.
- [11] M. Gu, G. Yang, S. Zhou, N. Ma, J. Chen, Q. Tan, M. Liu, and J. Bu, “Homophily-enhanced structure learning for graph clustering,” in *CIKM*, 2023, pp. 577–586.
- [12] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: online learning of social representations,” in *KDD*, 2014, pp. 701–710.
- [13] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *KDD*, 2016, pp. 855–864.
- [14] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [15] Q. Li, Z. Han, and X. Wu, “Deeper insights into graph convolutional networks for semi-supervised learning,” in *AAAI*, 2018, pp. 3538–3545.
- [16] T. S. Jepsen, C. S. Jensen, T. D. Nielsen, and K. Torp, “On network embedding for machine learning on road networks: A case study on the danish road network,” in *IEEE BigData 2018*, 2018, pp. 3422–3431.
- [17] M. Wang, W. Lee, T. Fu, and G. Yu, “Learning embeddings of intersections on road networks,” in *SIGSPATIAL*, 2019, pp. 309–318.
- [18] M. Wang, W. C. Lee, T. Fu, and G. Yu, “On representation learning for road networks,” *ACM Trans. Intell. Syst. Technol.*, vol. 12, no. 1, pp. 11:1–11:27, 2021.
- [19] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, “Graph convolutional networks for road networks,” in *SIGSPATIAL*, 2019, pp. 460–463.
- [20] N. Wu, W. X. Zhao, J. Wang, and D. Pan, “Learning effective road network representation with hierarchical graph neural networks,” in *KDD*, 2020, pp. 6–14.
- [21] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, “Relational fusion networks: Graph convolutional networks for road networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 418–429, 2022.
- [22] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, 2019, pp. 4171–4186.
- [23] Y. Chen, X. Li, G. Cong, Z. Bao, C. Long, Y. Liu, A. K. Chandran, and R. Ellison, “Robust road network representation learning: When traffic patterns meet traveling semantics,” in *CIKM*, 2021, pp. 211–220.
- [24] Y. Chang, E. Tanin, X. Cao, and J. Qi, “Spatial structure-aware road network embedding via graph contrastive learning,” in *EDBT*, 2023, pp. 144–156.
- [25] Z. Mao, Z. Li, D. Li, L. Bai, and R. Zhao, “Jointly contrastive representation learning on road network and trajectory,” in *CIKM*, 2022, pp. 1501–1510.
- [26] L. Zhang and C. Long, “Road network representation learning: A dual graph-based approach,” *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 9, pp. 121:1–121:25, 2023.
- [27] S. Schestakov, P. Heinemeyer, and E. Demidova, “Road network representation learning with vehicle trajectories,” in *PAKDD*, 2023, pp. 57–69.
- [28] S. Wozniak and P. Szymanski, “hex2vec: Context-aware embedding H3 hexagons with openstreetmap tags,” in *SIGSPATIAL*, 2021, pp. 61–71.
- [29] H. Yuan, G. Li, and Z. Bao, “Route travel time estimation on A road network revisited: Heterogeneity, proximity, periodicity and dynamicity,” *Proc. VLDB Endow.*, vol. 16, no. 3, pp. 393–405, 2022.
- [30] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong, “A survey on trajectory data management, analytics, and learning,” *ACM Comput. Surv.*, vol. 54, no. 2, pp. 39:1–39:36, 2021.

