Robust Road Network Representation Learning: When Traffic Patterns Meet Traveling Semantics

Yile Chen[1,2], Xiucheng Li[1], Gao Cong[1,2], Zhifeng Bao[3], Cheng Long[2],
Yiding Liu[4], Arun Kumar Chandran[5], Richard Ellison[6]
[1]Singtel Cognitive and Artificial Intelligence Lab for Enterprises@NTU, Singapore
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3]RMIT University, Australia [4]Baidu Inc, China [5]NCS Pte. Ltd., Singapore [6]DataSpark, Singapore
{yile001@e.,xli055@e.,gaocong@,c.long@}ntu.edu.sg,zhifeng.bao@rmit.edu.au,liuyiding.tanh@gmail.com
arunkumar.chandran@ncs.com.sg,richard.ellison@dsanalytics.com

ABSTRACT

In this work, we propose a robust road network representation learning framework called Toast, which comes to be a cornerstone to boost the performance of numerous demanding transport planning tasks. Specifically, we first propose a *traffic context aware skip-gram* module to incorporate auxiliary tasks of predicting the traffic context of a target road segment. Furthermore, we propose a *trajectory-enhanced Transformer* module that utilizes trajectory data to extract traveling semantics on road networks. Apart from obtaining effective road segment representations, this module also enables us to obtain the route representations. With these two modules, we can learn representations which can capture multi-faceted characteristics of road networks to be applied in both road segment based applications and trajectory based applications. Last, we design a benchmark containing four typical transport planning tasks to evaluate the usefulness of Toast and comprehensive experiments verify that Toast consistently outperforms the state-of-the-art baselines across all tasks.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Road networks; Spatio-temporal data mining; Urban computing

ACM Reference Format:
Yile Chen[1,2], Xiucheng Li[1], Gao Cong[1,2], Zhifeng Bao[3], Cheng Long[2],, Yiding Liu[4], Arun Kumar Chandran[5], Richard Ellison[6]. 2021. Robust Road Network Representation Learning: When Traffic Patterns Meet Traveling Semantics. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482293

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482293

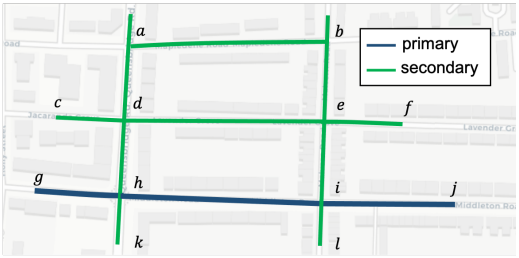

Figure 1: Road network example. Blue line denotes primary roads and green lines denote secondary roads.

1 INTRODUCTION

Road network, as a fundamental yet indispensable component in transportation systems, is closely related to numerous downstream transport planning tasks, including trajectory based tasks such as route inference [10, 19], and road segment based tasks such as traffic forecasting [5, 11]. Therefore, deriving effective representations that can capture intrinsic characteristics of the road network can directly boost the effectiveness of all these tasks. Since a road network is essentially a graph, a natural question to ask is whether we can apply graph representation learning models to address our problem. Unfortunately, it is not trivial due to two challenging issues.

The first is the *discrepancies* with regard to specified assumptions between common graphs and road networks. Most previous studies focus on citation or social network graphs and design methods based on some well-explored assumptions on these graphs [9, 26, 29], which may not hold in road networks. For example, a citation graph usually exhibits network homophily [26], which means interconnected nodes are more similar than distant nodes. However, spatially-neighboring road segments might not necessarily show similar traffic patterns on road networks. In Figure 1, road segments *dh*, *gh*, *hi*, *hk* are connected to each other but primary roads usually have different *traffic patterns*, e.g., traffic volume, with secondary roads since primary roads are travelled more frequently.

The second is the *feature uniformity* issue. Features on a road network, such as road type and lane number, are often shared across spatially-close nodes. More precisely, since a city manifests different functionalities for different sub-regions, such as commercial area and residential area, it is often the case that some fractions of a road network have the same features. This unique property in road networks would dampen the performance of standard graph

representation learning methods, especially graph neural networks (GNN) [42]. We illustrate this with an example in Figure 1, where all the connected road segments to the target road segment *de* share the same features (road type). The same cases happen for road segments *cd*, *ad*, *ab*, etc. The aggregation operation, which is the core part in GNN, reduces to an identity transformation for these road segments so it would lead to the issue similar to over-smoothing [18] in GNN. Notably, the two issues, *discrepancies* and *feature uniformity*, are two different perspectives of potential issues that can co-exist on road networks. For example, road segment *de* shares the same features with neighboring road segments, while it also has more traffic volume compared to *cd*, *ad*, *ab* since path $[c, d, e, f]$ is a straight route but others are detours.

Instead of serving a specific task such as traffic inference, our target is the same as that in [9, 26] for general graph representation learning that can be utilized in various applications. In other words, we expect the learned representations to be robust and generic when serving various types of road network applications, as highlighted at the beginning of this section.

Among all recent efforts in road network representation learning [14, 33, 34, 41], the models in [33, 34] aim to learn road intersection representations for road networks. It considers to capture road network structure and meanwhile integrate extra information if available, such as whether two intersections have the same tag (e.g., stop sign tag). However, this model follows the homophily assumption, so it cannot fully address the first issue. The models in [14, 41] adapt GNN to road networks in order to learn road segment representations, and hence it suffers from the second issue. Even worse, these models only focus on capturing partial characteristics such as road network structure, thus failing to learn effective representations that contain multi-faced knowledge about road networks. In summary, existing representation learning methods for both standard graphs and road networks suffer from the above two issues, and in this work we aim to address both of them.

We argue that in order to achieve robust and generic representations for road networks, two types of characteristics, namely **traffic patterns** and **traveling semantics**, need to be captured to handle the previously discussed two issues. In particular, traffic patterns (such as traffic volumes) could be regarded as a signal to overcome the limitations of assumptions made for common graphs (*discrepancies*). Meanwhile, traveling semantics (such as transition patterns) could be incorporated to identify differences for road segments that share the same features (*feature uniformity*). Taking Figure 1 as an example, transition patterns would tell that the path $[c, d, e, f]$ is more frequently traveled than the detour path $[c, d, a, b, e, f]$ between c and f, which demonstrates the dependencies among road segments. Furthermore, these two types of characteristics, which serve as the most generic features on road networks, should be properly encoded and integrated so that they can precisely enrich the desired multi-faced knowledge about road networks.

