

Graph Neural Networks for Intelligent Transportation Systems A Survey

Rahmani, Saeed; Baghbani, Asiye; Bouguila, Nizar; Patterson, Zachary

DOI

[10.1109/TITS.2023.3257759](https://doi.org/10.1109/TITS.2023.3257759)

Publication date

2023

Document Version

Final published version

Published in

IEEE Transactions on Intelligent Transportation Systems

Citation (APA)

Rahmani, S., Baghbani, A., Bouguila, N., & Patterson, Z. (2023). Graph Neural Networks for Intelligent Transportation Systems: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(8), 8846-8885. <https://doi.org/10.1109/TITS.2023.3257759>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Graph Neural Networks for Intelligent Transportation Systems: A Survey

Saeed Rahmani¹, Graduate Student Member, IEEE, Asiye Baghbani², Nizar Bouguila², Senior Member, IEEE, and Zachary Patterson³, Member, IEEE

Abstract—Graph neural networks (GNNs) have been extensively used in a wide variety of domains in recent years. Owing to their power in analyzing graph-structured data, they have become broadly popular in intelligent transportation systems (ITS) applications as well. Despite their widespread applications in different transportation domains, there is no comprehensive review of recent advancements and future research directions that covers all transportation areas. Accordingly, in this survey, for the first time, we provide an overview of GNN studies in the general domain of ITS. Unlike previous surveys, which have been limited to traffic forecasting problems, we explore how GNN frameworks have evolved for different ITS applications, including traffic forecasting, demand prediction, autonomous vehicles, intersection management, parking management, urban planning, and transportation safety. Also, we micro-categorize the studies based on their transportation application to identify domain-specific research directions, opportunities, and challenges, which have been missing in previous surveys. Moreover, we identify unique and undiscussed research opportunities and directions, which is the result of reviewing a wide range of transportation applications. The neglected role of edge and graph learning in ITS applications, developing multi-modal models, and exploiting the power of unsupervised and reinforcement learning methods for developing more powerful GNNs are some examples of such new discussions in this survey. Finally, we have identified popular baseline models and datasets in each transportation domain, which facilitate the development and evaluation of future GNN-based frameworks.

Index Terms—Graph neural networks, deep learning, intelligent transportation systems, GNN, ITS.

I. INTRODUCTION

THE unprecedented growth of cities has imposed an increasing burden on transportation systems. The emergence of new technologies and intelligent transportation systems, such as autonomous vehicles (AVs), shared mobility systems, and new paradigms of public transportation services, have made transportation systems even more complicated to analyze and manage. Fortunately, advancements in new

technologies have provided researchers with unique opportunities, which can be leveraged for tackling or alleviating the consequences of such complexities. The main endeavor, however, has been developing suitable and scalable algorithms and methods that can harness the full potential of such data for analyzing highly complex transportation systems.

Machine learning, and its more recent variant, deep learning algorithms, have turned out to be popular and practical solutions for many real-world applications and have proved to perform satisfactorily in the domain of intelligent transportation systems [1], [2], [3]. While deep learning algorithms have shown promising results, there are some shortcomings with these popular methods. Many deep learning methods have been developed assuming that real-world data can be represented in one, two, or three-dimensional Euclidean space. However, there are plenty of applications in which data is more realistically represented in the form of graphs, where the inter-relations between objects/features are in non-Euclidean space. Similarly, many transportation-related data inherently enjoy spatial features in a non-Euclidean space. For instance, in an urban road transportation network, links and intersections are influenced by specific links and intersections on the network, not necessarily the closer ones. This makes conventional deep learning methods, such as convolutional neural networks, unable to fully capture such spatial dependencies [4]. Recently, studies applying machine learning tasks on graphs have drawn special attention, and a new family of neural networks, namely graph neural networks has been introduced and embraced in various domains, including social science, chemistry, knowledge graphs, and e-commerce. This family of models has rapidly grown in many fields due to its superior performance, and starting from traffic forecasting problems [5], [6], has been applied to many transportation applications as well. These applications cover a wide range, including but not limited to traffic forecasting, travel demand prediction, autonomous vehicles, and intersection management. Considering the extensive use of GNNs in various domains of intelligent transportation systems and their fast-growing development, there is a need for a comprehensive review of current studies, their limitations and shortcomings, challenges, and future directions.

Accordingly, in this survey, we review the studies utilizing GNNs in the general domain of intelligent transportation systems. To the best of our knowledge, this is the first

Manuscript received 1 April 2022; revised 6 November 2022 and 31 January 2023; accepted 24 February 2023. This work was supported by the Mitacs under Grant IT18048. The Associate Editor for this article was W. Wei. (Corresponding author: Saeed Rahmani.)

Saeed Rahmani is with the Department of Transport and Planning, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: s.rahmani@tudelft.nl).

Asiye Baghbani, Nizar Bouguila, and Zachary Patterson are with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC H3G 1M8, Canada (e-mail: asiye.baghbani@mail.concordia.ca; nizar.bouguila@concordia.ca; zachary.patterson@concordia.ca).

Digital Object Identifier 10.1109/TITS.2023.3257759

survey on GNNs that is not limited to traffic forecasting problems and extensively investigate a wide range of applications in transportation engineering. Also, for the first time, we categorize the studies based on their transportation domains, which are traffic forecasting, demand prediction, autonomous vehicles, intersection management, parking management, urban planning, and transportation safety, and then investigate the studies per category. We believe such classification of studies is indeed essential for several reasons. Firstly, the problem design, graph construction, dataset treatment, and computational complexities highly depend on the transportation domain and application, and putting all studies under one umbrella results in too simplified analyses and overlooked properties of complex transportation systems. For instance, constructing graphs in traffic forecasting problems is more straightforward than in the AV domain, where there is an unknown number of dynamic and interactive objects. As another example, the influencing factors on a metro transit system are different from those influencing vehicular traffic.

Moreover, the nature of the dependent and independent variables are different among domains, which could greatly affect the design of GNN-based frameworks. In traffic forecasting and travel demand modeling, for example, GNNs are usually used for predicting a feature or variable over nodes of the graph, while in AV or intersection management domains, GNNs are usually used for learning control policies or unraveling agents interactions, and therefore, learning or predicting over the edges of the graph or over the whole graph is also of great or even more interest. These differences could greatly impact the design of GNN-based deep learning models and have been mainly neglected in previous related surveys. The second reason for such categorization of studies is that it helps transportation planners and traffic engineers to easily follow the state-of-the-art modeling endeavors and identify the limitations and challenges of current studies per transportation application. This allows them to explore how GNNs have evolved in each domain, and what modules, for which purposes, have been added to generic GNN-based frameworks, which can inspire them to design problem-specific frameworks based on the particular needs of each domain.

We should note that there are two related surveys in the transportation domain, but they do not overlap significantly with this study. First, they are limited to traffic forecasting problems and do not investigate problem-specific needs and challenges in other transportation domains, such as AVs, intersection management, and demand prediction. Moreover, their approach within the traffic forecasting domain is also different from the current survey. The first survey [7] mainly focuses on providing guidelines for building graph-based frameworks based on traffic datasets and available deep-learning tools. In other words, they try to map different *traffic problems* to different *deep learning modules* that are applicable to GNNs. However, we aim at identifying domain-specific open research areas and challenges by identifying state-of-the-art endeavors in each domain, which is not achievable in [7]. The second survey [8] provides a comprehensive *list* of studies

that have utilized GNNs for traffic forecasting problems and categorizes them based on their graph type, their adjacency matrix, and their dependent variable (i.e., traffic speed, traffic flow, and passenger flow). However, this survey does not review the individual studies and does not discuss the variables and features included/excluded in the literature. Therefore, identifying the problem-specific limitations and challenges is not feasible in this survey either.

Following the above discussions, the main contributions of this study could be summarized as follows:

- For the first time, the GNN studies in the general domain of intelligent transportation systems are reviewed and investigated in detail. Many of the studies in this survey, including AVs; demand modeling; intersection, urban, and parking management; and transportation safety, have been completely neglected or overlooked in previous related surveys.
- We categorize the studies based on the transportation application to identify domain-specific research needs and challenges and help researchers explore the state-of-the-art endeavors in their area of expertise. This practical classification of studies and the in-depth investigation of studies within each group is also lacking in previous surveys.
- We identify unique and undiscussed research opportunities and directions, which is the result of reviewing a wide range of transportation applications. Highlighting the importance of edge and graph learning, developing multi-modal demand prediction models, and exploiting the power of unsupervised and reinforcement learning for developing more powerful and efficient GNN-based frameworks are such examples.
- We identify popular baseline models in each transportation domain and provide baseline datasets and open-source codes, which facilitate the development and evaluation of GNN-based frameworks in different domains of intelligent transportation systems.

The rest of this survey is organized as follows. In the next section, we quickly overview the search methodology and investigated databases for identifying the reviewed papers. Then, we outline the background of GNNs and briefly introduce the concepts behind them. At the end of the section, we will also present a taxonomy of GNNs, which is helpful in understanding the different approaches used in the literature. In Section IV, previous surveys on graph neural networks, as well as those focusing on the transportation area, are reviewed. In Section V, we categorize the studies employing GNNs based on their transportation context, in order to explore how GNNs have evolved in different transportation domains and identify open areas worth further investigation. Finally, in Section VI, we look into open research areas and discuss current challenges facing the development and application of graph neural networks in the transportation domain.

II. METHODOLOGY

This section provides an overview of the methodology used for detecting and selecting the papers in this survey. The general procedure is depicted in Figure 1. Due to the

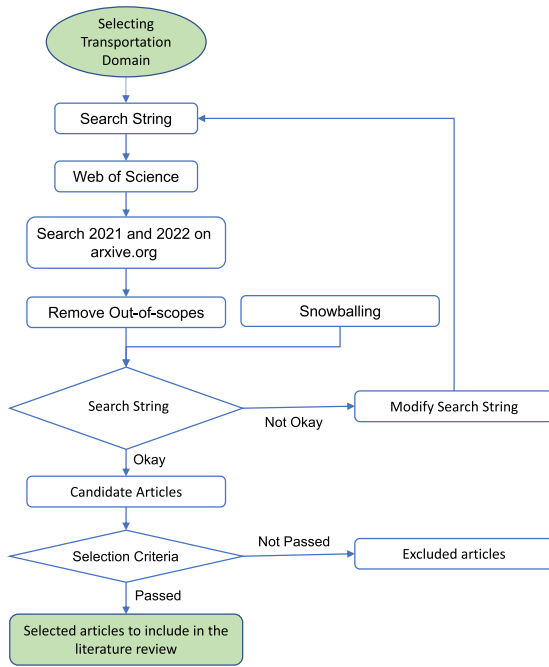


Fig. 1. Search methodology for detecting and selecting papers in this survey.

broad range of studies, we searched separately within each transportation domain by defining a unique search string. These strings are summarized in Table I. The asterisk sign (*) after a word expands the search to include different variations of the key search string. For instance, the term “transport*” includes “transportation” as well.

In the next step, the search string was used to find the relevant papers for the survey. The main database used is Web of Science (WoS), but the keywords were also searched in Google Scholar to identify the most recent papers that are not yet indexed in WoS or are only published in ArXiv.org. This was done because of the rapidly growing body of research in this domain. Afterward, we removed out-of-scope studies by reading their abstracts. In order to check if the selected search strings are appropriate, we also did a snowballing search and if the number of newly found papers during the snowballing was more than 30 percent of previously found papers, the search string was modified and the aforementioned procedure was repeated. After this step, the candidate articles were selected. However, due to the huge number of candidate papers in some categories (mostly traffic forecasting), we defined some criteria to select the most promising papers with true novelties and more added value. Criteria considered were the number of total citations, the number of citations per year, the impact factors of the journal in which the paper was published, and the importance of the conference in which the paper was presented. After this final step, those papers that passed the selection criteria were chosen as the final ones for inclusion in this survey. In total, 109 papers were selected to review.

III. GRAPH NEURAL NETWORKS - BACKGROUND

This section provides a short introduction to GNNs. First, we present basic notions in graphs, graph types, different tasks

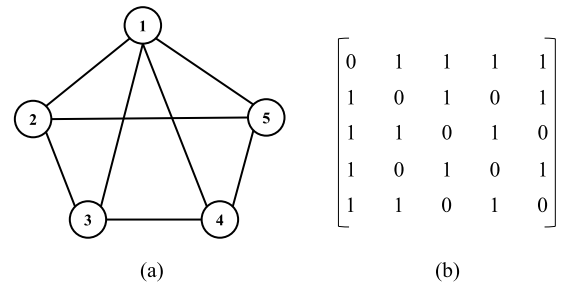


Fig. 2. (a) A simple graph of five nodes and eight edges, and (b) its adjacency matrix that denotes which nodes are connected to each other.

in graph representation learning, and computational modules in GNNs. Next, based on the introduced concepts, a taxonomy of GNNs is presented that is the basis for identifying and categorizing the structure of GNN models studied in this survey and developing future frameworks.

A. A Gentle Introduction to Graphs

As a general definition, a graph is simply a collection of objects (i.e., vertices or nodes), along with a set of interactions between the pairs of these objects (i.e., edges or links) [9]. A graph is typically defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (vertices) in the graph $\mathcal{G}$ and $\mathcal{E}$ is a set of edges (links) in this graph. Each edge can be denoted as $(u, v) \in \mathcal{E}$, where node $u \in \mathcal{V}$ and node $v \in \mathcal{V}$.

A graph is usually represented via its adjacency matrix. To form the adjacency matrix $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, nodes are first ordered in the graph so that each specific column/row represents a specific node in the graph. Then, the presence of edges in the graph is defined in the adjacency matrix as $A[u, v] = 1$ if $(u, v) \in \mathcal{E}$ and $A[u, v] = 0$ if $(u, v) \notin \mathcal{E}$. Figure 2 shows a simple graph with its adjacency matrix next to it. It is worth noting that in some types of graphs, the entries in the adjacency matrix are not limited to 0 or 1 and can accept arbitrary real values. The edges in these types of graphs are called weighted edges.

Another common term in spectral graph theory is graph Laplacian. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the Laplacian matrix is defined as $L := D - A$, where D is the degree matrix $D \in \mathbb{D}^{|\mathcal{V}| \times |\mathcal{V}|}$, and A is the adjacency matrix defined earlier. The Laplacian matrix is the discrete version of the Laplacian (the divergence of the gradient) operator over graphs.

After introducing how a graph is constructed and defined, in the following sections, we briefly introduce different types of graphs including directed/undirected graphs, heterogeneous/homogeneous graphs, dynamic/statistic graphs, multi-relational graphs, hypergraphs, and signed graphs.

B. Types of Graphs

Graphs that make up the basis of GNNs can be categorized as Directed/Undirected, Static/Dynamic, Homogeneous/Heterogeneous, Hypergraphs, and Signed graphs [8]. However, there are also some other specific types of graphs, and in the following subsections, we introduce them as well.

TABLE I
THE SEARCH STRINGS USED FOR FINDING THE LITERATURE PER EACH TRANSPORTATION DOMAIN

Transportation Domain	Search String			No. Found	No. Selected		
Traffic Forecasting	"GNN*" OR	AND	"traffic"	AND	"forecast*" OR "predict*" OR "estimate*" OR "ride-hailing" OR "ride-sharing" OR "transportation network compan*" OR "TNC" OR "shared mobility"	588	24
Demand Modeling	"Graph neural network*" OR "graph convolution"		"demand" OR ("passenger" AND ("flow" OR "count" OR "inflow" OR "outflow"))		"shared bike" OR "bike sharing" OR "shared bicycle" OR "bicycle sharing" OR "shared scooter" OR "e-bike sharing"	22	7
					"bus" OR "transit" OR "metro"	13	5
					"multi-mode" OR "multi-modal" OR "multi-demand"	37	10
					"percept*" OR "point cloud" OR "semantic segment*" OR "object detect"	7	3
Autonomous Vehicles			("automated" OR "autonomous" OR "driverless" OR "self-driving") AND ("vehicle" OR "car")		("motion" OR "move*" OR "intention") AND ("Predict*")	8	8
					("motion" OR "path") AND ("plan*")	15	18
Intersection Management			"intersection" OR "traffic signal"		"management" OR "control"	8	8
Parking Management			"parking"		"management" OR "control"	27	10
Urban Planning			"urban" OR "land use" OR "city"		"planning"	1	5
Transportation Safety			"traffic safety" OR "accident*" OR "crash"			81	6
						64	5

1) *Directed/Undirected Graphs*: Graphs can be broadly classified into directed and undirected graphs. In an undirected graph, edges do not have a direction, which means there is a two-way relationship, and information is passed in both directions. In other words, in an undirected graph, the adjacency matrix is symmetric along the main diagonal $((u, v) \in \mathcal{E} \leftrightarrow (v, u) \in \mathcal{E})$. On the other hand, directed graphs do not impose the restrictive assumption of symmetry on the edges and therefore can provide more flexible information propagation through the edges.

2) *Weighted Graphs*: In weighted graphs, there are arbitrary values associated with edges. The weights of the edges (links) can provide valuable information about the graph and its edges. For instance, the weights can represent the length of the edge or the cost/time needed to traverse the edge. In an unweighted graph, however, the adjacency matrix is defined only based on the binary condition of whether there is a link/relationship between two vectors; therefore, the values in the adjacency matrix are zeros or ones.

3) *Multi-Dimensional Graphs*: In addition to the concepts of directed/undirected and weighted edges, some graphs have edges of different types. In other words, the types of relationships between vertices might be different across the graph. Accordingly, a new term is added to the notation of edges, indicating the type of the edge $((u, \tau, v) \in \mathcal{E})$, where τ indicates the type of the edge from u to v . Subsequently, a specific adjacency matrix is defined per edge type, and the graph is represented by the adjacency tensor

$A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{R}| \times |\mathcal{V}|}$, where $\mathcal{R}$ is a set of relationships in the graph. Accordingly, in multi-relational or multi-dimensional graphs, a pair of nodes can share multiple types of edges. This is useful especially when there are different types of relationships between nodes. For example, in a transportation network, two specific regions might be correlated not only based on their direct distance from one another but also based on socio-economic characteristics. These types of relationships could be captured by multi-relational (-dimensional) graphs. An example of a multi-dimensional graph can be found in Figure 3.

4) *Heterogeneous Graphs*: In a regular graph, all nodes share the same set of features. For example, we may consider the stations of a public transportation network as nodes of the public transportation graph. For each station, we have a number of features such as the number of boarding passengers, the lighting condition, etc. In a heterogeneous graph, there are disjoint subsets of dissimilar nodes, and each node in the graph belongs to one subset:

$\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{V}_3 \cup \dots \cup \mathcal{V}_k$, where $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \forall i \neq j$. In the former example of the public transportation graph, we might distinguish between bus and metro stations and include some more features in the nodes for bus stations (features like whether the station has a shelter or if the station is before or after the intersection). Heterogeneous graphs provide much more flexibility when dealing with nodes that are inherently different but that interact with each other. This is quite common in multi-modal transportation networks, especially in urban areas.