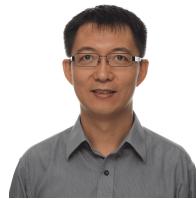
- [31] X. Li, G. Cong, and Y. Cheng, “Spatial transition learning on road networks with deep probabilistic models,” in *ICDE*, 2020, pp. 349–360.
- [32] Y. Liu, K. Zhao, G. Cong, and Z. Bao, “Online anomalous trajectory detection with deep generative sequence modeling,” in *ICDE*, 2020, pp. 949–960.
- [33] Z. Yang, H. Sun, J. Huang, Z. Sun, H. Xiong, S. Qiao, Z. Guan, and X. Jia, “An efficient destination prediction approach based on future trajectory prediction and transition matrix optimization,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 2, pp. 203–217, 2020.
- [34] H. Wu, Z. Chen, W. Sun, B. Zheng, and W. Wang, “Modeling trajectories with recurrent neural networks,” in *IJCAI*, C. Sierra, Ed., 2017, pp. 3083–3090.
- [35] B. Hui, D. Yan, H. Chen, and W. Ku, “Trajnet: A trajectory-based deep learning model for traffic prediction,” in *KDD*, F. Zhu, B. C. Ooi, and C. Miao, Eds., 2021, pp. 716–724.
- [36] M. Li, P. Tong, M. Li, Z. Jin, J. Huang, and X. Hua, “Traffic flow prediction with vehicle trajectories,” in *AAAI*, 2021, pp. 294–302.
- [37] S. Wu, X. Yan, X. Fan, S. Pan, S. Zhu, C. Zheng, M. Cheng, and C. Wang, “Multi-graph fusion networks for urban region embedding,” in *IJCAI*, 2022, pp. 2312–2318.
- [38] L. Zhang, C. Long, and G. Cong, “Region embedding with intra and inter-view contrastive learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9031–9036, 2023.
- [39] C. Yang and G. Gidofalvi, “Fast map matching, an algorithm integrating hidden markov model with precomputation,” *International Journal of Geographical Information Science*, vol. 32, no. 3, pp. 547–570, 2018.
- [40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS*, 2013, pp. 3111–3119.
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, 2017, pp. 5998–6008.
- [42] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, “struc2vec: Learning node representations from structural identity,” in *KDD*, 2017, pp. 385–394.
- [43] T. B. Brown, B. Mann, and et al., “Language models are few-shot learners,” *CoRR*, vol. abs/2005.14165, 2020.
- [44] J. Lu, D. Batra, D. Parikh, and S. Lee, “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks,” in *NeurIPS*, 2019, pp. 13–23.
- [45] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, “Videobert: A joint model for video and language representation learning,” in *ICCV*, 2019, pp. 7463–7472.
- [46] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A lite BERT for self-supervised learning of language representations,” in *ICLR*, 2020.
- [47] A. Pinkus, “Weierstrass and approximation theory,” *Journal of Approximation Theory*, vol. 107, no. 1, pp. 1–66, 2000.
- [48] B. Wang, E. D. Buccio, and M. Melucci, “Word2fun: Modelling words as functions for diachronic word representation,” in *NeurIPS 2021*, 2021, pp. 2861–2872.
- [49] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *NAACL-HLT*, 2018, pp. 464–468.
- [50] A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *NIPS*, 2007, pp. 1177–1184.
- [51] OpenStreetMap, [Online]. <https://www.openstreetmap.org/>.
- [52] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR*, 2017.
- [53] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *ICLR*, 2018.
- [54] Y. Chen, W. Huang, K. Zhao, Y. Jiang, and G. Cong, “Self-supervised learning for geospatial AI: A survey,” *CoRR*, vol. abs/2408.12133, 2024.
- [55] J. Jiang, D. Pan, H. Ren, X. Jiang, C. Li, and J. Wang, “Self-supervised trajectory representation learning with temporal regularities and travel semantics,” in *ICDE*, 2023, pp. 843–855.
- [56] Y. Liang, K. Ouyang, Y. Wang, X. Liu, H. Chen, J. Zhang, Y. Zheng, and R. Zimmermann, “Trajformer: Efficient trajectory classification with transformers,” in *CJMK*, 2022, pp. 1229–1237.
- [57] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” *Journal of machine learning research*, vol. 9, no. 11, 2008.



Yile Chen received the Ph.D. degree from Nanyang Technological University, Singapore, in 2022, and his B.Eng degree from Wuhan University, China, in 2018. He is currently a research fellow with the College of Computing and Data Science, Nanyang Technological University, Singapore. His current research interests include spatial data mining and data management.



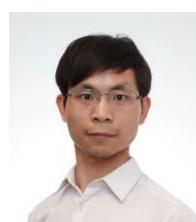
Xiucheng Li received the Ph.D. degree from the School of Computer Science and Engineering, Nanyang Technological University in 2020. He is an Assistant Professor at the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. Before joining HITSZ, he was a research fellow at the School of Computer Science and Engineering, NTU. His current research interests are implicit generative models and time series analysis.



Gao Cong is a Professor and the Division Head of Data Science at the College of Computing and Data Science, Nanyang Technological University. Prior to joining NTU, he worked with Aalborg University, Microsoft Research Asia, and the University of Edinburgh. He received the Ph.D degree from Nantional University of Singapore in 2004. His current research interests include geospatial data management and spatio-temporal data mining.



Zhifeng Bao received the Ph.D. degree in computer science from the National University of Singapore. He is currently a Professor with RMIT University and an honorary senior fellow with the University of Melbourne. His current research interests include data management and data governance.



Cheng Long (S'11-M'15-SM'22) is currently an Associate Professor at the College of Computing and Data Science, Nanyang Technological University. From 2016 to 2018, he was a lecturer at Queen's University Belfast, UK. He received his PhD degree from the Hong Kong University of Science and Technology, Hong Kong, in 2015, and his BEng degree from South China University of Technology, China, in 2010. His research interests are broadly in data management and data mining.