To this end, we propose a new framework called Toast (**T**raffic c**o**ntext **a**ware **s**kip-gram with trajectory-enhanced **t**ransformer), to learn robust representations that can capture both traffic patterns and traveling semantics on road networks. Different from the previous frameworks [14, 41], we propose to extend the skip-gram model [24] to enable the model awareness of traffic patterns by incorporating an auxiliary traffic context prediction objective. By doing this, the model is able to not only encode the graph structure of a road network with the original skip-gram objective, but also distinguish road segments in terms of traffic patterns, thus addressing the first issue *discrepancies*. To address the second issue *feature uniformity*, we propose to utilize trajectory data to extract traveling semantics for indistinguishable road network fractions caused by *feature uniformity*. As has been shown in previous studies [7, 40], trajectory data contains rich traveling semantics that can be modeled to enhance the effectiveness of road network representations. To achieve this goal, inspired by recent advances of Transformer model (i.e., BERT) [6] on text modeling, we propose to employ such an architecture to capture transition patterns embodied in trajectory data into the representations. Due to the inadequacy of the training tasks in BERT on road network settings, we further design two novel training tasks, *route recovery* and *trajectory discrimination*, to effectively encode the traveling semantics for road networks. Finally, we organize the aforementioned two modules in a unified way so that they focus on encoding complementary aspects of road network characteristics. These two modules are both based on self-supervised training paradigms in which traffic patterns and travelling semantics are directly treated as the training objective without further task-specific labeling information. It ensures that these characteristics are well encoded into representations, thus achieving robust performance in various applications.

Moreover, in addition to learning representations of road segments, a side benefit is that we can get a representation of a trajectory and consider it as a route representation. Such capability further enhances the utility for trajectory based tasks, such as trajectory similarity search.

To summarize, our contributions are as follows:

- We propose a novel road network representation learning method called Toast, which is featured with two new modules, a traffic context aware skip-gram module and a trajectory-enhanced Transformer module. Toast is able to capture both traffic patterns and traveling semantics on road networks. To the best of our knowledge, this is the first work that models trajectory sequences in learning road network representations and is able to address the two aforementioned concerns.
- The carefully designed framework allows us to learn robust and generic representations for road networks to benefit both road segment based applications and trajectory based applications. We consider four downstream applications and design the corresponding experimental settings. These tasks and experimental settings could serve as a benchmark for evaluating the representations of road networks.
- We conduct extensive experiments on four downstream applications and the results demonstrate that Toast consistently outperforms the state-of-the-art road network representation methods and representative graph representation methods across all these applications.

2 RELATED WORK

Representation learning on graphs. Representation learning on graphs [8] has received extensive attention for representing each

node as a low dimensional vector. Existing studies can be categorized based on various criteria. Some methods focus on capturing different graph properties, such as proximity and homophily. In particular, Deepwalk [26] and node2vec [9] employ random walk on the graph to get node sequences which are treated as sentences, and then skip-gram model, originally proposed to learn word embeddings [24], is applied to learn node representations. LINE [29] is proposed to preserve first and second-order proximity by explicitly optimizing the corresponding objectives. Moreover, when other data sources are available on graphs, such as text content [45] on nodes and community structure [39], specialized methods are proposed to incorporate such extra information to enhance representations. More recently, graph neural networks (GNN) [42], which aim to extend deep neural networks to deal with arbitrary graph-structured data, have been introduced for graph representation learning. GNN based models generate node representations by exchanging and aggregating features from neighborhoods, and different methods are proposed to explore different effective aggregation operations [12, 16, 31]. However, the superior performance usually requires the graph nodes to contain rich features which are diverse in neighborhoods, which is not the case for road networks. As discussed in Section 1, these graph representation learning methods are not designed for road networks and fail to capture unique characteristics in road networks such as traffic patterns.

Road network modeling. Road network modeling can facilitate applications of intelligent transportation system, such as traffic inference and forecasting, spatial query processing, and region functionality discovery, to name a few. In traffic inference and forecasting [11, 13], road network provides graph topology information, which specifies how the traffic status would propagate among road segments. In spatial query processing [25, 38], various methods are proposed to support particular search operations on road network, which improves the quality of traffic management service. In region functionality discovery [47], road network serves as a signal to help mine the underlying semantics of regions. Recently, some methods have been proposed to extend graph representation learning techniques to road networks [14, 33, 34, 41]. However, none of these methods propose to learn from trajectory data, which contains rich traveling semantics about road networks. Moreover, they fail to thoroughly address the two issues discussed in Section 1.

Trajectory mining. Trajectory data contains rich information about the behaviors of moving objects, and has been exploited for real-world applications [35]. Several trajectory management systems are built to optimize data indexing and storage [37], to support different operations, such as similarity computation [7, 21] and clustering [36]. On the other hand, the patterns encoded in trajectories enable us to build effective models in downstream applications, such as travel time estimation [20, 46], route recovery and inference [10, 19], and anomaly trajectory detection [22]. In these applications, road network explicitly provides structural constraints for trajectory data, while trajectory data in some sense implicitly reflects latent patterns for road networks. This motivates us to leverage trajectory data in learning representations for road networks.

3 PROBLEM FORMULATION

We start with our problem statement and then describe an overview of our proposed framework.

3.1 Problem Definitions

Definition 3.1. **Road Network.** A road network is represented as a directed graph $G = (\mathcal{V}, \mathcal{E}, C_{\mathcal{V}})$, where $\mathcal{V}$ is a set of vertices, each vertex v representing a road segment, $\mathcal{E}$ is a set of edges, each $e = (u, v)$ representing the intersection between road segments u and v, and $C_{\mathcal{V}}$ is a set of features on road networks.

Definition 3.2. **Trajectory.** A trajectory T is a sequence of sampled points $[p_i]_{i=1}^{|T|}$ from the underlying route of a moving object, where each point p_i corresponds to a coordinate of latitude and longitude.

Definition 3.3. **Route.** A route $\mathbf{r} = [r_i]_{i=1}^{n}$ is a sequence of adjacent road segments, where $r_i \in \mathcal{V}$ represents the i-th road segment.

In our study, given a road network G, a trajectory T is first mapped onto the road network to get the underlying route $\mathbf{r}$ by a map matching algorithm [44]; we represent the road segment set as $\mathcal{V}$ to be consistent with the notation usage in graph learning.

Problem Statement. Given a road network $G = (\mathcal{V}, \mathcal{E}, C_{\mathcal{V}})$ and a trajectory database $\mathcal{D} = \{T^{(i)}\}_{i=1}^{|\mathcal{D}|}$, we aim to 1) learn a low-dimensional vector representation $\{\mathbf{u}_v\}_{v \in \mathcal{V}}$ for road segments, and 2) derive the representation $\mathbf{u}_\mathbf{r}$ of any given route $\mathbf{r}$ on the road network.

It is worth noting that our target is to learn robust and generic representations for a road network, such that the derived road segment representations and route representations could be utilized in various road segment based applications and trajectory based applications, respectively.

3.2 Framework Overview

In this section we will illustrate how our Toast framework is able to address the two issues outlined in Section 1 (i.e., discrepancies and feature uniformity), towards robust road network representations. An overview of Toast is shown in Figure 2.