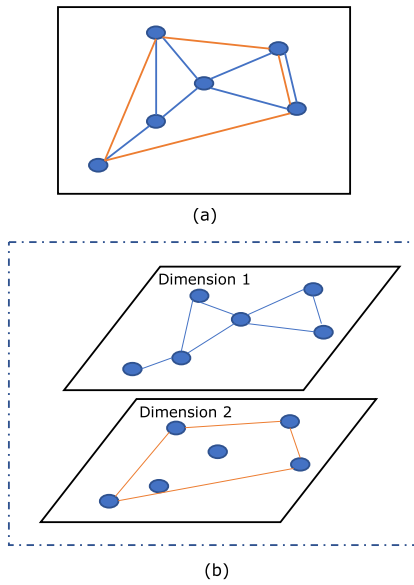


Fig. 3. An example of a multi-dimensional graph with two dimensions. As can be seen, the connected nodes and the adjacency matrices are different for the two dimensions. a) The multi-dimensional graph, a) the two dimensions of the graph depicted separately.

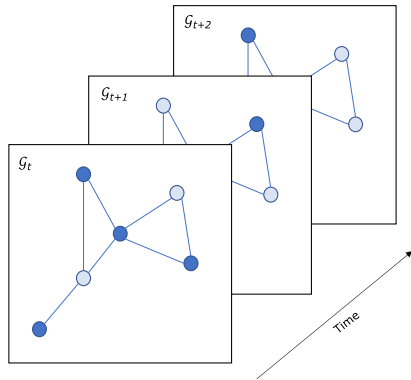


Fig. 4. An example of a dynamic graph. The active nodes and the connections change over time in this class of graphs.

5) *Multiplex Graphs*: Multiplex graphs consist of several layers, where each node belongs to every layer and the relationships between nodes are layer-specific. There are two types of edges in multiplex graphs: 1) intra-layer edges that form the relationships between nodes in each layer, and 2) inter-layer edges that connect the same nodes across different layers of the multiplex graph. Multiplex graphs can be useful in representing transportation networks, where each layer represents a different mode of transportation.

6) *Dynamic Graphs*: When input features or the topology of the graph vary with time, the graph is regarded as a dynamic graph [8]. Time information needs to be carefully considered in dynamic graphs. A dynamic graph could be defined as $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T)$, where $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ and $t \in T$, where T is the number of time intervals. In dynamic graphs, new nodes can emerge or disappear, and new relationships between nodes can be established or terminated in different time periods. A simple dynamic graph is shown in Figure 4.

7) *Hypergraphs*: In contrast to a regular graph in which an edge can only connect two vertices, in a hypergraph, an edge

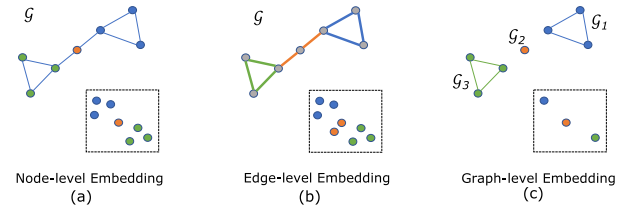


Fig. 5. Visual representation of the three different levels of granularity in graphs: a) node-level representation, b) edge-level representation, and c) graph-level representation. At each level, specific parts of the graph are embedded and learned.

can connect any number of vertices. Therefore, hypergraphs can establish relationships beyond pairwise nodes, which could be beneficial when there are relationships between multiple nodes. Gao et al. [10] provided insights into the methods and enumerated some applications of hypergraphs in their study.

8) *Signed Graphs*: Signed graphs are graphs with signed edges, i.e., an edge can be either positive or negative. This type of relationship could ease some representations when dealing with binary relationships between vertices. Also, it is sometimes desirable that edges have negative weights indicating the dissimilarity or distance between nodes. This type of representation adds still more flexibility when representing data in a graph structure.

9) *Nested Graphs*: Nested graphs, which are useful in hierarchically representing relationships, are another class of graphs in which nodes themselves are (sub)graphs. From a transportation point of view, nested graphs could be useful when dealing with multi-level information flows or prediction tasks. For example, one might be interested in the large-scale interaction between cities and at the same time, finer levels of interaction between subsets of roads within those cities. In this case, we might have two GNNs, one trained at a higher level for inter-city relationships, and the other for exploring relationships within cities.

The next section provides an overview of different machine-learning tasks related to graphs.

C. Machine Learning on Graphs

Graph-structured data can be analyzed or represented at different levels: node-level, edge-level, and graph level, as depicted in Figure 5. Each category can address a specific type of question and may need distinct algorithms for performing the task. Apart from the graph tasks, machine learning algorithms are also categorized as supervised, semi-supervised, and unsupervised. The combination of graph-related tasks and learning types provides tremendous flexibility when dealing with complex problems. Therefore, in this section, we first give a quick overview of different machine learning tasks on graphs [11] and then provide a taxonomy of GNNs, which will be beneficial when developing a graph-related deep learning structure, especially for new problems.

1) *Node-Level*: Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the goal of node-level learning is to learn the features of the targeted nodes in the graph $v \in \mathcal{V}$. This includes node classification, node regression, and node clustering. Node classification tries to categorize nodes into several classes, and node regression

predicts a continuous value or property for each node. Node clustering aims to partition the nodes into several disjoint groups based on their similarities, where similar nodes should be in the same group [8].

Node classification/regression is probably the most popular machine-learning task on graph data, especially in the transportation domain. In many common node classification problems, only a fraction of nodes are labeled for the training task and the goal is to predict the feature(s) for all nodes $v \in \mathcal{V}$. One major difference between graph-based node classification and traditional classification problems is that in graphs, nodes are not independent and identically distributed (i.i.d) [9], while in traditional classification problems, we either assume that each data point is statically independent of others or try to model the dependencies between data points.

2) *Edge-Level*: Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the goal of edge-level tasks is to apply the learning task on each edge $e \in \mathcal{E}$. Edge-level tasks generally include edge classification/regression and link prediction; the former requires the model to classify edge types or predict properties for edges of the graph, and the latter tries to predict whether there is an edge between two given nodes [8]. Link prediction (also known as relation prediction and graph completion) aims to predict the unknown or missing relations/links between nodes in the graph, given an incomplete set of links/relationships between a training set of nodes $\mathcal{E}_{train} \subset \mathcal{E}$.

3) *Graph-Level*: Given a dataset with multiple (n) different graphs, the goal is to develop machine learning algorithms on the entire graph. This class of tasks involves classification/regression, and clustering. In graph classification and regression tasks, the aim is to learn a feature over the entire graph and predict the label or property associated with each graph instead of predicting the individual components (vertices or edges) within the graphs. As an example, one might be interested in labeling the traffic situation in a whole city or region as highly congested or not. Also, the overall situation at an intersection could be measured at different levels of safety based on the interactions between the agents on the scene (vehicles, cyclists, and pedestrians).

As will be discussed in Section V, most of the current studies utilizing GNNs in the transportation domain have focused on node-level tasks, and the applications of link-level and graph-level learning tasks have been mainly neglected despite their great potential. We further discuss utilizing graph-level and edge-level learning in GNNs and their applications in the transportation domain in Section VI.

D. Taxonomies of GNNs

In this section, we present the taxonomies of GNNs introduced in previous relevant studies and then propose a taxonomy wrapping up the discussion in previous surveys on categorizing GNNs to ease the understanding of different frameworks and also to help researchers to combine different GNNs toward solving more complex problems.

Different taxonomies have been proposed in previous studies. Wu et al. [12] classified GNNs into recurrent graph neural networks (RecGNNs), convolutional graph neural

networks (ConvGNNs), graph autoencoders (GAEs), and spatial-temporal graph neural networks (STGNNs). The main difference between RecGNNs and ConvGNNs is that node representations in RecGNNs are learned using recurrent neural architectures, whereas in ConvGNNs it is done based on the aggregation of the features of the target node and its neighbors (stacking multiple graph convolution layers). GAEs first encode inputs (nodes/graphs features) into a latent space, and then the graph is reconstructed based on the encoded space. Finally, spatial-temporal graph neural networks aim to simultaneously capture the spatial and temporal dependencies in the graph.

In another proposed taxonomy of GNNs, Zhou et al. [13] introduced a taxonomy based on different computation modules in GNNs, namely propagation, sampling, and pooling. The propagation module, which is responsible for information dissemination through the graph, could be comprised of convolution, recurrent, and/or skip connection operators. The sampling module is usually integrated with the propagation module in large graphs to make the propagation of information efficient and thus feasible in such graphs. The sampling could be done on the node, layer, or subgraph levels. And finally, the pooling module is required when we aim to achieve representations for high-level sub-graphs and itself could be categorized as direct methods and hierarchical ones. With these definitions, the GNN layers in graph-based deep learning frameworks could be easily classified as is done in this study [13].

Abadal et al. [14] proposed a more comprehensive taxonomy by adding graph adversarial networks and graph reinforcement learning to the identified classification in [12]. This addition was inspired by the study of Zhang et al. [15] who initially considered Graph Adversarial Networks and Graph Reinforcement Learning as part of their taxonomy.

In order to better understand and compare the taxonomies presented in previous studies, and to provide a handy taxonomy of GNNs for transportation research, a collective graph of different taxonomies of GNNs is introduced in this survey, which is depicted in Figure 6. This figure has two levels; in the lower level, each study (denoted by filled rhombi) is connected to its proposed taxonomy using a colored arrow with the same color as the rhombus. This is useful for recognizing how previous studies have categorized and identified different GNNs. At the higher level, GNNs introduced in previous studies are grouped together under the umbrella of main GNN classes, which are recurrent GNNs, convolutional GNNs, spatial-temporal GNNs, graph reinforcement Learning, and graph autoencoders and adversarial GNNs. These higher-level classes are depicted using transparent big ellipses in Figure 6. Please note that the spatial-temporal GNNs are depicted as the intersection of convolutional GNNs and recurrent GNNs. Furthermore, we consider only one class for Graph Auto-encoders and Graph Adversarial Networks. This is because adversarial techniques have usually been used for training graph autoencoders to improve their generalizability and make them robust against adversarial attacks. In the following paragraphs,

we dive into each of these higher-level classes on GNN and describe them in more detail.

1) *Recurrent GNNs*: Recurrent modules in GNNs learn node representations using recurrent neural architectures. These types of models are the first generations of GNNs and may use different aggregation and updating functions, such as simple aggregation without edge transformation or gated units [14].

2) *Convolutional GNNs*: This class of GNNs, which is very popular in the transportation domain, aims to generalize the operation of convolution from grid data to graph-structured data [12]. The main idea is to generate the node representation by aggregating the features from the target node and its neighbors. Convolutional GNNs are broadly categorized as spectral-based methods that are developed based on the spectral graph theory and compute over the whole graph at once, and spatial-based methods that perform the convolution operator only on a subset of nodes that are considered as the neighbors of the target node. Spatial-based methods are more efficient as they do not need the whole graph to be analyzed at once and, therefore, are appropriate for large-scale problems and real-time applications.

3) *Spatial-Temporal GNNs*: As it is also illustrated in Figure 6, this class of GNNs, which are the most popular class in traffic forecasting studies [8], try to put together the benefits of both recurrent and convolutional component of graph neural networks in order to simultaneously capture the spatial and temporal interdependencies. This could bring a huge advantage in a wide range of transportation problems, since in most cases they deal with spatial data points that not only have interactions with each other but also show correlations with previous time slots. Spatial-temporal graph neural networks (ST-GNN) have been extensively used in traffic state and travel demand prediction tasks.

4) *Graph Autoencoders and Adversarial GNNs*: These generative models are unsupervised learning frameworks with an encoding-decoding structure. The encoder module encodes the graph into a latent space (latent representation) and the decoder tries to reconstruct a new graph similar to the initial one from this latent space. Integrating graph convolutions and recurrent units in the encoding and decoding steps could add great flexibility to graph autoencoders. Graph auto-encoders could also be used for multi-step prediction tasks in order to avoid error propagation in relatively long-term time horizons. Adversarial techniques could also be used for training the graph auto-encoders, making them robust against adversarial attacks. Sun et al. [16] reviewed strategies for adversarial learning on graph data.

5) *Graph Reinforcement Learning*: Combining reinforcement learning with graph neural networks could also be interesting, especially in graph learning tasks, where specific strategies or policies are desirable. Although integrating reinforcement learning into GNNs could be really helpful in solving many transportation problems, this area has been mainly overlooked and needs special attention. One interesting area that could be targeted by graph reinforcement learning methods is goal-oriented graph construction [17], in which the structure of the graph is not determined in advance and should

be learned based on specific target objectives. This could also be beneficial when improving or modifying the current structure of pre-defined graphs. In this specific example, an agent tries to find the most optimum graph structure by trial and error based on a pre-defined policy or objectives. The applications of reinforcement learning in GNN frameworks, as an arguably missed-out class of GNNs, are more discussed in Section VI.

After a brief introduction to different types of graphs and the taxonomy of graph neural networks, in the next section, we briefly review the current surveys on graph neural networks and then those focusing specifically on the transportation domain. Thereafter, we justify the need for a new perspective on current research utilizing GNNs and highlight the contributions of this survey.

IV. RELATED SURVEYS

In this section, we briefly overview the related surveys on GNNs. These surveys can be categorized as general surveys and transportation-related surveys. General surveys are covering GNNs in a broad range of applications, while transportation-related ones only focus on one or several domains related to transportation engineering, such as traffic forecasting.

A. General Surveys on GNNs

Zhang et al. [18] presented one of the first surveys on GNNs, but focused only on graph convolutional neural networks (GCNs). Their survey covered various aspects of GCNs, including their mathematical foundations, model architectures, and different variations and extensions of GCNs. The novelty of their study is two folded. Firstly, they introduced a taxonomy of GCNs based on the type of graph filtering operations, which can be broadly categorized as spectral-based and spatial-based methods. Secondly, they categorized GCNs based on their application domains, identified as computer vision, natural language processing, and science. They also discussed recent models in each category and identified several research directions based on the surveyed studies, including developing deeper GCNs, developing dynamic GCNs, and applying multiple-graph convolutional networks.

Wu et al. [12] argued that previous surveys on the general topic of GNNs included only some of the GNNs and reviewed a limited number of studies. Therefore, they provided a comprehensive review of GNNs, together with descriptions of representation models, comparisons between different GNN models, and summarizing the developed algorithms. Also, they introduced a new taxonomy of GNNs, namely, RecGNNs, ConvGNNs, GAEs, and STGNNs. Furthermore, they categorized the applications of GNNs (computer vision, natural language processing, traffic forecasting, and recommendation systems) and proposed suggestions for future research on GNNs by focusing on the depth of GNN models, scalability of GNNs, heterogeneity in GNNs, and dynamicity of GNNs, which were not highlighted comprehensively in previous surveys.

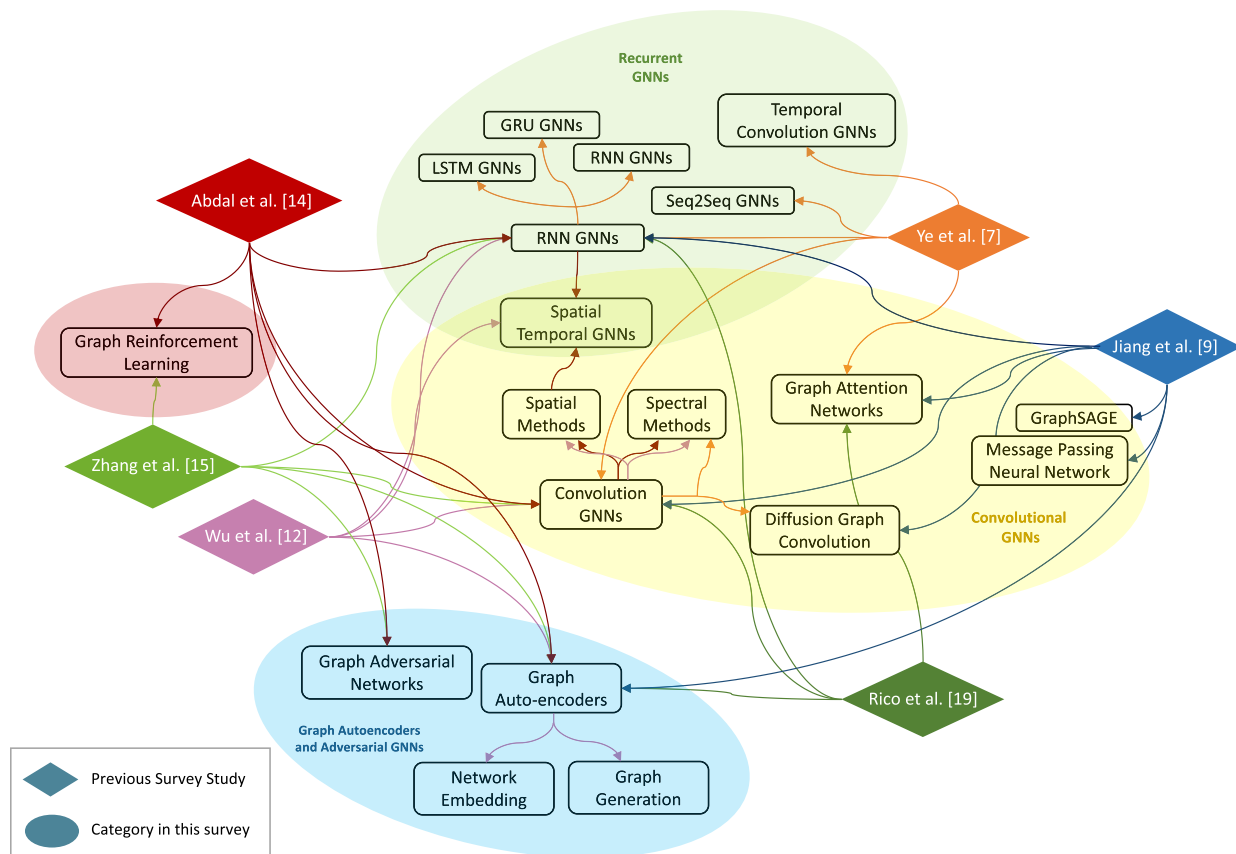


Fig. 6. Taxonomies of GNNs in the literature and the taxonomy used in this survey (denoted by transparent big ellipses).