First, apart from following the assumptions of common graphs (e.g., homophily, structural equivalence), we propose to distinguish discrepancies among road segments. To achieve this, we extend the skip-gram model [24], which is flexible in producing node representations based on various structural assumptions for graph data, to capture traffic patterns (e.g., traffic volume). In addition to the original skip-gram objective which is to predict the context neighbors of the target road segment, we introduce auxiliary tasks of predicting traffic-related context features (e.g., speed limit) in a self-supervised manner. Such a multi-task learning paradigm enables the representations to not only encode graph structures, but also discriminate between various traffic patterns which are indicated by context features.

Second, a unique property of road network is that it usually has many sub-regions with uniform features; unfortunately, the performance of standard graph representation learning methods, especially GNN, suffer from such feature uniformity. Furthermore, road segments within such a sub-region (e.g., residential area) of

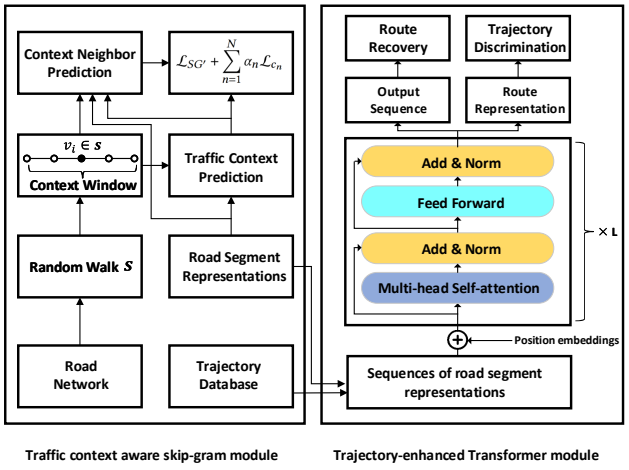

Figure 2: Framework overview

a road network also tend to be similar in terms of traffic patterns, making the situation even worse. To address this issue, we propose to learn from trajectory data to extract travelling semantics on road networks, including transition patterns and high-order dependencies between distant regions. To achieve this goal, we adopt a stacked bidirectional Transformer architecture [6] to model trajectory data. In particular, we design two novel and effective self-supervised training tasks for road network settings, *route recovery* and *trajectory discrimination*, to further tune the representations. Specifically, in the *route recovery* task, we randomly mask a partial sequence of road segments in a route, and then recover the masked sequence based on the remaining part of the route. In the *trajectory discrimination* task, given a valid route on a road network, we aim to discriminate whether it is a trajectory or a random walk on road network.

With the proposed techniques, Toast can encode multi-faceted yet mutually enhanced characteristics about road networks into the final representations. Moreover, Toast is able to produce the representation of a sequence, which allows us to obtain a route representation for any trajectory. In this way, our framework produces generic representations that 1) could be applied in both road segment based and trajectory based downstream applications, and 2) achieve effective performance across different applications.

4 TOAST FRAMEWORK

We present our Toast framework. We start with preliminaries of the skip-gram model and then discuss the extended skip-gram model with an auxiliary traffic context prediction objective. Next we describe the trajectory-enhanced Transformer module. Last, we present our proposed training tasks and illustrate how they capture information encoded in trajectory data into representations.

4.1 Preliminary: Skip-gram Model

The skip-gram model was first proposed in word2vec [24] to learn embeddings for words[1]. It is widely adopted in node representation learning methods later by viewing nodes in a graph as words in a document. These methods employ a set of random walks $\mathcal{S}$ on a graph and treat each random walk as a sentence. The model is trained to maximize the likelihood of observing the neighborhood nodes within a context window given the target node, which equals to minimizing the following loss function:

$$\begin{aligned}\mathcal{L}_{SG} &= -\sum_{v_i \in \mathbf{s}} \sum_{v_j \in \mathcal{N}(v_i)} \log p\left(v_j | v_i\right) \\ &= -\sum_{v_i \in \mathbf{s}} \sum_{v_j \in \mathcal{N}(v_i)} \log \frac{\exp\left(\mathbf{u}_i^T \mathbf{w}_j\right)}{\sum_{v'_j \in \mathcal{V}} \exp\left(\mathbf{u}_i^T \mathbf{w}'_j\right)},\end{aligned} \tag{1}$$

where $\mathbf{u}_i$ is the target embedding for node v_i, $\mathbf{w}_j$ is the context embedding for node v_j, $\mathcal{N}(v_i)$ are the context neighbors of node v_i, and $\mathbf{s}$ is a random walk sequence from the random walk set $\mathcal{S}$. By optimizing this objective, the final node representations could capture the structural properties (e.g., homophily) via various random walk sampling strategies [9, 27].

4.2 Auxiliary Traffic Context Prediction Objective

As discussed in Section 3.2, Toast aims to not only capture the structural assumptions of common graphs, but also incorporate traffic patterns in the representations. To achieve this, we propose to extend the skip-gram model by introducing auxiliary traffic context prediction tasks. For instance, there are some traffic context features available for road segments, such as speed limit and road type. We regard them as auxiliary context information that indicates the traffic patterns of the corresponding road segments. Based on this, given a target node (road segment) and its context neighbors, our key idea is to first determine the traffic context of the target node, and then predict the context neighbors. Specifically, to perform traffic context prediction for a target road segment, we first apply binarization to the selected features that infer traffic patterns. For example, assuming we choose road type c_n from traffic context feature set $\{c_n\}_{n=1,2,...N}$ as a particular feature which has $|c_n|$ possible options, it is processed into a $|c_n|$-dimensional label vector where each dimension is 0 or 1, representing the existence of one particular option of a target road segment. Formally, given a target road segment v_i and its corresponding N types of binarized traffic context features $\boldsymbol{\pi}(v_i) \overset{\text{def.}}{=} \{c_n^i\}_{n=1,2,...N}$, then for any context feature c_n we aim to minimize the binary cross entropy loss function:

$$\begin{aligned}\mathcal{L}_{c_n} &= -\sum_{v_i \in \mathbf{s}} \sum_{j=1}^{|c_n|} c_{nj}^i \log p\left(c_{nj}^i | v_i\right) + \left(1 - c_{nj}^i\right) \cdot \log\left(1 - p\left(c_{nj}^i | v_i\right)\right) \\ &= -\sum_{v_i \in \mathbf{s}} \sum_{j=1}^{|c_n|} c_{nj}^i \log \sigma(\mathbf{u}_i^T \mathbf{c}_{nj}) + (1 - c_{nj}^i) \cdot \log(1 - \sigma(\mathbf{u}_i^T \mathbf{c}_{nj}))\end{aligned} \tag{2}$$

where c_{nj}^i is the j-th entry of the n-th binarized feature c_n for node v_i, $\mathbf{u}_i$ is the target embedding for node v_i, $\mathbf{c}_{nj}$ is the feature embedding for c_{nj} that is shared across road segments, and σ denotes the sigmoid function.