Zhang et al. [15] reviewed different deep learning methods on graphs and identified two new classes of models not addressed in previous surveys: graph reinforcement learning (Graph RL) models, and graph adversarial methods. Accordingly, they proposed a new taxonomy of GNNs based on model structures and training strategies, which consists of RecGNNs, GCNs, GAEs, Graph RL models, and graph adversarial methods. The development history of each class of GNNs was also reviewed and the differences between them were discussed in detail. Finally, the future directions and open research areas were identified mainly considering different graph types, complex model structures, and the interpretability and robustness of GNN models.

Finally, Zhou et al. [13] provided a novel and different taxonomy of GNNs by focusing on classic GNN structures and building upon different computational modules used in the GNN frameworks. These computational modules, at the higher level, include propagation, sampling, and pooling. The propagation module is responsible for the transmission of information between nodes; sampling is usually combined with the propagation module in large graphs to facilitate the transition of information in large graphs; and pooling is used for producing coarsened representations of a graph or reducing the number of nodes by extracting and pooling the information from nodes. They also proposed a general and novel pipeline for designing GNN frameworks based on the graph types, loss function designs (node-, edge-, and graph-level learning tasks), and the investigated computational modules. Finally, they

systematically categorized the applications of GNNs based on the explicit or implicit structural relations of the data. Robustness and interoperability of GNN models, pre-training the graph, and using complex graph structures are enumerated as open research areas in this survey.

B. Transportation-Related Surveys

As was discussed briefly in the literature, almost all previous transportation-related surveys are focused on traffic forecasting problems. In this section, we briefly overview these studies to identify their main contributions. Ye et al. [7] were among the first researchers who conducted a survey of the studies on graph neural networks in the traffic domain. They associated different traffic problems with different research directions and tried to identify suitable deep learning algorithms that could be applied to specific traffic prediction problems. Moreover, they discussed how to build graphs and define their adjacency matrices from different traffic datasets. Interestingly, as one of their recommendations for future research directions, they encouraged developing GNNs for other transportation problems other than traffic state prediction, which also highlights the importance of the current survey. Also, they argued that most of the current studies in the traffic domain utilize spectral graph convolution networks or diffusion graph convolution networks and suggest utilizing more diverse deep learning techniques in GNN-based frameworks.

Rico et al. [19] took a slightly different approach to reviewing the GNN models for traffic forecasting. They first

classified GNN models into four main types, namely, recurrent GNNs, convolutional GNNs, graph attention networks (GAT), and graph autoencoders. Then, they discussed the literature per each GNN category, and summarized the state-of-the-art in traffic forecasting based on the scope of the studies (freeway, urban area), traffic variable predicted (speed, flow, or volume), and data types (loop detectors, or floating car data). Moreover, popular Python libraries and traffic datasets were introduced.

Jiang et al. [8] presented the most comprehensive survey on GNNs for traffic forecasting by reviewing more than 200 studies (including conferences and pre-prints) between the years 2018 and 2020. The literature in this survey is classified into four main groups, including traffic flow, traffic speed, traffic demand, and other studies. In the next step, different types of traffic graphs and adjacency matrices based on different traffic states (flow, speed, and demand) and sensors are recognized and discussed, which has not been discussed in such detail in previous surveys. Also, they introduced a more comprehensive taxonomy of GNNs, which consisted of recurrent GNNs, graph attention networks, graph convolutional networks, diffusion graph convolutions, graph autoencoders, GraphSAGE, and message-passing neural networks. They enumerate several unmentioned challenges in utilizing GNNs, including heterogeneous datasets, multi-task performance, practical implementation challenges, and model interpretation.

Bui et al. [20] focused only on spatial-temporal graph neural networks (ST-GNN) for traffic forecasting. The uniqueness of their study is two-folded. Firstly, they put forth a new taxonomy of ST-GNNs by dividing the existing models into four classes, namely graph convolutional recurrent neural network, fully graph convolutional network, graph multi-attention network, and spatial-temporal graph structure learning. Secondly, they conducted comparative experiments using selected benchmark datasets (METRLA [6] and UVDS [21]) to evaluate the performance of the representative models of each category.

After reviewing the existing surveys on GNNs in the transportation domain, it is apparent that almost all surveys have focused on the applications of GNNs for traffic forecasting. Although Ye et al. [7] and Jiang et al. [8] have mentioned some studies apart from traffic forecasting, their main focus and categorization have been again based on traffic forecasting problems, and the number of studies beyond traffic forecasting in these papers is insignificant and categorized as others. However, there has been a growing amount of research in recent years utilizing GNNs in other areas of transportation than traffic forecasting. Connected and autonomous vehicles, intersection management, safety studies, and shared mobility systems are among the newly trended areas that have been overlooked in previous studies and require more investigation. Moreover, most of the current studies on traffic forecasting have focused on node-level tasks, which is not surprising due to the type of data and goal in traffic forecasting problems. This has led previous studies and surveys to overlook other interesting learning tasks on graphs, such as link estimation and prediction and graph learning. Therefore, in this survey, we also aim to

open new discussions on the application of GNNs on edge-level and graph-level tasks, which we believe are the missing pieces of the puzzle in the current studies utilizing GNNs in the transportation domain. Finally, previous studies have mainly focused on the GNN structures and computational modules in graph neural networks. Although this perspective is important for understanding the mechanism of different GNN-based frameworks and identifying the technical gaps in developing new frameworks, reviewing the current studies based on their transportation contexts could also be equally important because different transportation problems require different sets of data with different characteristics, and computational cost and accuracy needs. Therefore, the GNN-based frameworks developed for traffic forecasting problems do not necessarily fit the needs of other problems. Moreover, categorizing the current studies on GNNs based on their targeted transportation domain helps us to identify the areas that need more exploration and formulate domain-specific needs and challenges.

In the next section, we conduct a comprehensive review on studying utilizing GNNs in different domains of transportation. These domains include traffic forecasting, demand modeling, autonomous vehicles, intersection management, parking management, urban planning, and transportation safety. Also, as some of the areas themselves cover a wide spectrum of studies, we sub-categorize the studies wherever needed.

V. GNNs IN TRANSPORTATION

In this section, we try to categorize the studies utilizing GNN-based deep learning frameworks for intelligent transportation systems based on their applications in the relevant transportation sector. We aim to recognize how GNNs have evolved in different domains of transportation and then identify the research gap and future research direction in each category. We start with the traffic forecasting problem as it has been arguably the most popular area among researchers.

A. Traffic Forecasting

Traffic forecasting has always been among the most interesting topics in intelligent transportation systems studies. It aims at predicting traffic characteristics (such as speed, flow, or density) in a short or long future time horizon in order to aid different ITS applications [22], such as advanced traffic management and control systems to travelers' information systems and shared, connected, and autonomous mobility systems operations. Accordingly, during the last three decades, a vast majority of studies in the area of intelligent transportation systems have focused on traffic forecasting and prediction [23], [24].

During the last decade, real-time measurements of traffic variables using emerging sensors have shifted traffic forecasting efforts towards data-driven methods, and numerous attempts have been made to develop flexible, large-scale, and real-time traffic forecasting models [22], [25], especially using deep learning methods [23], [26]. However, traditional deep learning models may neglect some interesting properties of transportation networks. For instance, most recent

studies on traffic forecasting using traditional deep learning architectures, have used convolutional operators to consider spatial dependencies among data points, while due to the special characteristics of transportation networks, the spatial correlations are not necessarily distributed in Euclidean space. In other words, two data points might be spatially close in Euclidean space but interact independently when considering traffic operations [4] in a transportation network based on the network connectivity/proximity. This is where GNNs change the game. Incorporating graph structures into deep learning frameworks allows GNNs to harness the power of artificial intelligence on graph data. Therefore, in recent years, there have been a growing number of studies incorporating GNNs for the purpose of traffic forecasting. In this section, we review the studies using GNNs in traffic forecasting problems and try to identify the shared and common modeling insights in these studies in order to identify current research directions, as well as open research areas.

One of the first attempts to utilize graph neural networks in traffic forecasting was the work of Shahsavari [5]. He proposed a graph-oriented model for considering spatial-temporal correlation amongst traffic sensor observations in a transportation network. In this framework, nodes correspond to the sensor locations (and corresponding features are extracted including traffic flow, density, and speed), and edges represent spatial interrelations forced by the network topology (such as length, capacity, and direction). Finally, a GNN model is trained in a supervised learning approach to predict short-term future traffic conditions. Later in 2017, Li et al. [6] incorporated the recurrent units into GNNs and developed a diffusion convolutional recurrent neural network model (DCRNN) for traffic forecasting problems. They modeled spatial dependencies in traffic networks as a diffusion process on a directed graph by proposing diffusion convolution and considered temporal dependencies using diffusion convolutional gated recurrent units by replacing the matrix multiplication in gated recurrent units with diffusion convolution. Also, for multi-step forecasting, a sequence-to-sequence learning framework was developed. Comparing the results with some benchmark models (historical average (HA), $ARIMA_{cal}$, Vector auto-regression, support vector regression, feed-forward neural network (FNN), and fully-connected long-short-term-memory (FC-LSTM)), they showed that the performance of their model is promising on the METR-LA and PeMS-BAY datasets.

STGCN (spatial-temporal graph convolutional network) was proposed [27] for traffic forecasting by incorporating the convolution operator in the GNN model. STGCN consists of two spatial-temporal convolutional blocks (ST-Conv), followed by a fully connected layer. Each ST-Conv block itself includes two temporal gated convolution layers (for considering temporal correlations) that surround one spatial graph convolutional layer (for considering spatial dependencies). They tested their model on two real-world datasets, and the results indicated the satisfactory performance of their framework compared to baselines (HA, linear support vector regression (LSVR), auto-regressive integrated moving

average (ARIMA), FNN, FC-LSTM [28], and DCRNN) in terms of training time, ease of convergence, number of parameters, flexibility, and scalability.

By utilizing a new gated attention network (GaAN), Zhang et al. [29] built the graph gated recurrent unit (GGRU) for traffic forecasting that enjoys multi-head attention-based aggregator with additional gates on the attention heads. By using multiple attention heads, they were able to explore features in different representation subspaces, which provided them with more modeling power. Their GGRU model outperformed the baselines (FC-LSTM, GCRNN, and DCRNN), but it showed to be incapable of considering link features.

Later, Zhao et al. [30] integrated gated recurrent units and graph convolutional networks and proposed T-GCN, a temporal graph convolutional network for traffic prediction, which is designed to capture the topological structure of traffic networks while considering spatial dependencies using a GCN. Also, similar to DCRNN, GRU is used for capturing temporal dependencies in traffic data.

Shin and Yoon [31] proposed a multi-weight traffic graph convolutional (MW-TGC) network model, which utilizes multi-weighted adjacency matrices for combining multiple features, including speed limit, distance, and the angle between two road segments. In this model, a spatially isolated dimension reduction operation is applied to the combined features to learn their dependencies and reduce the output size to a level that is computationally feasible. Moreover, the sequence-to-sequence model with LSTM units is used to learn temporal relationships from the multiweight graph convolution. Results of experiments on two study sites with varying geospatial configurations demonstrated that MW-TGC outperformed other state-of-the-art graph convolution models, including TGC-LSTM [4], STGCN [27], and Seq2Seq [28] on both sites.

To consider the dynamic temporal dependencies (i.e., short-term, daily, and weekly), Guo et al. [32] developed an attention-based spatial-temporal graph convolutional network model, which consisted of three independent components (for hourly, daily, and weekly time intervals). Each component itself is comprised of several spatial-temporal blocks consisting of spatial and temporal attention mechanisms for capturing dynamic spatial-temporal correlations, followed by graph convolutions for capturing spatial patterns and standard convolutions for describing temporal features.

Inspired by graph attention networks and encoder-decoders, Pan et al. [33] proposed ST-MetaNet, a spatial-temporal meta graph attention network for multi-step traffic forecasting. Their model is composed of an encoder-decoder each of which consists of four components: 1) RNN for embedding the sequence of historical data, 2) Meta-knowledge learner for learning meta-knowledge from nodes and edges attributes, 3) Meta-GAT for capturing diverse spatial correlations from meta-knowledge of all nodes and edges, and 4) Meta-RNN for capturing temporal correlations from meta-knowledge of all nodes. They tested their framework on two real datasets for taxi flow prediction and traffic speed prediction and concluded

their model could beat the baselines (HA, ARIMA, GBRT, Seq2Seq, GAT-Seq2Seq, and DCRNN).

Cui et al. [4] proposed the traffic graph convolutional long-short-term memory neural network (TGC-LSTM) model for network-wide prediction of traffic states. They added two regularization terms to the model's loss function (an L1-norm on graph convolution weights and an L2-norm on graph convolution features) to enhance the interpretability of their model. Moreover, by introducing the neighborhood matrix and the free flow reachability matrix, they defined the k-order traffic graph convolution (TGC) in order to consider both graph edge properties (e.g., the distance between sensing locations) and high-order neighborhood in the traffic graph. Their experiments showed that their proposed model could outperform state-of-the-art traffic forecasting baselines (ARIMA, SVR, FNN, LSTM, DCRNN [6], Conv+LSTM, spectral graph convolutional LSTM (SGC+LSTM), and localized SGC-LSTM). Also, Their proposed model was able to identify the influential roadway segments in traffic networks, which is regarded as an aspect of the interpretability of a deep learning model.

Wu et al. [34] enumerated two shortcomings in previous studies and propose the Graph WaveNet model for addressing these challenges. The two challenges they tried to address were 1) using a fixed graph structure and 2) using RNNs and CNNs for capturing temporal dependencies, which hinders modeling long-range temporal sequences. Graph WaveNet consists of several spatial-temporal layers. Each layer is comprised of a gated temporal convolution module followed by a graph convolution layer. Graph WaveNet learns an adaptive dependency (adjacency) matrix and assembles the standard graph convolution with dilated casual convolution to learn the graph structure dynamically and handle long sequences. The effectiveness of Graph WaveNet was evaluated by comparing it to the baselines (ARIMA, FC-LSTM, WaveNet [35], DCRNN, GGRU [29], and STGCN [27]).

To address the multi-step (and relatively long-term) prediction of traffic conditions, Zheng et al. [36] proposed GMAN, a graph multi-attention network, which adopts a spatial-temporal encoder-decoder architecture coupled with a transform attention layer between the encoder and decoder. GMAN aimed at addressing three common challenges in traffic forecasting: 1) dynamic spatial correlation, which means near locations in the traffic network graph are not necessarily highly correlated, 2) nonlinear temporal correlations, which means traffic conditions at one-time step are not always correlated with the traffic conditions in most recent time intervals, and 3) error propagation, which means small errors in each time step might amplify when we predict further into the future. In comparison to ARIMA, SVR, FNN, FC-LSTM [28], STGCN [27], DCRNN [6], and Graph WaveNet [34], GMAN showed to perform better, especially in predicting the farther future.

Wu et al. [37] also tried to consider a dynamic and flexible graph structure instead of assuming a fixed one. Their framework consists of three main components: 1) a graph learning layer, which adaptively extracts the adjacency

matrix, 2) graph convolution modules, specifically designed for directed graphs, for considering spatial dependencies among variables, and 3) temporal convolution modules, each of which consists of two dilated inception layers for handling large sequences. These modules utilize modified 1D convolutions for extracting sequential patterns of time series data. They compared their model with baselines for both single-step (autoregressive, VAR-MLP [38], Gaussian Process [39], RNN-GRU, LSTNet [40], TPA-LSTM [41]) and multi-step (DCRNN, STGCN, Graph WaveNet, ST-MetaNet, GMAN, and MRA-BGCN [42]) forecasting, and reported the overall competence of their proposed framework).

Chen et al. [43] argued that previous traffic forecasting models only consider limited and static external factors into account. Therefore, they proposed AARGNN, an attentive attributed recurrent GNN, for considering multiple static and dynamic factors during the traffic forecasting process. More specifically, they considered road network topology, driving distance, points of interest (POI), road physical properties, and incident data as the link-level features; traffic state data as the node-level feature, and weather and data information as the graph-level features. They also used an attention mechanism to identify the contribution of each factor to the prediction task. They achieved better accuracy in comparison to state-of-the-art models such as DCRNN, TGC-LSTM, and GMAN. There have been several other studies employing different types of graphs and mechanisms for traffic forecasting in recent years. As the number of studies is significantly high, we just here summarize some of these studies that proved to have more novelty and have been cited by other researchers more frequently. For a comprehensive review of current studies on traffic forecasting, readers are referred to [19] and [8].

A summary of studies on traffic forecasting using graph neural networks is presented in Table II. The studies are categorized based on their spatial and temporal modules, graph structure, number of nodes, baselines with which they have been compared, learning design, and features/factors that have been considered by the model.

Although a great deal of research in the transportation domain has been focused on traffic forecasting, there are still open research areas that need further investigation. Arguably, the real-world applicability of proposed models is one of the most important questions that need to be answered. Almost all previous studies have been developed and evaluated based on a relatively small network (sub-network) of a real urban transportation network. Based on the design of the framework, increasing the number of nodes in the graph may exponentially or linearly increase the complexity of the framework and the computational costs. Therefore, one important future research direction would be evaluating the applicability of current GNNs in real-world large networks.

Moreover, most graphs in the traffic forecasting domain are defined based on the topology of the network and take into account parameters such as connectivity, distance, and proximity. It would be beneficial to also consider other features and factors, such as land use and functional similarity of regions and links, presence of public transport stops

and stations, presence and types of intersections and other bottlenecks, and so on. This also helps to visualize the connections among the nodes in the graph in a more realistic manner.

Transfer learning could also benefit the applications of GNNs in traffic forecasting problems. Considering that many urban areas suffer from insufficient historical traffic data for training deep learning models, transfer learning could speed up the development of graph-based deep learning models in such areas. Nevertheless, the transferability of GNN-based frameworks for traffic forecasting has not yet been extensively studied and requires special attention. It is not practical to train every model for new cities or areas from the scratch, even if the required historical data would be available.

Evaluating the performance of traffic forecasting models when facing missing or noisy data and their robustness against unexpected events has not been extensively studied, while such disruptions in data collection endeavors are common in urban data sensing. Similarly, most of the currently developed models are trained and tested based on normal and periodic traffic patterns, and their performance under irregular traffic patterns has not been reported.

Using multiple sources of data for improving traffic prediction accuracy and robustness is another interesting research area that has been mainly neglected in previous studies. More flexible graph types, such as heterogeneous and multiplex graphs could be utilized to accommodate heterogeneous traffic data coming from different sensors (such as traffic loop detectors, cameras, and connected vehicles, to name a few).

Finally, graph-level prediction can be encouraging for traffic management and decision-making purposes. For instance, different areas of a city can be considered as graphs (or sub-graphs) and aggregated congestion indices could be predicted in short-term and mid-term time intervals. These indices can aid urban planners and decision-makers to adopt real-time strategies for regulating traffic congestion within different regions of a city, and due to their aggregated nature, these types of models will probably not suffer from high computational costs.