Here, node embeddings are optimized to produce accurate predictions on both traffic context and context neighbors, which are

[1]We use embeddings and representations interchangeably in this paper.

more adequate in road network setting than considering context neighbors only in normal graph embedding. Moreover, these prediction tasks are organized in an ordered way such that we are able to utilize traffic context to further enhance the prediction of context neighbors. In other words, when predicting the context neighbors, instead of being conditioned only on the target road segment v_i via the original skip-gram objective, we modify this objective to be

$$\begin{aligned}\mathcal{L}_{SG'} &= -\sum_{v_i \in \mathbf{s}} \sum_{v_j \in \mathcal{N}(v_i)} \log p\left(v_j | v_i, \tilde{\boldsymbol{\pi}}(v_i)\right) \\ &= -\sum_{v_i \in \mathbf{s}} \sum_{v_j \in \mathcal{N}(v_i)} \log \frac{\exp\left(\tilde{\mathbf{u}}_i^T \tilde{\mathbf{w}}_j\right)}{\sum_{v'_j \in \mathcal{V}} \exp\left(\tilde{\mathbf{u}}_i^T \tilde{\mathbf{w}}'_j\right)}\end{aligned} \tag{3}$$

Here, $\tilde{\boldsymbol{\pi}}(v_i) \overset{\text{def.}}{=} \{\tilde{c}_n^i\}_{n=1,2,...,N}$ with $\tilde{c}_n^i \overset{\text{def.}}{=} [\sigma(\mathbf{u}_i^T \mathbf{c}_{nj})]_{j=1}^{|c_n|}$ is the n-th predicted traffic context of road segment v_i; $\tilde{\mathbf{u}}_i$ is the traffic-enhanced target embedding for v_i, which is a concatenation of the original target embedding $\mathbf{u}_i$ and all the predicted traffic context $\tilde{\boldsymbol{\pi}}(v_i)$; $\tilde{\mathbf{w}}_j$ is the corresponding context embedding for node v_j.

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

$$\mathcal{L} = \mathcal{L}_{SG'} + \sum_{n=1}^{N} \alpha_n \mathcal{L}_{c_n} \tag{4}$$

where α_n are the hyperparameters to control the weight of auxiliary tasks. Compared to the original skip-gram model, we encode more semantic information (i.e., traffic patterns) into the representations with the help of our proposed auxiliary tasks. At the same time, the context neighbor prediction would also benefit from the knowledge of traffic context. As a result, this multi-task learning paradigm would produce more effective representations for a road network.

4.3 Bidirectional Self-attention Network

4.3.1 Network Architecture. To handle the feature uniformity issue faced by existing representation learning methods, we propose a novel trajectory-enhanced Transformer module to extract transition patterns and high-order dependencies on road networks. It has been proved that a stacked bidirectional self-attention network (i.e., stacked Transformer) is powerful in modeling text sequences to learn semantically useful word representations for numerous downstream tasks [6]. Inspired by the observation that trajectory is a type of sequence data, we propose to adopt such an architecture to learn representations for road networks. Next we describe the modeling process in a bottom-up fashion.

Input Embedding Layer. The road segment representations obtained from the first module serves as the input embeddings of our stacked bidirectional self-attention network. In contrast to recurrent neural networks (RNN) that process the inputs sequentially, self-attention network operates on the input tokens in parallel using the self-attention mechanism, and thus they are agnostic to the order of the input tokens. Hence, to preserve the sequential information of trajectory, we inject learnable *positional embeddings* into the input representations. Specifically, we construct the input representations as

$$\mathbf{x}_i = \mathbf{u}_i + \mathbf{p}_i \tag{5}$$

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

In this manner, the injected positional embeddings can help the self-attention network to be aware of the input order rather than treating them as a set of unordered road segments. Also, they enable the model to learn high-level semantics in the sense that one road segment might play a different role when it appears at different locations of a trajectory.

Multi-head Self-attention. Attention mechanism has been successfully applied in a wide variety of tasks ranging from machine translation to image captioning [3, 43]. In particular, self-attention enables the transformation of a sequence without relying on the external information. We follow the setting of scaled inner-product form of attention mechanism, which can be described as mapping a query and a set of key-value pairs to an output vector representation [30]. Formally, it is defined as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{6}$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are the stacked query, key, and value vectors of dimension d_k, d_k, d_v respectively. Notably, self-attention mechanism adopts a position-insensitive transformation, and thus the positional embedding proposed in the Input Embedding Layer is essential for the self-attention network to incorporate the input order information.

In our work, we adopt multi-head self-attention [30] to model trajectory sequences. Given the input representations $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_M}] \in \mathbb{R}^{M \times d_{in}}$ of a trajectory that consists of M road segments, they are mapped to output representations $\mathbf{Z} = [\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_M}] \in \mathbb{R}^{M \times d_{out}}$. Specifically, the input representations are projected into h subspaces with different learnable parameters as queries, keys, and values. Each projection indicates a "head" that allows the model to jointly attend to information from several independent subspaces. Then self-attention is applied in each subspace, followed by concatenation and another projection, to produce the output representations:

$$\begin{aligned}\mathbf{Z} &= \text{MH-Attn}(\mathbf{X}) = [head_1, \ldots, head_h] \cdot \mathbf{W}^O \\ head_i &= \text{Attention}\left(\mathbf{X}\mathbf{W}_i^Q, \mathbf{X}\mathbf{W}_i^K, \mathbf{X}\mathbf{W}_i^V\right)\end{aligned} \tag{7}$$

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 bidirectional interactions across different positions by multi-head self-attention, the output representations are sent to a fully connected feed-forward network. More precisely, the output representations $\mathbf{Z}$ are passed through a feed-forward network as follows:

$$\text{FFN}(\mathbf{Z}) = \Phi\left(\mathbf{Z}\mathbf{W}_1 + \mathbf{b}_1\right)\mathbf{W}_2 + \mathbf{b}_2 \tag{8}$$

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

Model Stacking. It is usually beneficial to learn more complex transition patterns in trajectory data by stacking multiple layers, each of which is composed of two sub-layers: multi-head self-attention and position-wise feed forward network described above. To alleviate possible training difficulty caused by the increasing depth of more stacked layers, we follow [30] to employ the residual connection

on each of the two sub-layers, followed by layer normalization [2]. It can be formally expressed as

$$\begin{aligned} \mathbf{Z}' &= \text{LayerNorm}(\mathbf{X} + \text{MH-Attn}(\mathbf{X})) \\ \mathbf{X}' &= \text{LayerNorm}(\mathbf{Z}' + \text{FFN}(\mathbf{Z}')) \end{aligned} \tag{9}$$

where LayerNorm denotes layer normalization and $\mathbf{Z}'$ denotes the final output of a layer, which is also the input for next layer.

4.3.2 Model Training. Despite the power of the bidirectional self-attention network, a critical question is how to ensure that the representations contain traveling semantics on road networks. In such a network architecture, it is usually the case that a well-designed loss function plays a vital role in achieving desired properties for various domains (e.g., language [6], image [23], video [28]) as it provides useful signals to guide the parameter tuning process. In other words, we need to design appropriate training tasks which call for the knowledge of traveling semantics on road networks to achieve effective performance. In this way, as the loss decreases with model training, the representations are gradually tuned to encode the required knowledge of traveling semantics.

In previous work where transformer architecture is employed, BERT [6] designs two self-supervised training tasks, namely masked language modeling (MLM) and next sentence prediction (NSP), to learn representations for various natural language processing tasks. Although these two tasks lead to successful representations, they could not achieve our target under road network settings.

First, in the MLM task, each word in a sequence is randomly masked with a certain probability (e.g., 15%), and then the model is asked to predict those masked words. Due to high flexibility and complexity of human languages, there are often a large number of syntactically appropriate words that could fill a masked location, and thus the model will be forced to learn the semantically meaningful word to best fit that position, which implies that high-level semantic meanings of the words would have to be captured by their representations. However, this is not the case for road segments prediction in a trajectory, because two consecutive road segments in a trajectory must be connected in a road network. In this case, given the left and right context road segments, the masked road segment could be trivially inferred from the knowledge of graph structure, as it is the only road segment that makes the sequence a valid route. Given that graph structure is well captured by the skip-gram objective, this task will not provide any new information for the representation learning.

Second, in the NSP task, we choose a sentence pair as input, and the latter sentence is either the next sentence of the former one or a random sentence from the corpus. Then the model is trained to distinguish which group this pair belongs to. However, this task will not take effect in our problem because it does not provide any essential information in road network settings.

To this end, we propose two new training tasks: *route recovery* and *trajectory discrimination*, to effectively encode the travelling semantics of trajectory data into their representations.

Route Recovery. Different from the MLM task where every single word is randomly masked, we mask a consecutive sequence of road segments to make a trajectory as a partially observed route. In particular, given a route we randomly mask 20% consecutive road segments in the sequence. In this task, we could not trivially recover the masked road segments by the awareness of graph structure. Instead, it requires the representations to capture more complex transition patterns and pick the most possible option for the masked parts. The model is trained to optimize the cross entropy loss between masked road segments and the predicted ones.

Trajectory Discrimination. We propose to train the model to judge whether a given trajectory is a real trip or not. Specifically, the real trips are sampled from our trajectory databases, and the fake trips are generated by random walks sampled on the road network. Then, we train the model to minimize the prediction error. The purpose of this task is two-fold. *First*, it is another way to enable the model to capture transition patterns. After training, the model is able to identify fake trips by observing that some sub-sequences do not follow the normal transition patterns. *Second*, since trajectories are scattered around a road network, this task would provide an overall understanding of traveling semantics on road network. In particular, there can exist some frequent trips between two distant regions; by correctly distinguishing such trips, high-order dependencies and correlations across distant road segments could also be well captured. The whole training procedure of Toast is described in Algorithm 1.

Algorithm 1: Pseudocode for training Toast

Input: Road network $G = (\mathcal{V}, \mathcal{E}, C_{\mathcal{V}})$; trajectory databases $\mathcal{T}$; epoch number e; target and context road network embeddings $\mathcal{U} = \{\mathbf{u}_i\}_{i=1:|\mathcal{V}|}$, $\mathcal{W} = \{\mathbf{w}_i\}_{i=1:|\mathcal{V}|}$; feature embeddings $C = \{\mathbf{C}_n\}_{n=1:N}$ where $\mathbf{C}_n = [\mathbf{c}_{nj}]_{j=1:|c_n|}$

```
for epoch ∈ 0, 1, ... e do
    Perform a set of random walks S on G;
    for s ∈ S do
        Calculate L = L_SG' + Σ_{n=1}^{N} α_n L_{c_n};
        Update U, W, C by minimizing L via
          backpropagation;
    end
    for t ∈ T do
        Sample a training task described in Sec. 4.3.2;
        Calculate loss L_t for the sampled task;
        Update U by minimizing L_t via backpropagation;
    end
end
```

Remarks. The road segment representations are encoded to capture multi-faceted characteristics of a road network, and hence become effective in numerous downstream applications. Furthermore, we can extract the route representation of a trajectory from the stacked self-attention network by pooling the final representations on the top layer, or following [6] to insert a placeholder in the first position throughout training tasks and take it as the route representation.

5 EXPERIMENTS

In this section, we evaluate the usefulness of road network representations learned by Toast. Similar to the experimental objectives of previous work on road network representation learning methods [14, 33, 34, 41], the objective of this experimental study is to

evaluate the utility of the road network representations. In other words, we aim to evaluate whether the proposed model can better capture underlying characteristics of road networks than the existing methods [14, 33, 34, 41].

5.1 Datasets

We use the road network and trajectories of two cities, *Chengdu* and *Xi'an*. The road networks are obtained from OpenStreetMap[2], and the trajectories are obtained from public datasets released by DiDi Chuxing[3]. To better 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 [44] to transform GPS records of trajectories into sequences of road segments. The statistics of the datasets are shown in Table 1.

Table 1: Statistics of the datasets

Dataset	#Road Segments	#Edges	#Trajectories
Chengdu	4,885	12,446	677,492
Xi'an	5,052	13,660	373,054

5.2 Downstream Tasks & Baseline Methods

Since Toast is able to produce both road segment representations and route representations, we evaluate its effectiveness on both road segment based applications and trajectory based applications.

5.2.1 Road segment based applications. We consider two typical tasks: 1) road label classification, and 2) traffic inference. We compare Toast with seven strong competitors, namely DW [26], GAE [15], node2vec [9], GraphSAGE [12], RFN [14], IRN2Vec [33] and HRNR [41]. Here, DW, node2vec, GAE and GraphSAGE are general graph node representation learning methods; RFN, IRN2Vec, and HRNR are the recent representation learning methods that are designed for road networks.

Note that we compare with the state of the art representation learning methods for road networks, and several representative graph representation learning methods. This is consistent with the objective of the experiments as discussed in the beginning of Section 5. We do not consider the specialized methods that are developed for a specific application because 1) our target is to evaluate the effectiveness of the learned representations, and thus we use the same task-specific components and make them as simple as possible for all the compared methods to reduce the effects of other elements; 2) specialized methods are highly dependent on the task-specific model designs which usually take more factors as input and it is not a fair comparison.

Details of the competitors and their parameter settings are provided as follows.