All in all, GNNs have shown to be promising for traffic forecasting due to their high flexibility in capturing multi-dimensional, dynamic, and non-Euclidean patterns in traffic datasets, and it seems they will play a major role in future traffic forecasting endeavors.

B. Demand Prediction

Demand prediction is another area of interest among researchers and decision-makers in transportation and urban planning. Traditionally, travel demand forecasting was almost done for mid-term or long-term decision-making and transportation systems developments [44]. However, with the advent of modern modes of travel, such as shared or on-demand mobility systems, short-time travel demand prediction has increasingly drawn the attention of researchers [45], [46], [47]. A great variety of methodologies, including time series analysis methods to machine learning and deep

learning frameworks, have been used for travel demand modeling. Recently, the use of GNNs in travel demand prediction has opened doors for considering complex and dynamic non-Euclidean spatial-temporal dependencies in large-scale travel demand prediction. In the following, the studies in various domains that have utilized GNNs for travel demand estimation and prediction are briefly introduced and discussed.

1) *Ride-Hailing Services*: The emergence and evolution of Ride-hailing services (such as Uber and Lyft) have dramatically changed people's travel behavior. The rapid adoption of these services has posed numerous challenges for transportation planners, researchers, and decision-makers because there has been little information about the changes in travel behaviors and patterns originated by such technologies. On the other hand, accurate prediction of travel demand and patterns is crucially important for transportation network companies (TNC) as they need these data for their vehicle dispatching and distribution. Accordingly, a great deal of research has focused on forecasting travel demand for (shared) ride-hailing services [48], [49], [50]. In this section, we focus on studies using graph neural networks for demand prediction in ride-hailing service contexts.

Bai et al. [51] proposed a framework, namely spatial-temporal graph to sequence model (STG2Seq) for multi-step city-wide passenger demand forecasting. They defined the connectivity in the graph according to the similarity of the passenger demand patterns (instead of geographic locations). Also, their framework consisted of two separate encoders (long-term and short-term) that operated simultaneously to make multi-step predictions without using RNNs. This approach prevents the accumulation of errors and information oblivion. The long-term encoder comprises a series of gated graph convolution modules (GGCN) and considers previous long-term timesteps aiming to capture the historical spatial-temporal patterns. The short-term encoder, however, is used for integrating the predicted demand for the purpose of multi-step prediction using a short-term sliding window over previously predicted demands. They evaluated their framework using three real datasets and by comparing against HA, ordinary linear regression (OLR), XGBoost [52], DeepST [53], ResST-Net [54], DMVST-Net [55], ConvLSTM [56], FCL-Net [57], FlowFlexDP [58], DCRNN [6], and STGCN [27].

Geng et al. [50] utilized a spatial-temporal multi-graph convolutional network (ST-MGCN) to consider multiple spatial correlations (such as neighborhood, functional similarities, and network connectivity) for region-level ride-hailing demand forecasting. They evaluated the performance of the framework on two real datasets and reported on average ten percent of improvement in error reduction compared to baseline models (HA, LASSO, auto-regressive model [59], gated boosted machine [60], ST-ResNet [54], DMVST-Net [55], DCRNN [6], and ST-GCN [27]).

Unlike many similar previous studies that had focused on zone-based demand prediction, Hu et al. [61] proposed a graph embedding-based multi-task learning (GEML) framework for predicting the origin-destination matrix in a ride-hailing service context. A grid embedding part, which considers

TABLE II

AN OVERVIEW OF MOST CITED STUDIES UTILIZING GNNs FOR TRAFFIC FORECASTING, WHICH HAS BEEN USED AS BENCHMARKS IN OTHER STUDIES

Model	Study	Spatial Module	Temporal Module	Adjacency Matrix	Nodes	MultiStep	No. Nodes	Compared Models	Learning Design	Category	Extra Features
STGCN	[27]	Graph CNN	Gated CNN	Connectivity, Distance Matrix	Sensors	-	228,1026	GCGRU	Node Level	Speed	-
DCRNN	[6]	Spectral Graph Convolution	Diffusion Convolutional Gated Recurrent Unit	Proximity	Sensors	+	207, 325	-	Node Level	Speed	-
T-GCN	[30]	Graph convolutional network	Gated recurrent unit	Connectivity	Roads	+	156, 207	-	Node Level	Speed	-
ASTGCN	[32]	Spatial Attention + Graph Convolution	Temporal Attention + Convolution	Connectivity	Sensors	+	307, 170	STGCN, GLU-STGCN	Node Level	Traffic Flow	Daily, Monthly, Weekly Temporal Trends
ST-MetaNet	[33]	Meta graph attention network	Meta RNN	Connectivity	Sensors	+	1024, 207	Seq2Seq, Seq2seq, DCRNN	Node Level	Taxi Flow, Traffic Speed	No. POIs, No. roads and lanes
TGC-LSTM	[4]	Graph convolutional network	LSTM	Connectivity	Sensors	-	323, 1014	-	Node Level	Speed	-
Graph WaveNet	[34]	Graph convolutional network	Gated Temporal Convolution	Connectivity	Sensors	-	207, 325	DCRNN, STGCN	Node Level	Speed	-
GMAN	[36]	Node2vec + fully-connect convolutions + Spatial Attention	Temporal Attention	Proximity	Sensors	+	95, 325	Graph DCRNN, STGCN	Node Level	Volume, Speed	-
MTGNN	[37]	Graph Learning Module	Temporal Convolution layer	Connectivity	Sensors	+	862	DCRNN, Graph GMAN	Node Level	Speed	-
MFFB	[64]	Gated Recurrent Unit	Learning during the model	Learning during the model	Sensors	+	200, 325	DCRNN, GCGRU	Node Level	Traffic Flow, Speed	Weather, Events
GLSTGCN	[65]	Graph Convolutional Network	Dilated causal convolution network	Graph learning module	Sensors	+	228, 50	DCRNN, Graph STGCN, WaveNet, ASTGCN	Node Level	Speed	-
TmS-GCN	[66]	Graph Convolutional Network	Gated Recurrent Unit	Weighted combination {Simple Connectivity, Traffic flow propagating, Distance Matrix}	Region	+	78	T-GCN	Node Level	Speed	-
MW-TGC	[31]	Graph Convolutional Network	LSTM	Demand Matrix	Roads	+	304, 1007	Seq2Seq, TGC-LSTM, ST-GCN	Node Level	Speed	-
DGCN	[67]	Attention layer	Temporal Convolution Layer + Graph Temporal Convolution Layer	Laplace Matrix of Feature Sampling and Spatial Attention	Roads	+	307, 170, 397	GCRU, Gated-STGCN, ASTGCN	Node Level	Traffic flow, Speed, occupancy	Recent, Weekly Trends
AAGRNN	[43]	Graph Convolution	LSTM + Attention Layer	Distance, Functional Similarity, Geographic, and Incident matrices	Sensors	-	362, 202	DCRNN, FC-LSTM, TGC-LSTM, GMAN	Node Level	Traffic Flow	Weather and calendar features

non-overlapping grids of equal sizes, is for capturing spatial correlations in the whole region. The flows from one grid to the others are modeled by inspiring from message passing and neighborhood aggregation functions in graph convolutions, and the grids embeddings are learned by aggregating the features of the connected grids (connectivity is defined as geographical closeness and passenger flows between the grids). Afterward, a multi-task learning framework considers a sequence of multiple embedding vectors for each grid in the past for capturing temporal patterns. In the last step, by defining a transition matrix, the OD matrix for the study area is estimated. Comparing the proposed model with the baselines (HA, LSTM, LSTNet [40], and GCRN [62] on two real ride-hailing datasets, they concluded that their model could outperform the baseline models.

Guo et al. [63] also proposed a deep learning framework called spatial-temporal encoder-decoder residual multi-graph convolutional network (ST-ED-RMGC), to predict the OD-based demand for on-demand ride-sourcing services. However, they used a different approach for defining the OD graph, where each vertex in the graph represents an OD pair, and each edge represents the connections between OD pairs. Their framework is composed of two encoders for spatial and temporal encoding; the spatial encoder consists of several

RMGCs, and the temporal encoder utilizes an LSTM model. The decoder part also uses several RMGCs to transform the encoded information into an output OD graph. They evaluated their framework on a real dataset and compared its performance to the baselines including HA, XGB, multi-layer perceptron (MLP), GBDT, RF, LASSO, LSTM, Spatial LSTM, a multi-graph convolution network (MGC), and encoder-decoder multi-graph convolution network (ED-MGC), and a residual multi-graph convolution network (RMGC).

By constructing multiple interpretable virtual graphs, Jin et al. [68] developed a framework called DMVST-VGNN (a deep multi-view spatial-temporal virtual graph neural network) to forecast citywide ride-hailing demand that overcomes the limitations of spatial data sparsity in fine-grained prediction. In order to improve the learning capabilities of long sequences, both long- and short-term temporal dynamics were considered in this framework. In particular, the DMVST-VGNN utilized structures of 1D CNN for the purpose of short-term temporal dynamics modeling, multi-graph attention neural network for the purpose of spatial dynamics modeling, and transformer networks for the purpose of long-term temporal dynamics modeling. DMVST-VGNN demonstrated superior performance in the experiments compared with some state-of-the-art baseline

models, such as ST-ResNet, DCRNN, Graph Wavenet, Multi-GCN, and ST-MGCN.

Huang et al. [69] proposed DMGC-GAN (a dynamic multi-graph convolutional network with a generative adversarial network) in which they combined GANs and GNNs in order to predict OD-based ride-hailing demand. Their approach involved developing the temporal multi-graph convolutional network (TMGCN) layer containing different dynamic OD graphs to capture their spatial topologies in time and using GAN structure to overcome the high sparsity of OD demand data. The experimental results on the real-world ride-hailing demand dataset from the Manhattan district, New York City demonstrated that the proposed model in this study performed the best against nine baseline models such as T-GCN and TMGCN.

In another study, a multi-task matrix-factorized graph neural network model (MT-MF-GCN) has also been proposed by Feng et al. [70] in order to predict both zone-based and OD-based demand simultaneously in ride-hailing services. Two major components make up the proposed model; the GCN basic module, which captures the spatial correlations among zones via a mixture-model graph convolutional network, and the matrix factorization module, which is utilized for multi-task predictions of zone-based and OD-based demand. The study demonstrated that the proposed model outperformed the state-of-the-art baseline methods, such as GraphSAGE, GEML, and ST-GCN, in both zone- and OD-based predictions using real-world data from Manhattan and Haikou.

2) *Bike Sharing Systems*: In recent years, shared bike systems have become popular in many cities around the globe. Generally speaking, these systems could be grouped into dock-based and dockless systems. In dock-based systems, the bikes are taken and returned to predetermined stations, while in dock-less systems, travelers are free to leave the bike everywhere they want. In either of these systems, predicting the travel demand is crucially essential for proper distribution and rebalancing of bikes to serve the near future demand. For instance, in a dock-based system, inappropriate demand prediction might lead to empty stations against overcrowded stations. In the last decade, several studies have paid attention to bike-sharing system travel demand prediction [71], and in recent years, GNN-based frameworks have drawn the attention of researchers due to their ability to incorporate spatial-temporal dependencies in large-scale.

In an early attempt, Lin et al. [72] proposed GCN-DDGF a graph convolutional neural network with a data-driven graph filter for capturing pairwise correlations between bike-sharing stations and learning the graph structure instead of assuming a predefined structure. They developed four variants of the proposed model (GCN) by employing four types of predefined adjacency matrices, including spatial distance matrix, demand matrix, average trip duration matrix, and demand correlation matrix. They concluded that their proposed data-driven filter can capture some hidden correlations among stations that were not revealed by any of the predefined adjacency matrices.

Chai et al. [73] proposed a multi-graph convolution neural network model for station-level bike flow (demand) prediction.

Their framework consists of three successive layers: 1) a multiple graph generation layer (for considering heterogeneous and multiple inter-station relationships), followed by 2) a multi-graph convolution layer that consists of a graph fusion part for merging different graphs into one graph, and 3) a prediction network that is composed of an LSTM encoder-decoder for temporal correlations and a fully connected network for confidence estimation. The bike-sharing system in this study is represented as a weighted graph, in which weights represent the strength of relations between stations based on three the distance between stations, interaction (flows) between two stations, and the correlation between inflow and outflow at stations. They evaluated their model using two real datasets and the baseline models (HA, ARIMA, SARIMA, gradient boosting regression tree, and LSTM) and reported around 25% reduction in prediction error.

Xiao et al. [74] proposed an end-to-end deep learning framework based on STGCN. Their framework consists of two spatial-temporal convolution units followed by a fully connected layer. Each spatial-temporal unit is comprised of a spatial convolution layer based on a graph convolutional neural network, surrounded by two temporal convolution layers constructed based on the gated convolutional neural network for representing temporal dependencies (similar to [27]). They evaluated the performance of their framework by comparing its performance in predicting pick-up and returning demand with simple RNN, LSTM, and GRU models, and their results indicated their framework outperformed the baseline on both picking-up and returning demand prediction. Also, the time needed for training their model was significantly lower compared to the baselines.

By incorporating historical bike-sharing trip data, land-use data, weather data, and users' personal information, Ma et al. [75] proposed a spatial-temporal graph attentional long short-term memory (STGA-LSTM) to predict the pick-up and drop-off demands of shared bikes. The spatial information was mined using the combination of GCN and attention mechanism, and the temporal information was explored by using the combination of LSTM and attention mechanism. Furthermore, in order to construct the graph of bike-sharing stations, the authors used the demand connections among those stations. A learnable adjacency matrix was also used in the model for facilitating the construction process and describing the relationship between stations. Based on real data from Nanjing bike-sharing systems, the proposed model was evaluated and proved to be more accurate and efficient compared to baseline models, including the graph-based model GC-LSTM.

Another study by Li et al. [76] proposed a data-driven spatial-temporal graph neural network, called STGNN-DJD, to solve the bike demand and supply prediction problem by integrating two spatial-temporal graphs referred to as the flow-convoluted graph and the pattern correlation graph. These graphs were used to represent the flow relationships between stations at various time slots and the dynamic demand-supply patterns between stations, respectively. A graph neural network was then employed to generate embeddings for docked bike prediction based on flow-based and attention-based

aggregators. A comparison of the proposed model with other baseline models, including GCNN, MGCNN, ASTGCN, and STSGCN revealed that it performed better than the other baseline models.

3) *Passenger Flow Prediction*: Passenger flow prediction in public transportation systems, such as subways and bus rapid transit systems, has gained special attention from researchers in the transportation demand modeling area [77], [78]. Accurate prediction of passenger flows is crucially important for the real-time management of transit systems, as well as mid-term and long-term planning for the development of these systems. Among many applied methods, deep learning-based methods have drawn the special attention of researchers in this area [77], [78], [79], [80]. Due to the graph-like structure of public transportation networks (such as metro and bus networks), graph neural networks have also formed a new family of methods for predicting passenger flows in transit systems. This section briefly reviews the state-of-the-art in using GNNs in passenger flow prediction.

Although numerous studies using GNNs for passenger flow predictions exist in the literature, to the best of our knowledge, the early studies go back to no sooner than 2018. Li et al. [81] proposed a graph convolution neural network model by combining graph modeling to consider the interrelations between subway stations and CNN for the spatial-temporal modeling of passenger flow features. By separating the inflow and outflow volumes, they constructed two-channel graph matrices for different time scales and then integrated these matrices and extracted the spatial-temporal features using CNNs. Evaluating their model on a real subway dataset and comparing its performance with the baseline models (ARIMA, SARIMA, HA, vector auto-regressive (VAR), a fully connected station-temporal deep neural network (ST-ANN), ST-Res-Net [54]), they concluded their modeling framework could significantly improve the passenger flow prediction accuracy.

Han et al. [82] proposed STGCNmetro (spatial-temporal graph convolutional neural networks for metro) to predict inflow and outflow passenger counts citywide. They first defined the metro network as an undirected graph and captured the spatial-temporal dependencies among adjacent stations using stereogram graph convolution. They then constructed a deep GCN structure by stacking multiple GCN layers to capture the spatial-temporal dependencies between *distant stations*. The historical passenger flows are divided into recent, daily, and weekly patterns in order to consider time-varying temporal patterns. These three outputs are fused to compute the loss function. They compared their model with the baselines (multi-variate linear regression (MLR), LSVR, Bayesian regression, principal component analysis coupled with k-nearest neighbors (PCA-KNN), non-negative matrix factorization KNN (NMF-KNN), LSTM, and CNN) on a real-world dataset and reported Superior performance of their model.

Peng et al. [83] proposed a dynamic graph structure $G = f(V, E, W, A, t)$, where V refers to stations, E refers to a directed relationship between stations at time t , W refers to the weights of the edges at time t , and A refers to

the inflow and outflow of the stations. By this definition, they tried to model the dynamic traffic station relationships over time as spatial-temporal incident dynamic graphs based on historical passenger flow in the stations. Then, the developed dynamic-GRCNN, which is a dynamic graph recurrent convolutional neural network model, to learn the spatial-temporal features representations. They also sampled the training data based on short-term, mid-term, and long-term observations to capture the periodicities and trends in passenger flows. Validating their framework on several datasets (subway, bus, and taxi) and comparing their results with the baselines (HA, ARIMA, SARIMA, VAR, ST-ANN, DeepST [53], ST-ResNet [54], AttConvLSTM [84], DMVST-Net [55], DCRNN [6], STGCN [27], T-GCN [30], and GCNN [81]), they reported performance beyond the baselines.

Instead of constructing the passenger flow prediction graph solely based on the network topology, Liu et al. [85] proposes two new inter-station relationships: 1) inter-station flow similarity, which means two metro stations with similar passenger flow evolution patterns; and 2) inter-station flow correlation, which means the correlation between inflow or outflows between two stations (determined by historical OD distribution of ridership). Based on this, they proposed a physical-virtual collaboration graph network (PVCN) that constructs three graphs based on the physical topology of the network, inter-station flow similarities, and inter-station flow correlations. Next, these graphs are integrated into a collaborative gated recurrent module (CGRM). In the final step, a seq2seq model is used for sequential forecasting of passenger flow at the next several time intervals. They compared their model's performance on two real datasets of subway systems with the baselines (HA, random forest, GBDT, MLP, LSTM, GRU, ASTGCN [32], STG2Seq [51], DCRNN [6], GCRNN, Graph-WaveNet [34]) and reported the superiority of the proposed PVCN.

Chen et al. [86] incorporated a stacked bidirectional unidirectional LSTM network with a GCN and proposed the GCN-SBULSTM framework. They built a structured graph of the metro network with a k-hop matrix consisting of travel distance, flow volume, and station adjacency. The SBULSTM module is designed to simultaneously consider backward and forward temporal dependencies. Unlike many previous studies, the output of the GCN and SBULSTM modules are parallelly calculated and concatenated to avoid the distortion of temporal patterns (which occurs in ordinary sequential CNN-LSTM frameworks). Finally, they validated the effectiveness of their methodology using three ridership datasets over the state-of-the-art baseline models: LSTM, CNN, GCN, DMVST-NET [55], CNN-LSTM [80], SRCNs [87], SBULSTM [88], DCRNN [6], STGCN [27], Graph-WaveNet [34], and PVCN [85].