- DW [26]: It first transforms the network into node sequences by truncated random walks, and then applies the skip-gram model on the node sequences to learn representations. The walk length, the number of walks per node, and the window size are set to be 30, 25, and 5 respectively.
- node2vec [9]: It is a variant of DW and employs a biased random walk procedure to explore neighborhood of a node, which captures both the local and global structural properties of a network. The walk length, the number of walks per node, and the window size are set to be the same as in DW. The biased random walk parameter p and q are tuned in the set of {1/8, 1/4, 1/2, 1, 2, 4, 8}.
- GAE [15]: It uses graph convolutional network (GCN) to learn node representations, and is trained to reconstruct the original road network network structure. The model consists of two GCN layers followed by an MLP layer.
- GraphSAGE [12]: It is a GNN based framework that learns node representations by sampling and aggregating features from nodes' local neighborhoods. We apply the unsupervised training paradigm in the original paper. The depth is set to be 2 and mean aggregator is applied as different aggregators' performance are relatively comparable in our datasets.
- RFN [14]: It aims to learn representations based on node-relational and edge-relational views of road network graphs, where message passing and interaction are performed for on both views. We use network reconstruction as the training objective and the number of layer is set to be 3.
- IRN2Vec [33]: It applies a multi-task learning framework which applies skip-gram model to predict the information of geo-locality and intersection tags given road segment context. We set the parameters the same as that in DW.
- HRNR [41]: It is a GNN based framework adapted to model different semantic levels of road networks, namely functional zones, structural regions and road segments in a hierarchical way. We follow the experiment settings in the original paper.

For the task of road label classification, we choose road type as the label, and the total label number is 5 in both datasets. For the task of traffic inference, we choose average speed on road segment as the inference objective. In evaluation stage, the learned road segment representations are input to a one-vs-rest logistic regression classifier and a linear regression model for road label classification and traffic inference, respectively. We apply 5-fold cross validation to evaluate the performance of all the methods.

5.2.2 Trajectory based applications. Our framework is evaluated on two typical tasks: 1) trajectory similarity search, and 2) travel time estimation. Given a trajectory, route representation is derived first, and then sent to a task-specific component to get the final prediction result. Specifically, for the task of trajectory similarity search, we use route representations to calculate the similarity score between trajectories; for the task of travel time estimation, we adopt a multi-layer perceptron as a task-specific component, which takes a route representation as input and outputs the estimated travel time. It is noted that we do not apply task-specific complex models or use other form of inputs (e.g., external data sources) because our target is to verify the effectiveness and ubiquity of our model with other methods in the same setting. We compare our framework with the following route representation baselines.

- para2vec [17]: An embedding methods to learn paragraph representations. Here we treat a trajectory as a paragraph and derive its representation. We consider each trajectory as a paragraph and train the whole trajectory dataset for 20 epochs.

[2]https://www.openstreetmap.org/
[3]https://outreach.didichuxing.com/research/opendata/en

Table 2: Results for road segment based applications

Task	Road Label Classification				Traffic Inference			
	Chengdu		Xi'an		Chengdu		Xi'an	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	MAE	RMSE	MAE	RMSE
DW	0.522	0.493	0.552	0.524	7.32	9.14	6.78	8.57
node2vec	0.524	0.495	0.586	0.559	7.12	9.00	6.41	8.22
GAE	0.432	0.328	0.447	0.339	6.91	8.72	6.41	8.39
GraphSAGE	0.452	0.324	0.466	0.347	6.48	8.52	6.12	7.98
RFN	0.516	0.484	0.577	0.570	6.89	8.77	6.57	8.43
IRN2Vec	0.497	0.458	0.531	0.506	6.52	8.52	6.60	8.59
HRNR	0.541	0.527	0.631	0.609	7.03	8.82	6.52	8.45
Toast	**0.602**	**0.599**	**0.692**	**0.659**	**5.95**	**7.70**	**5.71**	**7.44**

Table 3: Results for trajectory similarity search

	Chengdu		Xi'an	
	MR	HR@10	MR	HR@10
para2vec	216.92	0.251	279.38	0.205
t2vec	46.17	0.781	38.67	0.806
LCSS	67.72	0.487	83.94	0.469
EDR	458.20	0.174	529.74	0.119
Fréchet	21.17	0.847	22.79	0.894
Toast	**10.10**	**0.885**	**13.71**	**0.905**

Table 4: Results for travel time estimation

	Chengdu		Xi'an	
	MAE	RMSE	MAE	RMSE
para2vec	220.45	302.72	244.73	345.49
t2vec	165.18	240.72	207.56	311.04
Road-Pool	151.80	223.02	185.47	293.82
Toast	**127.80**	**190.86**	**175.68**	**265.09**

- t2vec [21]: This is a state-of-the art method of learning trajectory representation for similarity computation. It is an encoder-decoder framework which is trained to reconstruct the original trajectory. We use a sequence of road segment to represent a trajectory and apply two layers of LSTM for both encoder and decoder.

In the task of trajectory similarity search, we directly use route representations to calculate the similarity score between trajectories for all the baselines. Besides, we also compare with three widely adopted trajectory similarity measures, namely LCSS [32], EDR [4] and Fréchet [1]. In the experiment, we randomly select 5,000 trajectories to serve as queries. To evaluate the effectiveness of different models, for each query we generate a similar trajectory as a ground truth. Specifically, given a trajectory, we choose a start and an end point, then we make a detour between these two points by travelling along another path deviating from the original one, and the average detour length is 8.8% of the whole path. We randomly sample 100,000 trajectories together with the generated ground truth trajectories as the trajectory database for query processing.

For the task of travel time estimation, we also apply another method denoted by Road-Pool, which applies average pooling on the road segment representations for a trajectory to get its route representation. Road-Pool can be considered as a variant of Toast without the learned road segment representations going through stacked self-attention layers. In the experiments, we randomly sample 80,000 trajectories for training and 20,000 trajectories for testing.

Evaluation Metrics. For road label classification, Micro-F1 and Macro-F1 are used to measure the classification accuracy. For traffic inference and travel time estimation tasks, MAE (Mean Average Error) and RMSE (Root Mean Square Error) are used to measure the closeness between the predicted value and the real value. For trajectory similarity search, we treat it as a ranking problem and adopt Mean Rank (MR) and Hit Ratio@10 (HR@10) for evaluation.

Parameter Setting. We set the size of representations for both road segment and route to be 128 across all methods. We choose road type prediction as an auxiliary traffic context prediction task where the weight is set to 1, and we apply negative sampling [24] to improve training efficiency. In order to avoid data leakage, we do not adopt this auxiliary task when the task is road label classification. Multi-head self-attention layer is stacked 2 layers and the head number is set as 4. We choose the batch size to be 32 for both modules. The framework is first trained by SGD with a learning weight 0.001 for the traffic context aware skip-gram module, then the learned representations are further tuned in the trajectory-enhanced Transformer by Adam with a learning rate of 0.001.