For predicting passenger flow in urban transit, He et al. [89] proposed an approach referred to as the multi-graph convolutional-recurrent neural network (MGC-RNN). the multi-graph in this study represents the inter-station correlations impacted by different factors, such as points of interest (POI) information, network structure, network

distance, operational information, and recent flow correlation. Afterward, multiple GCNs are used to extract correlations from each graph. Moreover, this study utilizes LSTM_encoder-decoder architectures to extract temporal dependencies based on exogenous factors such as national public holidays, and days of week.

Wang et al. [90] attempted to use the hypergraph concept with hyper-edges to take into account the connection between different stops of the same line. Specifically, this study utilized two types of hypergraphs: primary hypergraphs and advanced hypergraphs. The primary hypergraph pictured the fundamental topology of a metro network and was constructed with stations as vertices and lines as hyperedges, connecting stations on the same track. The Advanced Hypergraph revealed additional spatial information regarding the OD pattern of passengers in different time zones, including daily, weekly, and hourly. A real-world experiment was conducted on the metro datasets of Beijing and Hangzhou in China and the authors demonstrated that their model outperformed some state-of-the-art non-graph and graph-based methods, including DCRNN [6] and STGCN [27].

4) *Multi-Modal Demand Prediction Studies*: multi-modal modeling of transportation demand is another interesting area in demand prediction, in which the interactions between the demands of different modes of transportation are considered. Despite its great importance in urban demand modeling, only a few studies have addressed this class of demand modeling problems. Ke et al. [91] attempted to predict multimodal ride-hailing demand, arguing that demands for different modes are correlated, and historical observations of demand for one mode can help predict the demand for other modes. They approached the problem by using a number of multigraph convolutions (MGC) that were used to predict the demand for each mode separately. Additionally, multitask learning modules were utilized to share knowledge across multiple MGC networks. A real ride-hailing dataset for Manhattan that included solo and shared ride demand was used to evaluate the performance of the proposed framework.

Using a different approach, Liang et al. [92] proposed a multi-relational spatial-temporal graph neural network (ST-MRGNN) capable of predicting multimodal demand with heterogeneous spatial units. The authors introduced a multi-relational graph neural network (MRGNN) to capture cross-modal spatial dependencies using inter-modal and intra-model graph convolutions. Also, an attention-based aggregation module is used for summarizing different relationships. The performance of the proposed method has been evaluated using the data of subway and ride-hailing in New York City, which demonstrated improved performance over existing methods like STGCN [27], Graph-WaveNet [34] and MGCN [50].

5) *General Studies*: Graph neural networks have also been used for travel demand prediction on traditional problems. Hu et al. [93] proposed an end-to-end learning framework to forecast stochastic origin-destination (OD) matrices by addressing two main challenges: data sparseness, and spatial-temporal correlations. They addressed the data sparseness problem by factorizing the sparse OD matrix into two small dense matrices with latent features from the

source and destination regions. Also, to address the spatial-temporal correlations, they combined graph convolutions with recurrent neural networks to simultaneously model spatial and temporal correlations, respectively. Finally, the two dense matrices are multiplied to achieve the full predicted OD matrix. They validated their model on two real trajectory datasets by comparing with the baselines, including RNN using GRU gates [94], multi-task representation learning (MR) [95], naive histograms (NH), Gaussian process regression (GP) [96], multi-variate vector auto-regression [97], and reported that their model outperformed all baselines on both datasets.

A summary of the studies in travel demand prediction is presented in Table III. The studies are categorized based on their prediction type, learning design, the nature of their adjacency matrix, the spatial and temporal modules, the number of nodes in their case study, compared models, and extra features. Also, modules and mechanisms incorporated in each modeling framework are included in the table.

Reviewing the GNNs used for travel demand forecasting reveals that most of the current studies have focused only on one mode of transportation, while graph neural networks provide a great opportunity for multi-modal transportation demand modeling [91], [92]. This is important because there are high correlations among the travel demands between different transportation modes, and focusing on just one mode might not be the most appropriate approach. Also, graph neural networks could be used for unraveling the spatial-temporal inter-correlation among travel demands of different public transportation modes (bus, subway, and ride-hailing systems).

In addition, most of the studies on public transportation have been focusing on metro passenger flow prediction. Metro networks have special characteristics, which make them different from bus networks. For instance, bus transportation usually shares a significant portion of its route with vehicular traffic, which itself is a highly dynamic phenomenon and can greatly affect the performance of buses within the network. This will also influence the capacity and demand of transportation systems, which has been mainly neglected in previous studies. Also, bus networks are usually more interconnected (more stops and routes) with more sparse data compared to metro systems. This makes the modeling endeavor for bus systems much more complex. This is where GNNs can play an important role, but their power in considering highly complex public transportation networks has not yet been fully investigated.

The treatment of missing and sparse data in this domain is also noteworthy, given the fact that there are many large networks of transportation in the cities in which there is always a substantial amount of missing or sparse data (especially in bus networks). This issue requires further investigation

Finally, an integral part of graph models is temporal modeling. The problem with the temporal modeling of GNN-based models is that most models have to take into account an equal time interval all over the network. This leads to considering the same behavior for stops or stations with different headways of the transportation mode, which is not the case in reality. Accordingly, it would be beneficial to consider

these differences into account when developing the temporal modules of GNN frameworks in future studies.

C. Autonomous Vehicles

Autonomous vehicles, also known as automated vehicles or self-driving cars [98], are expected to play an important role in the future of smart cities. Since the first Grand DARPA Challenge in 2004 and its subsequent challenges in 2005 and 2006 [99], learning-based methods have shown promising ability in dealing with the complexity of urban environments, and therefore, many research institutes and industrial companies initiated utilizing machine learning (ML) methods for the operation and control of autonomous vehicles. In recent years, GNNs have also been widely used in different applications of AVs, and in this subsection, we review the mainstream of these studies. As different sub-systems in an autonomous vehicle play specific roles, this section is also subdivided into three main sub-sections, namely, perception, motion prediction, and motion planning.

1) *Perception*: The perception mechanism is the first and one of the most challenging parts of designing and developing autonomous vehicles. This module aims at detecting and classifying objects surrounding the ego vehicle. Two main tasks for assuring the safe operation of autonomous vehicles are semantic segmentation and classification, and object detection and tracking [100]. Semantic segmentation is the task of clustering and assigning a particular class to a set of pixels in a picture or a point cloud. Point clouds are efficient 3D representations of real-world objects and have become increasingly popular in recent years with different applications, including the navigation of autonomous vehicles. These representations (measurements) are usually made by 3D sensors or Light Detection and Ranging (LiDAR) technology and are crucially important in object detection and identification of AVs' motion planning. Traditional deep learning methods often convert point clouds to 3D voxel grids (voxels are similar to pixels in 2D images) or a collection of images in order to prepare them for being fed into deep neural networks. However, these methods have shown to be inefficient due to information loss and computational costs [101], and many researchers have argued that graph representation of point clouds is an efficient yet accurate method to deal with such data. In the following, we overview the studies that utilized GNNs for point cloud or image segmentation and object detection, focusing on AVs.

Te et al. [101] were among the firsts who utilized graph neural networks analyzing point clouds. They considered the feature of points in a point cloud as signals on the graph and updated the Laplacian matrix of the graph in each layer of their model, RGCNN, according to the corresponding learned features to adaptively capture the structure of dynamic graphs. Moreover, they added a graph-signal smoothness function to regularize the learning process. RGCNN consists of two main parts, one for feature extraction using graph convolution, and the other for segmentation and classification using MLP and a combination of max pooling and MLP, respectively. They evaluated the performance of their model

by using the ShapeNet part dataset [102] and comparing it to the state-of-the-art baselines (VoxNet [103] (Classification only), ShapeNet (Segmentation Only) [102], PointNet [104], PointNet++ [105], and SyncSpecCNN (for segmentation only) [106]) and reported competitive performance with lower computational complexity.

Later, in an attempt to recover the topological information from point clouds, Jakub et al. [107] developed DGCNN, a dynamic graph convolution neural network, by utilizing a simple operator called EdgeConv. EdgeConv is used to dynamically construct k-NN graphs in each layer of the network. It generates edge features describing the relationships between a point and its neighbors. EdgeConv can consider local neighborhood information, and, at the same time, can be applied to learn global shape characteristics. They also compared the performance of their framework with popular baselines, such as PointNet [104], PointNet++ [105], and concluded that GNNs can significantly improve the semantic segmentation accuracy over CNN-based approaches.

Jin et al. [108] argue that applying convolution operators to point clouds is not efficient because point clouds are not evenly distributed over grids. Therefore, they proposed a GNN-based framework, namely Point-GNN, for object detection from LiDAR data. They first translated the point clouds into a graph and then utilized graph neural networks for predicting the category and shape of the objects belonging to each node of the graph. They compared their method with the state-of-the-art models on both the 3D and Bird's Eye View object detection datasets and reported an overall superior performance.

Baghbani et al. [109] explain that previous machine learning approaches that learn semantic representations from HD maps have two shortcomings: their rasterization process results in more or less information loss, and using a 2D convolution might be insufficient for capturing the complex topologies of maps. As an example, lane pairs of opposite directions have completely different semantic meanings, although they are spatially close together, and this is where GNNs could be utilized. In their proposed method, a lane graph is constructed instead of rasterizing the HD maps to avoid information loss, and thereafter, a graph convolutional network is employed to consider the complex topological interactions.

Zhao et al. [110] proposed a convolutional vicinity aggregation graph neural network (CVA-GNN) for point cloud classification. Convolutional vicinity abstraction (CVA) is a module that extracts features from the points' vicinity in a hierarchical way. The extracted features are then translated into graph embeddings. The novelty of this module is that it considers the inter-relations at two levels: between successive neighbors in the convolution layer and between all neighbors at the aggregation level. They reported superior performance in the classification task compared to the state-of-the-art baselines on the ModelNet40 dataset, which is a popular reference dataset in point cloud classification.

Zou et al. [114] presented a multi-task Y-shaped graph neural network, MTYGNN, for exploiting 3D point clouds. MTYGNN has two branches for performing the classification and segmentation tasks in point clouds at the same time. To increase the accuracy of the segmentation task, the

TABLE III

AN OVERVIEW OF MOST CITED STUDIES UTILIZING GNNs FOR DEMAND PREDICTION, WHICH HAVE BEEN USED AS BENCHMARKS IN OTHER STUDIES

Model	Study	Spatial Module	Temporal Module	Adjacency Matrix	Nodes	Multi-Step	No. Nodes	Compared Models	Learning Design	Category	Extra Features
ST-MGCN	[50]	spatial-temporal multi-graph convolution network	Contextual Gated Recurrent Neural Network (GCRNN) - GRU	Multi-graph {Neighborhood, Functional Similarity, Connectivity}	Grids	-	1296, 896	ST-ResNet, DMVST-Net, DCRNN, STGCN	Node Level	Ride Hailing	-
GCNN-DDGF	[82]	Graph Convolutional Network (GCNN) with Spectral Convolution	LSTM	Combination {Distance Matrix, Demand Matrix, Average Trip Duration Matrix, Correlation Matrix}	Bike Stations	-	272	-	Node Level	Bike Sharing	-
Multi-graph CNN	[73]	Multi Graph Convolution Network	Encoder-Decoder with LSTM	Multi-graph {Distance Matrix, Demand Matrix, Correlation Matrix}	Bike Stations	-	-	-	Node Level	Bike Sharing	Temperature, Wind speed, Day of week
GEML	[61]	Graph Convolution	Multi-Task LSTM	Weighted Combination {Distance Matrix, Demand Matrix}	Grids	-	-	GCRN	Link Level	Ride Hailing	Weather
STG2Seq	[51]	Gated Graph Convolutional Module	Gated Graph Convolutional Module	Demand Correlation Matrix	Stations	+	-	DCRNN, STGCN, ST-ResNet	Node Level	Ride Hailing, Bike Sharing	-
Dynamic-GCRNN	[83]	Graph Convolution	LSTM	Dynamic Demand Matrix	Stations	-	329, 2321, 128	DCRNN, STGCN, T-GCN, GCNN	Node Level	Ride Hailing Flow Prediction	-
STGCNN-metro	[111]	Graph Convolution	Graph Convolution	Connectivity Matrix	Metro Stations	-	289	-	Node Level	Passenger Flow	Weekly-Daily Temporal Trends
GCNN	[81]	Graph Convolution	Graph Convolution	Demand Matrix	Metro Stations	-	268	ST-ResNet	Node Level	Passenger Flow	Weekly-Daily Temporal Trends
PVGCN	[85]	Graph Convolution	Gated RNN	Weighted {Connectivity, Correlation, Demand Matrix}	Metro Stations	+	288, 80	ASTGCN, DCRNN, STG2Seq, GCRNN, Graph-WaveNet	Node Level	Passenger Flow	-
GCN-SBULSTM	[86]	Graph Convolution	Stacked bidirectional-LSTM	Weighted {Connectivity, Distance Matrix, Demand Matrix}	Metro Stations	+	166, 80, 288	DCRNN, STGCN, Graph-Wavenet, PVGCN	Node Level	Passenger Flow	-
BNG-ConvLSTM	[112]	Graph Convolution	LSTM	Weighted {Connectivity, Proximity}	Bus Stops	-	929	-	Node Level	Passenger Flow	-
MFST-DMGCN	[113]	Dynamic multi-graph convolutional network	Convolution GRU	Connectivity matrix	Metro Stations	+	301	TGCN, STGCN, AST-GCN, Graph WaveNet	Node Level	Passenger Flow	-
MGC-RNN	[89]	Multi-graph Convolutional	LSTM-encoder-decoder	Multi-graph {Distance, Functional Similarity, Correlation, Operational Information, Geographic}	Metro Stations	+	118	-	Node Level	Passenger Flow	Day-of-Week, Holiday, Hour-of-Day, Weather
STGA-LSTM	[75]	Graph Convolution + attention	LSTM + Attention	Learnable Parameter	Bike Stations	-	-	-	Node Level	Bike Sharing	Land-use, Weather, Passenger information

classification prediction is then combined with the semantic features. They applied their framework to several datasets and reported superior performance compared to popular baseline models, RGCNN [101], DGCNN [107], and LDGCNN [115].

AGNet was proposed [116] by using an attention-based feature extraction module called AGM, which constructs a topology structure in the local region and aggregates the important features using an attention-pooling operation. In this framework, the local feature information is extracted by constructing a topological structure, which facilitates better extracting of the spatial information with different distances.

Some other researchers have also utilized GNNs for other applications in the perception subsystem or AVs. Meyer et al. [117] employed graph neural networks for analyzing raw radar data coming from autonomous vehicle sensors. They argued that utilizing radar data that is robust to adverse weather conditions can improve the redundancy and robustness of the perception task. They used graph neural networks because radar signals fade into adjacent cells and are not propagated just locally. In this way, they presented a network for turning the raw radar data into 3D objects. They reported a 10 percent improvement in object detection accuracy.

To wrap up, GNNs have shown promising performance in cloud point analysis and the perception subsystem of AVs. They facilitate the handling of sparse and irregular point cloud data and take into account the interrelations between neighboring points in point clouds or pixels in images. However, many of the algorithms developed in the literature are trained and tested on datasets for non-urban environments, and their applicability in complex urban areas is still under question. More comprehensive studies are needed to test the accuracy and applicability of developed algorithms in urban environments, especially when experiencing severe lighting and weather conditions. Also, most of the studies focusing on urban areas have utilized bird-eye view pictures or videos, while in real-world scenarios, such information and pictures are not usually available. Therefore, it is essential to develop and test future models on images from the vehicle's perspective.

In addition, extracting the topological information about road infrastructure and reasoning about the semantic information of such infrastructures (such as lane direction, traffic signal status, signs' meaning, etc.) is of great importance. Previous attempts in recognizing traffic signs and signal state detection are mainly made using CNNs, and if GNNs are going to be implemented in image recognition and classification

tasks, such more in-depth analyses should be also included via utilizing solely GNNs or in combination with traditional CNN-based modules. Moreover, studies show that the GPU power needed for graph-based semantic segmentation algorithms is higher than traditional methods [118]. This is especially important for their application in the AV industry in terms of implementation costs and real-time applications in highly complex urban environments, such as intersections where real-time processing of information is crucially important.

Finally, combining different sources of data (such as images and point clouds) for semantic segmentation and improving the robustness of the results may be beneficial. Multi-dimensional and heterogeneous graphs could be used for such purposes. Also, the applicability of the developed models when the sensor outputs are noisy or faulty has not been extensively studied and is crucial for real-world application data-driven algorithms.

2) *Motion Prediction*: Motion prediction or trajectory prediction is concerned with predicting the future trajectories for the surrounding objects of an autonomous vehicle. These objects could be other vehicles, cyclists, and pedestrians. Motion prediction plays a crucial role in the safe operation of autonomous vehicles, and therefore has become one of the most popular topics in recent years [119]. Due to the complexity of the problem and the high number of objects, especially in urban areas, deep learning methods have constituted the main research line in this field. And not surprisingly, GNNs are becoming popular in the area thanks to their ability in unraveling complex interrelations between objects.

Li et al. [120] argued that the previous RNN and CNN-based methods for trajectory prediction only focus on one vehicle and ignore the interactions among adjacent objects. Therefore, they constructed an undirected graph of the ego vehicle and its neighboring objects, in which each object is a node in the graph. Edges in this graph are defined in two ways. Firstly, objects that are at a certain distance from the ego vehicle are connected to each other (spatial connection). Secondly, each object is connected to itself in the past and future time steps (temporal connection). By applying a graph convolutional model and a two-layer encoder-decoder LSTM model, they achieved a 30 percent improvement in the state-of-the-art motion prediction performance and five times less computational time.

Casas et al. [121] proposed SpAGNN, a spatially-aware graph neural network model for simultaneous object detection and behavior forecasting. They built a fully connected directed graph of the actors in the scene and employed a 3-layer MLP for message passing. The bi-directionality allows their model to capture the asymmetric relationships between pairs of vehicles (for example, the follower and the leader vehicles have different impacts on each other). Moreover, they developed a probabilistic relational behavior forecasting model, inspired by Gaussian Markov random field (Gaussian MRF) and utilizing graph neural networks.

Jeon et al. [122] aimed at developing an efficient and scalable framework that preserves a high prediction performance for a large number of vehicles. To this end, they

proposed SCALE-Net based on edge-enhanced graph neural network (EGNN) [123]. EGNN updates the node feature using an attention mechanism induced by edge features of the neighboring nodes. However, they state that their model still cannot consider the road structures.