5.3 Overall Performance

We first compare Toast with six competitors on two datasets in terms of road segment based applications. The results are presented in Table 2 and we make several observations. First, GNN models and random walk based methods show different performance superiority on two tasks due to their capability of handling two issues as argued in Section 1. Compared to the GNN models, the random walk based methods perform much better on the task of road label classification, which validates our claim that feature uniformity on a road network would greatly dampen the effectiveness of GNN models. Meanwhile, they perform worse on traffic inference task. Since random walk based methods follow the assumptions that connected road segments or road segments with similar topology structure would share similar representations, they may not hold in terms of traffic patterns in road network settings. Second, Toast consistently achieves the best performance on both datasets across all evaluation metrics. For example, it outperforms the second best method by 9.8% in Macro-F1 and 8.5% in RMSE on average. It shows that Toast can better capture the traffic patterns and traveling semantics with the help of auxiliary traffic context prediction objective and trajectory-enhanced Transformer module. Third, three road network representation learning methods do not show their superiority, because they could only partially handle the issues faced by road network representation learning problem. In

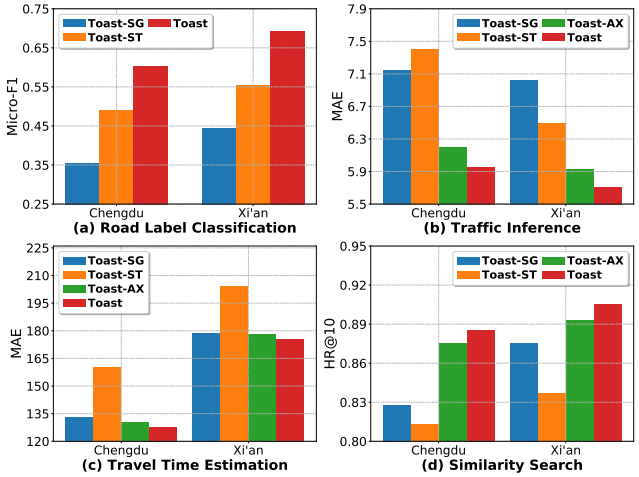

Figure 3: Ablation Study of Toast

contrast, Toast leverages both traffic context and trajectory data to encode multi-faceted knowledge about road networks, and address the two issues discussed in Section 1.

Next, we discuss the performance on trajectory similarity search and travel time estimation in Table 3 and Table 4 respectively. It can be observed that Toast consistently outperforms all competitors on both tasks. Additionally, for the task of travel time estimation, Road-Pool, deriving route representation by pooling from road segment representations outperforms the rest baseline methods, which demonstrates the effectiveness of the learned road segment representations of Toast from another perspective. For the task of trajectory similarity search, three trajectory similarity measures result in very different performance: EDR collapses while Fréchet is the second best among all the methods.

5.4 Model Analysis

5.4.1 Ablation Study for our Toast. We conduct an ablation study by independently removing different key components of Toast to understand their impacts on model performance. Specifically, we evaluate the following model variants: 1) Toast-SG: the model that removes the the traffic context aware skip-gram module and trained with only trajectory-enhanced Transformer module; 2) Toast-ST: the model which replaces self-attention network by RNN sequence-to-sequence learning as in [21, 40]; and 3) Toast-AX: the model that does not use auxiliary traffic context prediction objective and apply only the original skip-gram objective.

Figure 3 shows the results of all the model variants on four downstream tasks, and we make some observations. First, the performance drops significantly for Toast-SG with only the trajectory-enhanced Transformer on road segment based applications. Second, for trajectory based applications, self-attention network architecture shows its superiority compared to RNN (Toast-ST), which validates our choice of adopting it to capture the information encoded in trajectories. Third, the removal of auxiliary tasks (Toast-AX) also leads to a degradation of model performance, which indicates the contribution of the auxiliary traffic context prediction objective. In summary, these findings verify the effectiveness of different model components, and justify the rationale of our model design.

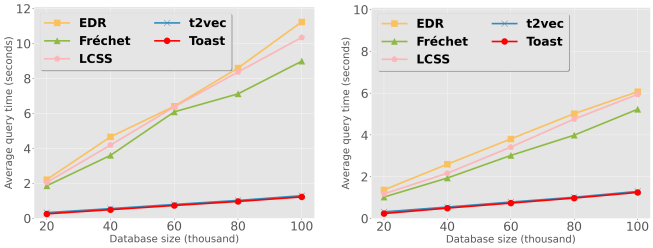

Figure 4: Trajectory similarity search efficiency

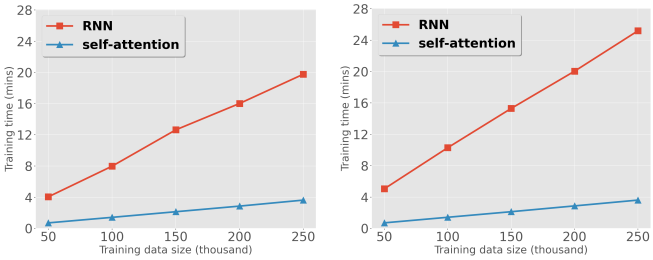

Figure 5: Training scalability

5.4.2 Model Efficiency. We choose the task of trajectory similarity search to evaluate the model efficiency. We vary the trajectory database size and record the average query time.

From the results presented in Figure 4, we find that route representation learning methods, namely t2vec and Toast, can be an order of magnitude faster than calculating trajectory similarity measures. In these methods, trajectories are encoded as vectors and we could simply use cosine similarity for query processing, which takes much less time compared to intricate pruning techniques applied in trajectory similarity measures. Moreover, the representation generation process can be done offline and is also efficient. It costs less than 4 minutes to encode all trajectories into vectors in our experiment. Therefore, our Toast scales well on large datasets, and is capable to support interactive analysis for computationally expensive operations such as trajectory clustering.

5.4.3 Training Scalability. We also investigate the training scalability of Toast. Here we focus on trajectory-enhanced transformer part, since skip-gram based model has been applied in many previous methods and already proved its scalability on large graphs [9, 26]. We compare the self-attention network with RNN on trajectory modeling efficiency, and all the trainings are conducted with a batch size of 32 on a single NVIDIA Tesla 100 SXM2 GPU. Figure 5 shows the results.

It can be observed that self-attention network has lower time cost than RNN. This is because self-attention attends all positions in parallel, so it requires $O(1)$ sequential operations; in contrast, an RNN requires $O(n)$ sequential operations to process the whole sequence. Also, the training time grows linearly w.r.t the training size, demonstrating that the model could handle large-scale datasets.

6 CONCLUSION

In this paper, we proposed a novel framework called Toast to learn both effective road segment representations and route representations, and leverage them to benefit different types of downstream applications. Two new modules were proposed in our framework: 1) traffic context aware skip-gram module to incorporate traffic contexts into the learning process, and 2) trajectory-enhanced Transformer module to capture the travelling semantics encoded in trajectory data. Our experiments demonstrated that Toast outperforms the state-of-the-art methods consistently on four different tasks.

ACKNOWLEDGMENTS

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from Singapore Telecommunications Limited (Singtel), through Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU). Cheng Long is supported by the Nanyang Technological University Start-Up Grant from the College of Engineering under Grant M4082302 and by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG20/19 (S)). Zhifeng Bao is supported by ARC DP200102611.

REFERENCES

[1] Helmut Alt and Michael Godau. 1995. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.* 5 (1995), 75–91.
[2] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR* abs/1607.06450 (2016).
[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.
[4] Lei Chen, M. Tamer Özsu, and Vincent Oria. 2005. Robust and Fast Similarity Search for Moving Object Trajectories. In *SIGMOD*. 491–502.
[5] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. 2016. Latent Space Model for Road Networks to Predict Time-Varying Traffic. In *KDD*. 1525–1534.
[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.
[7] Tao-Yang Fu and Wang-Chien Lee. 2020. Trembr: Exploring Road Networks for Trajectory Representation Learning. *ACM Trans. Intell. Syst. Technol.* 11, 1 (2020), 10:1–10:25.
[8] Palash Goyal and Emilio Ferrara. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowl. Based Syst.* 151 (2018), 78–94.
[9] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *KDD*. 855–864.
[10] Chenjuan Guo, Bin Yang, Jilin Hu, and Christian S. Jensen. 2018. Learning to Route with Sparse Trajectory Sets. In *ICDE*. 1073–1084.
[11] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In *AAAI*. 922–929.
[12] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*. 1024–1034.
[13] Jilin Hu, Chenjuan Guo, Bin Yang, and Christian S. Jensen. 2019. Stochastic Weight Completion for Road Networks Using Graph Convolutional Networks. In *ICDE*. 1274–1285.
[14] Tobias Skovgaard Jepsen, Christian S. Jensen, and Thomas Dyhre Nielsen. 2019. Graph Convolutional Networks for Road Networks. In *SIGSPATIAL*. 460–463.
[15] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *CoRR* abs/1611.07308 (2016).
[16] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
[17] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*, Vol. 32. 1188–1196.
[18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. In *AAAI*. 3538–3545.
[19] Xiucheng Li, Gao Cong, and Yun Cheng. 2020. Spatial Transition Learning on Road Networks with Deep Probabilistic Models. In *ICDE*. 349–360.
[20] Xiucheng Li, Gao Cong, Aixin Sun, and Yun Cheng. 2019. Learning Travel Time Distributions with Deep Generative Model. In *WWW*. 1017–1027.
[21] Xiucheng Li, Kaiqi Zhao, Gao Cong, Christian S. Jensen, and Wei Wei. 2018. Deep Representation Learning for Trajectory Similarity Computation. In *ICDE*. 617–628.
[22] Yiding Liu, Kaiqi Zhao, Gao Cong, and Zhifeng Bao. 2020. Online Anomalous Trajectory Detection with Deep Generative Sequence Modeling. In *ICDE*. 949–960.
[23] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*. 13–23.
[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*. 3111–3119.
[25] Dian Ouyang, Dong Wen, Lu Qin, Lijun Chang, Ying Zhang, and Xuemin Lin. 2020. Progressive Top-K Nearest Neighbors Search in Large Road Networks. In *SIGMOD*. 1781–1795.
[26] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *KDD*. 701–710.
[27] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. 2017. *struc2vec*: Learning Node Representations from Structural Identity. In *KDD*. 385–394.
[28] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. VideoBERT: A Joint Model for Video and Language Representation Learning. In *ICCV*. 7463–7472.
[29] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW*. 1067–1077.
[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.
[31] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.
[32] Michail Vlachos, Dimitrios Gunopulos, and George Kollios. 2002. Discovering Similar Multidimensional Trajectories. In *ICDE*. 673–684.
[33] Meng-xiang Wang, Wang-Chien Lee, Tao-Yang Fu, and Ge Yu. 2019. Learning Embeddings of Intersections on Road Networks. In *SIGSPATIAL*. 309–318.
[34] Meng-xiang Wang, Wang-Chien Lee, Tao-Yang Fu, and Ge Yu. 2021. On Representation Learning for Road Networks. *ACM Trans. Intell. Syst. Technol.* 12, 1 (2021), 11:1–11:27.
[35] Sheng Wang, Zhifeng Bao, J. Shane Culpepper, and Gao Cong. 2021. A Survey on Trajectory Data Management, Analytics, and Learning. *ACM Comput. Surv.* 54, 2 (2021), 39:1–39:36.
[36] Sheng Wang, Zhifeng Bao, J. Shane Culpepper, Timos Sellis, and Xiaolin Qin. 2019. Fast Large-Scale Trajectory Clustering. *Proc. VLDB Endow.* 13, 1 (2019), 29–42.
[37] Sheng Wang, Zhifeng Bao, J. Shane Culpepper, Zizhe Xie, Qizhi Liu, and Xiaolin Qin. 2018. Torch: A Search Engine for Trajectory Data. In *SIGIR*. 535–544.
[38] Sibo Wang, Xiaokui Xiao, Yin Yang, and Wenqing Lin. 2016. Effective Indexing for Approximate Constrained Shortest Path Queries on Large Road Networks. *Proc. VLDB Endow.* 10, 2 (2016), 61–72.
[39] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community Preserving Network Embedding. In *AAAI*. 203–209.
[40] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang. 2017. Modeling Trajectories with Recurrent Neural Networks. In *IJCAI*, Carles Sierra (Ed.). 3083–3090.
[41] Ning Wu, Wayne Xin Zhao, Jingyuan Wang, and Dayan Pan. 2020. Learning Effective Road Network Representation with Hierarchical Graph Neural Networks. In *KDD*. 6–14.
[42] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2020. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).
[43] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*. 2048–2057.
[44] Can Yang and Gyozo Gidofalvi. 2018. Fast map matching, an algorithm integrating hidden Markov model with precomputation. *International Journal of Geographical Information Science* 32, 3 (2018), 547–570.
[45] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. 2015. Network Representation Learning with Rich Text Information. In *IJCAI*. 2111–2117.
[46] Haitao Yuan, Guoliang Li, Zhifeng Bao, and Ling Feng. 2020. Effective Travel Time Estimation: When Historical Trajectories over Road Networks Matter. In *SIGMOD*. 2135–2149.
[47] Jing Yuan, Yu Zheng, and Xing Xie. 2012. Discovering regions of different functions in a city using human mobility and POIs. In *KDD*. 186–194.