In a later work, Casas et al. [124] aimed to characterize the joint distribution over motion forecasts of multi-actors using an implicit latent variable model (ILVM). In order to overcome some challenges such as roads' complex geometries, the environment's partial observance, and the variable number of actors in the scene, they utilized an interaction graph, in which nodes are the actors in the scene. They then leveraged GNNs to encode the scene into a latent space (learning a distributed latent representation of the scene), and afterward, decoded the latent samples into socially consistent trajectory forecasts. They reported state-of-the-art performance in motion forecasting and capturing complex interactions.

Mo et al. [125] combined RNN with GNN to develop a new method for trajectory prediction. The RNN is utilized to model the historical and dynamic features of vehicles, and GNN is used to capture the interaction among them. Also, a third RNN-based module serves as a decoder and jointly considers the historical dynamics and the interaction feature among vehicles for making predictions.

Sheng et al. [126] applied a GCN for capturing spatial interactions among neighboring vehicles, and a CNN for tackling temporal correlations among features. The spatial-temporal features are encoded-decoded via a GRU network in their framework. Singh and Rajeev [127] also employed a multi-scale GNN coupled with an LSTM-based encoder-decoder to fulfill the trajectory prediction task.

Among different tools, the attention mechanism has been one of the most popular ones applied to many of the GNN-based frameworks in the motion forecasting of AVs. Chen et al. [128] utilized the attention mechanism to consider the varying social interaction between vehicles in the scene. Carrasco et al. [129] utilized a graph attention network for considering the varying interactions amongst agents toward developing a socially aware and consistent trajectory prediction. Attention-GCN [130] was developed by applying an attention mechanism for considering the mutual influence between close pedestrians. The basic idea in this model is that close pedestrians have more influence on each others' decisions. Zhou et al. [131] employed a double-attention mechanism; the first one aiming at capturing the spatial interactions among all agents—and the second one for considering the temporal movement patterns of each agent in the past. Monti et al. [132] applied a double attention-based GNN to consider each agent's future goals, as well as the interaction among different agents. Similarly, Li et al. [133] employed a double-attention mechanism on a dynamic spatial-temporal GNN to consider the historical and future features obtained from the state, relation, and context information. State here refers to the position, velocity, and heading information of the agents, relation refers to the relative information between each pair of agents, and contextual information is extracted from local occupancy density maps and local velocity field for each agent.

In addition to different deep learning tools and modules, different graph types have been employed to improve the performance of the developed frameworks. Kumar et al. [134] constructed a hypergraph in which nodes are composed of traffic actors and traffic elements (stop signs and traffic lights) in the scene. Jo et al. [135] employed a hierarchical GNN to consider the impacts of unobserved maneuvers in multi-agent trajectory prediction. Lu et al. [136] utilized the concepts of dynamic and heterogeneous graphs to capture the varying road conditions and interactions among vehicles. Tang et al. [137] aimed at studying the temporal dependencies at different time scales. To this end, they developed a multi-scale spatial-temporal GNN that utilizes stacked layers of temporal convolution networks and graph convolution networks, followed by an LSTM-based encoder-decoder for trajectory generation.

Despite the ubiquitous application of GNNs in the motion prediction of AVs, constructing the graph of agents in the scene is still an open research area. The first challenge is to identify the agents and objects that have a significant influence on the ego vehicle or interacts with each other resulting in an influence on the ego vehicle. Especially in an urban area, with multiple agents (vehicles, cyclists, and pedestrians), identifying the nodes is very demanding. Moreover, unlike many other transportation applications where the number of nodes is usually fixed through time, the influencing agents vary with time, and therefore, we are usually facing a dynamic graph. In this regard, utilizing dynamic, heterogeneous, and multi-dimensional graphs could improve the performance of GNN-based motion prediction models, but their application has been very limited in this field.

Another challenging task in motion prediction using GNNs is identifying and weighing the interactions among vehicles. In many of the current endeavors, interactions are weighted based on their distance from the ego vehicles and this distance is independent of the location of surrounding objects and the speed of the ego vehicle. However, in real-world scenarios, the ego vehicle is usually more influenced by its direct leader(s) in the same lane and less influenced by the preceding vehicles. Moreover, these interactions greatly depend on the speeds of the vehicles and the road structure and properties. These are the factors that should be considered in future studies for improving the accuracy of motion prediction tasks using GNNs. For instance, reinforcement learning can be utilized for learning the edges and weighting them in such complex environments.

Finally, GNNs can only capture the correlations among agents. Therefore, they are not appropriate in their current form to capture causal relationships or for causal learning. However, in urban areas, due to the complexity and diversity of different scenarios, it is really hard to train the model for every possible scenario, and therefore, using a model that is trained solely based on observed correlations might not be safe and accurate in new unobserved scenarios even in the same environment. Accordingly, it is important to investigate the possibility of causal learning in GNN applications, which has been mostly neglected in current research.

3) *Motion Planning*: Another component in the operation of autonomous vehicles is motion planning. Motion planning is responsible for the safe and smooth maneuvers of the ego vehicle while avoiding the static and dynamic obstacles and agents in the scene. owing to its importance, numerous studies have focused on this level of autonomous vehicles' operation, and a wide range of tools and methods have been applied by researchers, from model predictive control (MPC) to deep learning-based algorithms, and end-to-end frameworks [138], [139]. Although GNNs have been recently used in the motion planning of in-door robots and unmanned aerial vehicles (UAVs) and have shown superior performance [140], [141], [142], their application in the motion planning of autonomous vehicles in the real-world situations has been yet relatively limited. In the domain of the autonomous vehicle, Hogle et al. [143] proposed Graph-Q for the control of autonomous vehicles in urban and multi-agent scenarios by considering the interactions among different vehicles in the scene in form of a graph. For constructing the graph, they followed two approaches: the first approach, called close-agent connection, only connects each ego-vehicle to its leader in the same, left, and right lanes. The second approach, which is called all-close connection, connects all close vehicles (to the ego vehicle) to their followers in the same, left, and right lanes. Edge weights in the two scenarios are calculated based on the direct distance between pairs of vehicles. They also applied the Deep Scene-set algorithm [143] to extend the application of Graph-Q to multiple input types and sizes (such as vehicles, lane markings, and signs).

Hart and Knoll [144] utilized graph neural networks in the actor-critic (AC) reinforcement learning method to take the advantage of GNNs in unraveling the interaction among vehicles. In order to evaluate the performance of GNNs, they conducted the same experiments with conventional deep neural networks and concluded that GNNs can handle varying numbers of vehicles in different scenarios and improve the generalizability of the model.

Jin and Han [145] utilized relation learning on graphs to identify ghost objects (false positive detected objects). Their idea is that in a normal driving scene, all vehicles are affected by their neighbors, so the behavior of real vehicles is more or less logical, while the behavior of ghost vehicles is not.

Chen et al. [86] proposed a deep reinforcement learning-based model by integrating graph convolutional networks and deep Q network (GCQ) to enable multiple AVs in a scene to collaboratively make lane-changing decisions. The graphical structure of the AV network comprises two layers: 1. a local network that is a star graph, including the ego vehicle and its surrounding human-driven vehicles; 2. and a global network, in which the nodes are all AVs on the road. The AV gathers information from both human-driven and autonomous vehicles and sends the information only to autonomous vehicles in the network. The proposed framework claimed to be able to address the dynamic-number agent problem (DNAP), fuse the multi-source information from cooperative sensing, do safe, efficient, collaborative lane changes, and be robust against traffic density changes.

Cai et al. proposed DiGNet [146] and DQ-GAT [147] for scalable self-driving policy learning, where a graph attention-based network is used to process the heterogeneous traffic information. The idea here is to enable the autonomous vehicle to learn the driving task for generic driving scenarios by unraveling the interactions among agents instead of being trained for a specific scenario. To this end, they designed a two-layer graph attention network, in which nodes include the state features of the agents in the scene, and their features are updated through the self-learning attention mechanism. The difference between DiGNet and DQ-GAT is that in the former, a supervised method is used for controlling the autonomous vehicle, while in the latter, the derived features vector is processed within two separate MLPs to generate the advantage functions and the state value in D3QN algorithm [148], and an RL method is utilized to control the vehicle. They argue that their model is generalizable to unseen traffic conditions and have conducted experiments in a wide variety of seen and unseen scenarios.

Finally, Klimke et al. [149] utilized GNNs for developing a cooperative motion planning of multiple vehicles at urban intersections. By using the graph representation of the vehicles, they were able to deal with the dynamic number of vehicles in the scene. Comparing their method with a first-in-first-out method and traffic governed by static priority rules, they reported significant improvement in outflow and the number of stops at the intersection.

Reviewing the studies on utilizing GNNs in motion planning of AVs, arguably the most important open research area is developing generalizable driving policies utilizing GNN-based frameworks (usually coupled with RL or imitation learning). Only a few studies have focused on and tried to address the generalizability and transferability of motion planning algorithms considering the unique nature and different characteristics of transportation networks.

Moreover, coupling GNNs with RL could be promising in motion planning applications. Most existing multi-agent reinforcement learning studies focus on a single AV or a fixed number of AVs. The versatility of GNNs in capturing the interactions among a dynamic number of objects, together with the ability of RL to achieve optimal control strategies without the need for expert demonstration, is a great opportunity to overcome major challenges in multi-agent motion planning of AVs in complex urban areas.

Finally, most of the current studies in the literature for motion planning focus on safety-relevant effects. Therefore, the impacts of learning-based motion planning algorithms on the efficiency of traffic networks have not extensively been studied [150]. For instance, few studies have evaluated the performance of motion planning algorithms on string stability and congestion mitigation in urban or motorway scenarios. GNNs, due to their ability in considering the interactions among vehicles, can greatly benefit this research area through cooperative and joint planning for a group of AVs.

Table IV summarizes the main findings from the studies on motion prediction and planning of autonomous vehicles utilizing GNNs. The extracted information includes the graph type and its adjacency matrix, the experimental datasets and

baseline models being used in each study, the types of nodes in the graph, and spatial-temporal modules of the GNN framework.

D. Intersection Management

Efficient management of intersections is crucial to alleviating traffic congestion and improving safety. Traditional approaches in intersection management used fixed timing plans based on historical demand data. With the advent of modern data collections tools, adaptive traffic signal control methods emerged utilizing various approaches emerged, from metaheuristics algorithms to mixed-integer programming to computational intelligence and machine learning algorithms [151], [152], [153]. However, the presence of different road users and the complex interactions between adjacent intersections still put forth serious challenges when dealing with real-world networks of intersections. Recently, reinforcement learning-based approaches for traffic signal control and intersection management have been increasingly popular [153], [154] because they are able to learn directly from the observed environment without making explicit and unrealistic assumptions with regard to traffic conditions and environmental factors [154].

Many recent studies on reinforcement learning for traffic signal control, use neural networks or convolutional neural networks to extract features from the network; however, such vector representations of traffic networks cannot guarantee the extraction of geometric features of the road network because the interactions between intersections are not necessarily extended in Euclidean space. Therefore, some researchers have started adopting graph-based neural networks for modeling and managing intersections. Nishi et al. [155] were among the first who proposed a reinforcement learning-based approach using graph neural networks for addressing such problems in a multiple intersection network. Their model uses the GCNN method proposed by [156] to automatically extract geometric network features. Multiple stacked layers are used to extract features in a graph based on distant vertices rather than only using one layer to extract all features at once. Thereafter, a reinforcement learning algorithm learns the policy for managing the intersection. They evaluated their method on a six-intersection network and reported that their model was able to reach almost the same policy with 50% less run time. Also, their model was able to deal with more dynamic traffic demands.

Wei et al. [157] proposed CoLight, utilizing graph attentional networks for network-level cooperation of traffic lights. In order to solve the conflicts in learning the influences of neighbors on the target intersections, they utilized an index-free model learning with parameter sharing, in which they take the average over influences of all neighboring intersections with the learned attention weights, instead of using fixed indexing for the neighbors. Moreover, they evaluated their RL model for signal coordination for the first time on a real-world large-scale network (including 196 intersections) and compared the performance of their model with fixed-time [158], Max Pressure [159],

CGRL (an RL-based method) [160], individual RL [161], oneModel [162], Neighbor RL [163], and GCN [155], and concluded that their model outperforms all baselines with regard to average travel time.

Li et al. [164] developed a deep imitation learning framework based on graph neural networks for traffic signal control of multiple intersections. The input data is divided into two classes: state data indicating the traffic state variables, and strategy data that include the control strategy based on specific state data. GCN is used for unraveling the spatial-temporal traffic demand features. The whole road network is transformed into an undirected graph with intersections as nodes and roads as edges of the graph. They also use a masking method to let the model know about missing data and sensor working state; i.e., for each sensor, the data dimension is doubled to incorporate the working status of the sensor (as a binary status). Their structure consists of several LSTM models for each intersection (handling variable-length sequences and extracting temporal features), followed by a GCN, using the outputs of LSTMs to link the intersections with each other and to model the network as a whole. In the end, the GCN is followed by several deep neural networks (each for one intersection) to generate unique traffic signal control plans for individual intersections. They evaluated the performance of their model using simulation with real-world data and reported about a 7% reduction in waiting time, time loss of vehicles, and throughput.

Hu et al. [165] proposed GPLight, a deep reinforcement learning framework for dynamically controlling the phase and duration of traffic lights at intersections. This framework utilizes GNNs for short-term prediction of traffic states in multiple intersections and employs the predicted traffic states combined with current traffic conditions for controlling the intersection traffic lights. For this purpose, they used a weighted undirected graph to represent the network of intersections. The traffic flow prediction module consists of two spatial-temporal convolution blocks, followed by a fully-connected neural network. The convolution block themselves are composed of a spatial graph convolution layer surrounded by two temporal gated convolution layers. Finally, a deep Q-Network, combining Q-learning and deep neural networks, is used to control the traffic light at the intersection. Comparing the results of their algorithm with baselines (fixed time, Max Pressure [159], CoLight [157], and PassLight [166]), they concluded that GPLight is able to increase the throughput and reduce the delay at the studied intersections.

Yang et al. [167] identified three major shortcomings in multi-agent deep reinforcement learning algorithms for intersection control: transferring learned policies to diverse traffic networks, dynamically tackling the time-varying number of vehicles in the network, and capturing heterogeneous features of objects in the network. To overcome these drawbacks, they proposed IHG-MA, an Inductive Heterogeneous Graph Multi-agent Actor-critic algorithm, for multi-intersection traffic signal control. They defined the traffic network as a heterogeneous graph, in which there are different types of nodes and links. Node types in this

graph include the type of objects that could be the traffic signal controller (TSCer), intersection, lane, and vehicle. Accordingly, there are four types of relationships, namely, TSCer-control-intersection, lane-connect-intersection, vehicle-traverse-intersection, and vehicle-traverse-lane. The proposed algorithm conducts representation learning using a proposed IHG algorithm, and policy learning using a proposed MA framework. The aim is to design an algorithm to encode heterogeneous features for each TSCer and its neighbors and then learn the corresponding embeddings, compute the Q-value and the corresponding policy for each SDRL agent, and finally optimize the whole algorithm via the Q-value and policy loss using a decentralized multi-agent framework. They evaluated their proposed algorithm based on both synthetic and real-world datasets and compared with baselines (Max-pressure control [168], CoLight [157], MetaLight [169], MA2C [162], and IG-RL [170]) and concluded that their algorithm outperforms the state-of-the-art algorithms in terms of average intersection delay, average queue length and average travel time.

Zhong et al. [171] argued that recent studies using RL for coordinating traffic signals have drawbacks as they either design the state of the agents heuristically or model the traffic states deterministically. To address these issues, they proposed TSC-GNN (Traffic Signal Control via Probabilistic Graph Neural Networks), which aims to consider traffic uncertainties while learning the latent representations of agents and calculating the Q-value. They achieved this goal by variationally clustering latent representations of adjacent intersections with attention coefficients. This mechanism enables Bayesian inference in their proposed algorithm. TSC-GNN is comprised of three main parts: a graph attention module for identifying the importance of intersections inter-correlations (the same as [157]), a variational graph inference module for learning latent representation of intersections (instead of assuming a deterministic representation), and a module for predicting Q-value. Finally, they compared the framework with baselines (fixed time [158], Max Pressure [159], CGRL [160], Individual RL [161], OneModel [162], Neighbor RL [163], GCN [155], and Co-Light [157]) on two real datasets, and concluded that their model outperforms the state-of-the-art.

Yoon et al. [172] argued that previous RL methods were unable to adapt to unseen and unexperienced conditions, and therefore, proposed a transferable control policy based on RL and GNNs. To this end, they represented the traffic states as graphs and then trained them by utilizing GNNs. Learning the relationships between features across intersections enabled them to transfer partially-trained policies to inexperienced situations. To validate their argument, they conducted a test and compared the performance of their model with a conventional DQN model for scenarios where training data had not been available. The results indicated that the GNN model performed significantly better than the conventional DQN model on inexperienced data and covered a wider region of the search space.

Deep graph Q-network (DGQN) [173] was proposed to alleviate the limitations of value-based RL methods for

TABLE IV
AN OVERVIEW OF MOST CITED STUDIES UTILIZING GNNs FOR AUTONOMOUS VEHICLES, WHICH HAVE BEEN USED AS BENCHMARKS IN OTHER STUDIES

Study	AV Study Type	Graph Type	Adjacency Matrix	Experiment Datasets	Nodes Type	Temporal Module	Spatial Module	Compared Models
[130]	Motion Prediction	Undirected Static	Weighted Attention matrix	USyd Pedestrian Dataset, ETH, UCY	Pedestrians	Time-Extrapolator CNN	GCN	Social GAN, SoPhie, Next, STGCN, Social-BiGAT, Social_STGCNN
[134]	Motion Prediction	Hybrid Static-Dynamic	Directed Matrix of all actors in the defined distance	NuScenes, Collected dataset	Traffic actors, Traffic elements	Recurrent Module	GNN	CSP, MATE, RasterNet, SpAGNN
[135]	Motion Prediction	Undirected Static	Distance Matrix	NGSIM, Collected dataset	Vehicles	LSTM + MLP	GCN	Constant velocity, IMM-KF, V-LSTM, CS-LSTM, GRIP, GRIP++
[136]	Motion Prediction	Heterogeneous Context-Aware	Heterogeneous Aware Matrix	Context-INTERACTION, NGSIM	Vehicles	Temporal Attention + dilated temporal convolution	Graph Convolution	Arima, CS-LSTM, CNN-LSTM, VectorNet, Social-WaGAT, EvolveGraph
[137]	Motion Prediction	Regular	Euclidean-distance Matrix, Collision-avoidance Matrix, and Self-adaptive adjacency matrix	NGSIM, highD, ApolloScape	Vehicles	Dilated temporal convolution	Graph Convolution	CS-LSTM, GRIP++, PIP, TrafficPredict., StarNet.
[145]	Motion Planning	Regular	Neighborhood matrix based on defined distance	Collected dataset	Vehicles	multi-head GAT + VAE + MLP	LSTM-autoencoder + CNN + GAT	CIL-Net
[146]	Motion Planning	Regular	Weighted Attention Matrix	Collected dataset	Agents including the ego-vehicle	-	VAE + Multi-head attention graph	CILRS, MLP, GCN, GAT
[147]	Motion Planning	Undirected	Weighted Attention Matrix	OpenDD	Agents including the ego-vehicle	-	GAT	IL-Conser, IL-Aggr, H-REIL, FSM-TTC
[149]	Motion Planning	Directed	-	Highway-env	Vehicles	-	GCN	-

application in large-scale networks with a high number of traffic signals. In specific, the authors developed a graph-based Q-network to efficiently capture the spatial-temporal dependencies in a large network. In addition, they devised a parametrized adjacency matrix to take the effects of congestion propagation into account. With this framework, they could outperform both state-of-the-art RL algorithms and fixed-signal operation.

Lastly, Wang et al. [174] developed MetaSTGAT, a spatial-temporal graph attention neural network for considering the spatial-temporal correlations among intersections and implementing an adaptive traffic signal control. They also utilized a meta-learning method for GNN with generation to adapt the designed framework to dynamic traffic flow.

Despite the growing number of research in utilizing GNNs in multi-intersection control and management, the applicability of such methods for real-time signal control is under question. Most of the current studies do not report their run time. This is especially important for intersections because a short delay in processing the information could result in significant delays or even unsafe situations. Also, the robustness of such frameworks against missing or noisy input data (due to detector failure) has not yet been investigated in the literature. Intersections are commonly important urban bottlenecks and any disruption in their operation may result in gridlock.

Moreover, the generalizability of the developed frameworks should be explored in irregular traffic conditions, such as holidays or during peak/off-peak hours. Finally, many of the developed frameworks would fail to converge in real-world large-scale networks [173] with hundreds of intersections. Applying multi-agent RL and incorporating an attention mechanism for identifying the most influential intersections may improve the scalability of GNN-based frameworks in traffic signal control problems.

E. Parking Management

A significant portion of traffic congestion, especially in urban areas, is caused by cars looking for parking places. Therefore, appropriate management of parking spaces could alleviate a great portion of traffic congestion. Different strategies could be adopted for properly managing parking areas, from accurately predicting the parking demand to the development of new parking spaces. In this subsection, we review the studies on parking management that utilized GNNs as part of their methodology.

Yang et al. [175] leveraged a GNN-based framework for block-level parking occupancy prediction. They leveraged GCNs to extract the spatial relationships of traffic flow in large-scale networks. Their framework consists of three main modules: GCN, LSTM, and decoders. Input features (parking meter transactions, traffic conditions, and weather conditions) are fed into GCN, followed by LSTM to explore the temporal correlations. Finally, the output is distributed over city blocks using a decoder layer. They compared the performance of their model with baselines, including the latest observation, historical average, support vector regression, Kalman filter, MSTARMA [176], LASSO, and LSTM, and showed that their framework outperforms all baselines by a significant margin.

Zhang et al. [177] proposed SHARE (semi-supervised hierarchical recurrent graph neural network) for addressing the missing data issue in parking availability. They utilized a hierarchical graph convolution module and a recurrent neural network model to capture the spatial and temporal dependencies, respectively. The graph convolution module itself is comprised of a contextual graph convolution block for capturing local spatial dependencies, and a soft clustering graph convolution for modeling the global autocorrelations. They used an approximation module for estimating missing parking availability data by fusing a propagating convolution block and a temporal module through

an entropy-based mechanism. Finally, comparing with the baselines (logistic regression, GBRT [52], GRU [178], Google-Parking [179], Du-Parking [180], STGCN [27], DCRNN [6], CxtGNN (SHARE without parking availability approximation), CAGNN (without soft clustering module), they concluded that SHARE is able to outperform previous modeling framework for parking availability prediction in 15, 30, and 45-minute intervals.

Similarly, Wu et al. [181] also aimed at recovering parking availability missing data. They proposed G-RGAN, the graph recurrent generative adversarial net, by embedding GCN and GRU into the generator and discriminator modules of Generative Adversarial Networks. The idea is to use the GCN for capturing the spatial correlation between the parking lots, and the GRU for modeling the temporal ones. They defined the structure of the generator and discriminator in an almost symmetry way to balance the ability of the model to learn and grow. In the training process, they first fed real data into the discriminator and got a Score of S . Then, the generator was fed with noise sequential data, and sequential data X was generated. This generated X data was fed into the discriminator, and a new score for the generated X was calculated as S_G . Next, the loss is calculated and the parameters are modified using gradient descent. Finally, they compared the performance of the model with some simpler variants of the framework, including GAN, RGAN, G-GAN, and G-RGAN; and concluded that the G-RGAN could outperform all other variants of their model.

Zhao et al. [182] developed a system for real-time city-wide parking availability prediction based on parking transaction data and contextual information. To this end, they integrated the inflow and duration prediction models in order to achieve the outflow information for different time slots. In this way, they could achieve a framework for real-time parking availability prediction. The inflow prediction model consists of LSTM modules followed by multiple graph convolutional networks for capturing physical and semantic similarities between nodes, which are parking blocks. The contextual information (such as weather) is also entered into the framework through LSTM modules to enable exploring the temporal correlations. They evaluated the performance of their model on a four-month real-world dataset against state-of-the-art baselines. The baselines include HA, ARIMA [183], SVR [184], LASSO, backpropagation neural network (BPNN), stacked autoencoder (SAE) [185], GRU, LSTM, Du-Parking [180], and a model based on their own framework but only using a single convolutional neural network for considering the physical adjacency of parking blocks (and ignoring the semantic correlations between parking blocks). They argue that their model is capable of outperforming the state-of-the-art by 43% in terms of relative error.

Xiao et al. [186] proposed a hybrid spatial-temporal graph convolutional network (HST-GCN) for on-street parking availability prediction. The hybrid refers to integrating an attention mechanism called distAtt into their modeling framework for capturing long-term spatial correlation in conjunction with a spatial-temporal convolution block for

capturing instantaneous spatial-temporal correlations. They compared the performance of their model with linear and deep learning models, including HA, ARIMA, LSTM, DCRNN [6], STGCN [27], and ASTGCN [32], and their results showed that their proposed framework could perform the best in all evaluation metrics (MAPE, MAR, and RMSE).

After reviewing the studies on parking management and prediction, the lack of studies in this area is apparent. More advanced frameworks are needed by considering time-of-day and day-of-week variations for parking availability prediction. Also, all studies have only focused on one type of parking facility, while there are different types of parking lots in an urban area, from on-street parking spaces to public parking lots and commercial centers' parking areas. The ability of more flexible GNNs, such as heterogeneous and multi-dimensional graph neural networks, can empower decision-makers to handle the real-time management of multiple parking types at the same time. Also, the only situational variable that has been employed in previous studies is the weather conditions. However, land use patterns and commercial/non-commercial temporal activity patterns could be of even more importance in parking management and prediction studies.

F. Urban Planning

Urban planning studies cover a wide range of domains, from land-use modeling to urban development, and network resiliency. In this subsection, we conduct a review of the existing studies using GNNs to solve urban planning problems. These studies cover various areas, including urban knowledge discovery, transportation resiliency, roads attribute inference, and human activity pattern exploration.

Zhang et al. [187] built an urban knowledge graph in order to develop an end-to-end framework for large-scale urban studies. They employed convolutional graph neural networks to analyze the structured prior knowledge in urban areas for prediction and decision-making purposes. Their general framework, which is called UKG-NN, conducts an automatic feature extraction at three levels, namely global, propagation, and local level. Afterward, this information is fused and fed into a graph neural network. One of the main features of their framework is its relative interpretability based on propagation graph features. They applied their framework to two real-world studies, optimal store placement and traffic incident inference, and reported improved performance compared to traditional methods such as random forest, support vector classification Huff Gravity Model, and Geo-spotting.

Zhu et al. [188] utilized GCN for predicting the characteristics of geographical places in urban areas. The idea of using GNNs in this study is that the attributes of a place have correlations with the characteristics of the places to which it is connected. Therefore, they constructed a graph of different places, in which nodes represent the places themselves and attributes are the features of the nodes. Also, the edges represent the connections between different places. They utilized their framework to predict the attributes of some places based on their observed characteristics and contextual information.

Wang et al. [189] developed an end-to-end deep learning model utilizing diffusion graph neural networks for predicting spatial-temporal patterns of transportation resilience under extreme weather conditions in urban road networks. Traffic speed is considered the measure of network resiliency, thus the goal of the framework is to predict network-wide speed. The transportation road network is represented as a weighted directed graph with sensors as nodes of the graphs and the evaluation is done using urban big data, including traffic speed data, meteorological data, and weather forecasting data. The overall structure of the proposed model is very similar to [6]. Firstly, the urban data (meteorological, weather forecast, and traffic speed) are fed into a spatial-temporal graph as features of its node for different time horizons. Then, a diffusion convolutional recurrent mechanism is employed in both the encoder and decoder parts of the prediction model. Finally, the traffic speed is predicted as the graph signals in the last part. Based on the results, they concluded that aggregated data of precipitation events related to transportation systems could be used for modeling transportation resilience under extreme weather conditions even when facing a sample imbalance problem (for instance, due to a lack of historical disaster data). Moreover, to evaluate the performance of their model in the speed prediction task, they compared their speed prediction results with some baselines, including GCRNN, S2S-Att, Seq2Seq, Bi-LSTM, Bi-GRU, LSTM, and GRU, and argued that their framework could outperform competitors in terms of accuracy.

He et al. [190] utilized graph neural networks for road attribute inference (such as lane count and road type) from satellite images. The problem with using satellite imagery data is that, sometimes, a significant portion of a road might be occluded due to trees or buildings beside the road. GNN here is employed to capture the spatial correlation of features along the roads (e.g., assuming that the number of lanes remains the same in a specific link) with the aim to compensate for occlusion in satellite images. They compared the performance of their model with a CNN image classifier on two datasets and reported better accuracy in terms of the “number of lanes” and “road type” identified. Also, they concluded that their framework, called RoadTagger, is generalizable to city-scale graphs.

Hu et al. [191] developed a geo-semantic framework for exploring the relationship between traffic interaction and urban functions (i.e. commercial, public, and traffic roads). They first translated the data from taxi trajectories and transportation road segments into words and sentences (building a so-called Road-trajectory corpus) and then learned a geo-semantic embedding representation using Word2Vec with the aid of point-of-interests (POI) data. Finally, based on the extracted embedding features, a graph convolutional neural network is utilized to predict the social functions of road segments. They compared the performance of their model with linear regression, KNN, SVM, and random forest and reported significant improvements over the baseline. The novelty of their methodology is that they have incorporated intermediate GPS records of taxi trajectories, such as the movement flows

and traffic states, into their learning procedure instead of relying solely on drop-off and pick-up information. Secondly, They have used a GCNN to improve the classification accuracy based on the fact that moving vehicles in urban areas are restricted to the road network, and adjacent roads have interactions with each other.

Li et al. [192] employed graph neural networks in conjunction with recurrent neural networks to predict the intensity of human activities using mobile phone data across a country. They constructed three separate graphs for this purpose. The first graph is a distance graph in which the edges represent the geographical distance between the cell phone towers. The idea is based on Tobler’s first law of geography [193], stating that geographical distance affects the similarity of two different places. The second graph is based on the movement of people between spatial cells, and the third graph is based on the phone call interaction records (an indication of social interactions between the cells). Finally, using a graph transformation step and with the aid of graph convolutional networks, they were able to integrate the physical and social interactions between spatial units in the studied area to capture dynamic spatial interaction patterns and predict future activity intensity variation. They compared their method with several deep learning (e.g., ASTGCN, and LSTM), machine learning (KNN and GBDT), and time series (ARIMA) prediction models and concluded that their deep learning framework is able to outperform all baselines in terms of prediction error and stability.

In general, the ability of GNNS in taking into account the spatial and temporal dependencies among data points makes them very useful for urban planning and management studies. However, the application of GNNS in urban studies is relatively new and has been yet limited. Many relevant applications can be imagined for GNNS such as urban dynamics, which includes the development of cities and the dynamic flow of socio-economic activities; social segregation analysis, which concerns the differentiation of different urban areas from the social and demographic points of view; and urban sensing, which aims at identifying different land uses and social activities based on social media or volunteered user data [194].

Nevertheless, one specific limitation for many of the urban planning studies is the availability of high-fidelity data. Traditionally, many studies only relied on aggregated statistics, which had the issue of not being updated for long periods. Recently, new sources of data, such as social media and other user-generated data, are being used for many smart urban studies. Although this has the benefit of accessing an almost real-time collection of data, these data types have been shown to be a biased representation of the population. For instance, the users of social media are usually the younger generations with higher income levels [195]. Therefore, future research should take into account such partialities to prevent inaccurate estimation of urban dynamics.

Finally, the interpretability of GNNS, like many other deep learning methods, is still under question, especially compared to traditional linear and statistical spatial regression methods. As the aim of many urban planning studies is to aid

the policy-makers with their decisions, developing a black-box framework without explaining which factors correlate to/causes the changes might not be beneficial to practical applications.

G. Transportation Safety

The safety of transportation systems has always been one of the hottest research areas. Safety measures could be evaluated from various aspects, either psychological view or technical aspects. In any approach, identifying the areas with high risks of accidents and incidents could help us with identifying the hotspots for future improvements. Therefore, a great deal of research in the transportation domain has focused on locating the areas with higher safety risks or accident prediction, and a wide range of prediction tools has been employed for this purpose, from traditional statistical tools to data-driven methods [196], [197], [198]. In this section, we conduct a review of the existing studies concerned with the safety analysis of transportation networks utilizing graph neural networks.

Zhang et al. [199] developed a traffic risk forecasting framework utilizing two social media and remote sensing datasets. Traffic risk forecasting refers to the risk of occurring accidents in urban areas evaluated at a fine-grained level; for example, intersections or off-ramps. They introduced GraphCast, a multi-modal graph neural network framework, which consists of three main parts: 1) a GNN module for learning the dynamics of traffic accidents from social media data, 2) an attention mechanism for learning the spatial correlation of traffic accidents based on multiple GNN instances, and 3) an optimized learning process that jointly optimizes the parameters of the two networks for GNN and attention modules. The graph nodes in the framework are the cells for which there have been traffic accident reports. The weighted edges are defined based on the similarity of visual features obtained from remote sensing data. The Twitter and Google Map Satellite Imagery data were used for model training and the Vehicle accident report data as the ground truth. The baseline models used in this study include linear regression, Ridge regression [200], Gaussian Process, Bayesian Automatic Relevance Determination [201], Multilayer Perceptron, W&D [202], LSTM, and GRU.

In order to improve the spatial-temporal granularity of accident predictions, Zhou et al. [203] proposed a differential time-varying graph convolution network to capture the dependencies within traffic variations at multiple spatial scales and different temporal steps. They divided the study area into q regions and m sub-regions and then constructed the undirected urban graph based on these sub-regions. They also defined two types of features for the nodes of the graph: 1) static features that consist of the characteristics of the transportation road network in each sub-region, and 2) the dynamic traffic features, which are defined as the weighted summation of the number of accidents and their severity in each sub-region. The final aim of the model is to predict the dynamic features at all regions and sub-regions for r time steps in

the future. In order to account for non-euclidean relationships between sub-regions, the authors use a differential time-varying graph neural network (DT-GCN) encoder, including an affinity matrix. The affinity between two sub-regions is defined based on static road network features, dynamic traffic states, and the transition of traffic flows between sub-regions. The term “differential” in DT-GCN refers to the idea of feeding *differential* images into the GCN based on the fact that the occurrence of accidents is more related to variations in traffic states. Finally, They employed an LSTM encoder and a context-guided LSTM for encoding and decoding features in the short and long term. The purpose of using CG-LSTM is to incorporate contextual factors like weather information to enhance the spatial representations in the hierarchical LSTMs. The performance of their framework is evaluated using two real datasets and baseline models including ARIMA, LSTM, Hetero-ConvLSTM [204], STGCN [27], STG2seq [51], STSGCN [205], STDN [206], DFN [207], MTPSO [208].

Chandar et al. [209] employed graph neural networks for capturing complex and nonlinear inter-relations in high dimensional feature space in order to predict the safety index for a road. They used the alerts recorded by buses on the studied road to define a measure of proneness to accidents. Moreover, they included other features related to time, location, and weather. They followed a batch processing schema in which each batch consists of 20 graphs (one graph per bus), and nodes of the graph are individual alert events for a specific bus on the road on a specific day. The edges of the graph handle the order of the events, thus considering a time-series nature for the events in each graph. The labels for each node in the graph are defined based on the accident proneness on a scale of 1-5. The novelty of their proposed structure is that it uses a sequence of trainable graphs. Comparing their method with baselines, including logistic regression, feed-forward neural network, and LSTM, they could achieve a performance competency of more than 50%.

DSTGCN (deep spatial-temporal graph neural network) was proposed by Yu et al. [210] to predict the risk of a traffic accident at the level of road segments. They collected a wide range of features, including weather conditions, traffic flows, road structures, Point Of Interest (POI) distributions, and traffic accident records. Thereafter, a ST-GNN is used to unravel the dependencies of mechanisms that cause traffic accidents. They reported superior performance compared to traditional methods, such as linear regression and SVM, as well as state-of-the-art deep learning algorithms.

Huang et al. [211] proposed a gated graph convolutional multi-task (GGCMT) framework for city-wide traffic accident prediction. They divided the study area into squares of the same size and constructed a weighted graph of these virtual regions. The risk factor for each region is calculated based on the number and severity of accidents in that area. The weights over the links in the graph are defined based on the similarity of the risk factors between the regions. Finally, a gated graph convolutional neural network is utilized to predict the accident risk factor for multiple time steps in the future.

Despite the great potential of GNNs for transportation and traffic safety studies, their application in this domain has been relatively limited. One important characteristic of GNNs that makes them suitable for safety investigations is their ability in exploring correlations in non-euclidean space. Almost all previous statistical and machine learning methods in safety analysis would use aggregated data only over *neighboring* regions or links. However, due to the complexity of land uses in urban areas, this aggregation and assumption for correlation exploration are not realistic. For instance, the type of accidents at the entrance gates of cities are usually similar, while they are the farthest points within a city. Similarly, the type and severity of accidents in an intersection could be totally different from its adjacent links or roundabouts. GNNs provide a great opportunity for traffic safety analysts to consider different accident types, their severity, the specific locations, their connection with external factors such as traffic states and weather conditions, and their correlations within an urban area by going beyond distance-based measures.

VI. DISCUSSION AND OPEN RESEARCH AREAS

Although a great deal of research in recent years has focused on developing and applying machine learning and deep learning on graph-type data, there are still areas that need more investigation. In this section, we briefly discuss several challenges as well as open issues that are worth considering for future studies based on the current review of the literature and the state-of-the-art in graph neural networks.

A. Graph Construction in Graph Neural Networks

In general, GNNs begin by generating or constructing graphs as their first step. The process of producing a graph concerns defining the nodes and the edges between those nodes. Although many previous studies and surveys have neglected the importance of problem-specific graph generation, the appropriate construction of a graph highly depends on the type of problem. In this sub-section, we briefly overview different options for defining the graphs from two main perspectives: nodes, and adjacency matrices.

1) *Nodes*: In many traditional traffic forecasting problems, defining the nodes in the graph was considered the most obvious step in graph construction. For instance, many studies would consider the location of the detectors as the nodes of the graph. This was the same for many public transportation and passenger flow prediction studies, where the locations of stops were considered as nodes of the graphs. However, some researchers utilized novel approaches for defining the nodes in their studies and achieved superior performance. For instance, Ma et al. [212] tried to identify the locations of bus stops and intersections in a city by using a density-based clustering algorithm instead of relying on labeled stops.

The introduction of GNNs to other areas than traffic forecasting complicates the problem even more. For instance, defining the nodes in AVs is really important and challenging. Identifying the agents that interact with each other and have an influence on the driving behavior of the ego vehicle is not a straightforward task. Many researchers consider the

agents/vehicles within a certain distance from the ego vehicle as the nodes of the graph, but assuming a fixed distance is not consistent with reality because the interaction among vehicles highly depends on their speed and the complexity of the environment (e.g., urban areas or motorways). Moreover, the number of nodes in these graphs is not static, and traditional GNN algorithms might not be efficient for tackling such graphs.

As another example, in ride-hailing and shared-mobility systems with no explicit stations, identifying the nodes of the graph is a non-trivial task that can have significant impacts on the final results. The development of public transport systems and other urban infrastructure greatly depends on the partitioning of cities. Therefore, exploring the most suitable zoning strategies, and accordingly, building or learning the most meaningful graph structure in terms of vertices is of great importance. In summary, the authors would like to emphasize the importance of node definition in graphs in GNNs, which they believe has been overlooked in many transportation-related studies concerning GNNs.

2) *Adjacency Matrix*: As mentioned previously, the connectivity between nodes is reflected in the network's adjacency matrix. Defining a suitable adjacency matrix could be even more complicated than defining the nodes in a graph. Adjacency matrices can have simple or complex, multi-dimensional definitions depending on the problem context and the network in which they are defined. As an example, studies in the traffic forecasting domain have usually employed the simple adjacency matrix that is derived from the real-world transportation network properties, which most of the time is distance or connectivity [4], [30], [32]. However, some other studies in traffic forecasting have tried to learn the adjacency matrix instead of pre-defining it [37], [64], [65]. Typically, this approach uses the main and basic network data, such as the distance matrix, as the initial adjacency matrix, and then updates this matrix during the training process. This is due to the fact that these studies take into account that the spatial relationships between roadways are likely to change over time, and a fixed adjacency matrix is not capable of accurately reflecting those spatial relationships.

Contrary to traffic forecasting studies, which usually consider a single adjacency matrix, many demand prediction studies consider a number of different aspects with regard to the relationship between the nodes. In some studies, only one aspect is taken into consideration when making the adjacency matrix [75], [82], [93], while in others, different aspects are combined in a weighted manner [61], [85], [93], [212]. Some others also use the multi-graph concept instead of combining the multiple adjacency matrices into one matrix [49], [63], [73], [113]. In this subsection, we provide a list of different approaches for defining adjacency matrices in various studies:

- *Distance*: One way to encode the connection between stations is simply through the spatial distance. There are several ways to construct a distance matrix, including using spatial distances [61] or spherical distances [82] using known latitudes and longitudes, network-based distances [86], and even network-based travel time [72], [89].

- *Demand*: This matrix is more commonly used in demand or flow-based studies [49], [93]. A historical urban traffic record between two or more nodes can provide considerable information for constructing an adjacency matrix. A demand matrix, also known as an interaction matrix [73], is a measure of the flow/interaction between two nodes based on records of their demands/flows.
- *Connectivity*: spatial-temporal predictions require consideration of the transportation system as well. In theory, regions that are geographically distant but are easily accessible may be correlated with one another. There are several types of connectivity, which are either caused by roads such as motorways and highways, or by public transportation, such as subways. A connectivity matrix, also called “physical matrix” [85], is a kind of adjacency matrix that has been used in several studies to demonstrate such associations between nodes [86], [112], [113].
- *Neighborhood*: This kind of adjacency matrix is suitable for graphs whose nodes have been defined as zones or grids [50], [63]. In this situation, adjacent zones often interact with each other, and a connection between them should be considered. As an example, it is likely that the increase in traffic in one region will have effects on its neighboring regions as well. The neighborhood matrix has been designed to take into account this type of relationship in the network.
- *Functional Similarity*: In making predictions about a node’s value, it is intuitive to refer to other nodes that have functionally similar characteristics to the target node or that are in the same functional zone. There have been some studies suggesting that the “functionality” matrix, or the “social functionality” matrix, could be defined using a combination of density and categories from POI data [50], [212], or even by using information about land use for the nodes [63]. As a result, the similarity between the two related nodes in each cell is taken into account in the calculation of each cell in this matrix.
- *Correlation*: It is also possible to construct the edges of a graph based on the correlation among the features of the nodes. In this regard, a correlation matrix, which has been given different names in different studies including “mobility pattern”, “demand correlation”, and “flow correlation”, is defined based on the assumption that nodes with similar patterns of one value can correlate with each other [63], [85], [86], [89]. For example, demand for two bus stops located far apart, but both near large shopping centers, can be correlated, and a rise in demand at one bus stop could mean an increase in demand at another bus stop.
- *Geographic*: This matrix has been also referred to as network structure correlation matrix [89]. Analysis of a traffic or transportation network can consider several geographic factors so that the target value patterns of two nodes with the same characteristic are likely to be the same. Depending on the concept of the network, these features may differ; for example, for a traffic

network, they may be the length, width, and number of intersections of the road. Public transportation systems can also have several geographical features, including degree and distance, opening dates, and distance to the city center of stops/stations [89], [212].

- *Operational Information Correlation*: This notion is more applicable in the transit networks like metro or bus networks. The theory is based on the assumption that stations/stops with similar operational patterns might also be correlated. Specific information regarding line headways during peak hours and off-peak hours, or fleet capacity, are considered part of each public transportation’s operational characteristics to make this kind of adjacency matrix [212].

To conclude, one of the most important steps in many studies is the specification of the type of adjacency matrix that will result in more accurate and reliable predictions. Additionally, it is important to note that there is no guarantee that a model that considers all the matrices outlined above will perform well, so the most optimal model should be determined by checking all the combinations of the matrices in different ways. Furthermore, there are dynamic adjacency matrices, which can be adjusted based on the weather, holidays, or other variables, and they can be applied depending on the network of interest. A further point to consider in this regard is the use of graph neural networks for the purpose of link prediction as a means of constructing adjacency matrices in the presence of insufficient data about the network under investigation. A final point that can assist in making an accurate graph is the concept of a hypergraph. The hypergraph represents non-pairwise relationships between vertices by utilizing hyperedges. This means that hypergraphs can be used to represent the inherent relationship between data of higher-order [90]. As an example, hypergraphs could be useful in a metro or bus network, in which many stops share the same route or line, or when several bus stops are located in a city’s downtown area where the demand for them is all correlated.

B. Loss Function Design and Type of Learning

After reviewing the current studies utilizing graph neural networks, it appears that most of the current studies have focused on node-level learning, aiming to predict the feature over the nodes of a graph. However, as discussed throughout the paper, other interesting–yet overlooked–learning tasks could also benefit intelligent transportation systems. For instance, edge-level learning on graphs could be utilized for predicting values over edges or to explore the relationships between nodes. As an example for the former application, in the example of an AV in a multi-agent scene, assuming that we are considering a set of agents around the ego vehicle as the candidate inter-related agents (nodes of the graph), the edge-level learning task could aim for predicting the presence of any interactions between the ego vehicle with other agents, and also between the agents themselves. Also, as an example of the former application of edge-level learning, one might be interested in predicting travel times along the links or the strength of relationships between data points in a

transportation network. Utilizing edge-level tasks in predicting link features (like travel time) is more intuitive compared to the current methods that assume the links as the nodes of the graph and ignore the presence/properties of intersections as one of the main causes of bottlenecks in urban transportation networks.

Similarly, graph-level tasks have been mainly neglected in previous studies. This class of tasks allows us to predict a value for the whole graph based on the features of its nodes and edges. For instance, a safety index could be predicted for an intersection based on the interactions of the vehicles within the intersection in real-time. As another example, a congestion level index can be calculated based on the speed/number of vehicles detected at data collection sites. These indices could be also computed for sub-regions of a city to facilitate the quick analysis of different regions of the city.

C. Extending the Applications of GNNs

Although GNNs have been used in various transportation problems, from traffic forecasting to travel demand modeling and autonomous vehicle operations, there is still great potential to apply graph neural networks to new problems. Multi-modal transportation demand modeling is one of these areas where GNNs can make significant contributions. In a multi-modal network, not only are there correlations between data points of one travel mode but also strong correlations could be observed between data points of different modes of travel. For instance, adjacent bus and metro stations are expected to experience similar patterns of demand, especially in peak hours, although not being connected to each other. Also, today, with the development of multi-modal transportation systems, a significant portion of the demand is usually shared between two or more modes of travel. Passengers using shared scooters might use the bus network to fulfill their trip, or travelers may use the bus and metro within the same trip. Separating the demand for different modes of travel, which is a common practice in travel demand modeling, makes it impossible to consider these micro-interactions into account. Moreover, intermodal shifts during adverse weather conditions or other abrupt and unexpected situations can be of great importance. For example, during rainy days, a significant deal of public transportation demand might shift towards ride-hailing systems or private cars. Therefore, considering the inter-correlations and interactions between these multi-modal points is really important in prediction tasks. Multi-dimensional and heterogeneous graphs have great potential for dealing with such problems.

Multi-scale and multi-level prediction tasks can also be considered in future modeling endeavors. Decision-makers are usually interested in national indices, and at the same time, some indices at finer levels for important cities. In these situations, the interactions between cities could be defined as one scale, and the interaction within the intended cities could be modeled at another scale using such types of graphs.

Last but not least, hyper-graphs and dynamic graphs seem to be the next game changers in the applications of GNNs

in transportation. Although a few studies have employed such graphs, these types of graphs, due to their great flexibility in defining and connecting vertices, could be used in many transportation problems. One example could be utilizing dynamic graphs for considering time-varying correlations or when the number of active objects (nodes) changes over time. Also, using hyper-graphs allows considering multiple correlations among more than two nodes in a graph. This is especially attractive in modeling public transportation routes or a series of intersections in an urban corridor. All in all, more complex graph structures might be utilized for addressing more complex transportation problems, and researchers are encouraged to go beyond using the generic GNNs, which have been the most popular types of graphs in intelligent transportation systems studies.

D. Integrating New Paradigms Into GNNs

As also indicated in previous studies [8], [20], most of the current studies on GNNs have utilized spatial-temporal graph neural networks. Although a few studies have employed novel mechanisms like attention modules and adversarial networks, there is still a great opportunity to go beyond ST-GNN, especially for real-time, large-scale, and multi-step prediction tasks, where we usually face sparse data, computational limitations, and long-term prediction needs.

Moreover, new paradigms in deep learning such as deep reinforcement learning (DRL) and semi-supervised learning could be integrated into current GNN frameworks. Specifically, DRL can be employed for graph learning tasks, where a reward function is defined for learning the best structure of a graph that yields to the highest reward according to the problem-specific strategy. Semi-supervised learning might also be used for situations where there are not sufficient training data for training the GNN frameworks. This is a common issue in many traffic forecasting and demand modeling problems.

E. Generalizability

The generalizability of a model refers to its ability to perform well in unseen situations (the situations that were not available in the training step). This is really important in many transportation problems, and ironically, these unseen situations are usually those that are more important compared to normal situations based on which the models are trained. For instance, training a path planning model for autonomous vehicles based on regular driving data could result in unsafe maneuvers in critical situations and corner cases for which there had not been enough data during the training process (this is because crashes are rare events compared to regular driving). As another example, traffic forecasting models should perform well in adverse and unexpected weather conditions or in special events. Accordingly, investigating the generalizability of GNNs is an important issue and should be considered during the development of such models. However, according to the current literature review, it seems that this important property has been mainly overlooked.

F. Efficiency

One of the biggest challenges with GNNs in real-world applications, especially in the transportation domain, is their ability to handle large amounts of data in high-dimensional graphs in real-time. Most of the current studies have focused on the performance of their proposed framework mostly in terms of accuracy and error minimization, and few studies have tried to demonstrate the run-time efficiency of the proposed frameworks in real-world applications. Although the conducted experiments indicate promising results of GNNs, it is also important to compare the proposed frameworks with the baseline models in terms of run time and other efficiency measures. In addition, current studies mostly have utilized a relatively small sub-network of real urban networks in their study, while for real-world applications of GNNs, it is important that these models are capable of handling large amounts of data on large graphs of urban networks. The challenge is actually to develop GNN frameworks to enable the maximum possible performance while being flexible to handle flexible graph sizes. Utilizing GNN accelerators and efficient software frameworks that enable parallel processing of GNNs can also be helpful. Previous works have also demonstrated that software-hardware cooperation could hugely benefit the efficiency of GNN frameworks [14]. The graph awareness approach, which means being aware of the graph characteristics like its size and input features dimension, has proved to be useful for speeding up GNNs [213], [214].

G. Data Heterogeneity

One big challenge for developing accurate and efficient GNN-based frameworks in the transportation domain is data heterogeneity. Transportation data usually come from different sensors, are accompanied by noise, have a time-varying and dynamic nature, and are of different types. Most of the current studies have utilized only one source of transportation data while accommodating the data coming from different types of sensors, and dealing with imprecise and noisy measurements can be considered in future studies. Also, integrating data fusion into the prediction tasks using graph neural networks [215] can improve the robustness of the predictions.

H. Interpretability

One of the drawbacks of many deep learning methods, in spite of their promising performance, is their relatively low interpretability. Although in some domains, the model performance and its prediction accuracy might be more important, in the transportation domain, which is closely concerned with decision-making and what-if analysis, interpretability is crucially important. It is not surprising that many decision-makers and urban planners prefer to apply even less accurate models that are instead more interpretable just because they prefer to understand how the model is working and how they can test and evaluate different policies using interpretable, predictive models. Moreover, interpretability helps to understand the mechanism behind

prediction, which could facilitate identifying the model deficiencies and improving its performance in edge-case scenarios.

To make the long story short, for any deep learning model (with no exception for GNNs) to be applicable and attractive for real-world applications, interpretability should be guaranteed at least to some degree. Although several attempts have been made to make deep learning models more interpretable, this area is still an open issue and requires specific attention. Few studies utilizing GNNs for transportation prediction purposes have evaluated the interpretability of the proposed framework, which makes this area an open one for future research.

I. Transfer Learning

Transfer learning means applying a model that is trained for a specific task to another task (or as an initial model for another task). It is common in many areas of deep learning, especially in image recognition and classification, but has not been adequately discussed and investigated in previous studies on GNNs in the transportation applications domain. Transfer learning is essential in intelligent transportation applications because, in many situations, there is not enough data for training a deep learning model from scratch, and using a pre-trained model could be of great benefit. For instance, A model might be trained with traffic data from loop detectors in a specific city with enough detectors and then could be used as a base framework for another city with more sparse data or less frequent traffic data. As another example, models trained based on the data from a specific mode (road traffic counts) might be used for prediction purposes in other contexts or for other travel modes (such as public transportation passenger flow prediction or shared mobility demand forecasting).

VII. CONCLUSION

Graph neural networks have shown promising for in intelligent transportation applications and therefore have been widely used in different fields of transportation. Although there have been a few review studies on GNNs in the transportation domain, they have mainly focused on traffic prediction problems and overlooked some interesting areas, such as autonomous vehicle's operation, transportation safety analysis, intersection management, and urban planning studies. Moreover, previous surveys and studies have addressed a relatively narrow field of GNN applications, which is known as node-level learning tasks; whereas, edge-level and graph-level learning tasks in GNNs can greatly benefit intelligent transportation systems. Therefore, this survey aims to open the discussion toward broader applications of GNNs and demonstrate the overlooked research areas and learning approaches in utilizing graph neural networks for intelligent transportation systems. To this end, different applications of GNNs in the general domain of intelligent transportation systems were reviewed. The reviewed studies were categorized based on their transportation problem in order to explore problem-specific research gaps and challenges. We found out

that although a great number of studies have used GNNs, they are still limited to utilizing a specific functionality of GNNs and further research is needed to fully harness the power of GNNs in the transportation domain. Also, it turned out that there are still many challenges to address and interesting areas to investigate for enabling real-world applications of GNNs. Making GNN models more efficient, interpretable, generalizable, and able to handle heterogeneous data on large graphs are identified as the main challenges facing their real-world applications. Also, graph learning, link prediction/estimation, transfer learning, graph reinforcement learning, extending GNNs to a wider transportation systems application, such as multi-modal travel demand prediction, and utilizing more complex graph structures, such as heterogeneous graphs and hypergraphs, were identified as research directions that are worth consideration for future studies.

APPENDIX A PUBLIC DATASETS AND OPEN-SOURCE CODES

Reviewing the literature, it seems there are valuable sources of public datasets and openly accessible codes that can be used by researchers for model evaluation and comparison purposes. Moreover, these datasets could also be used for evaluating the generalizability of the models across different contexts. In this section, we aim to introduce the popular public datasets and open-source codes that have been used/introduced in previous studies, as well as new datasets we believe are worth considering in future studies. Also, we try to categorize the datasets based on their transportation context and applications and the open-source codes based on their programming languages and platforms.

A. Public Datasets

In this section, we categorized different public datasets that have been used (or have the potential to be used) by researchers for development, evaluation, or comparison purposes.

1) Traffic Data:

- METR-LA

METR-LA, together with the PeMS dataset, is probably the most frequently referenced dataset in traffic forecasting problems, at least in studies using deep learning methods and graph neural networks. The dataset includes the traffic data (speed and volume) collected by loop detectors of highways of Los Angeles county, USA. The dataset used by [6] includes 207 detectors from March 1st to June 30th in 2012 and has been frequently used by researchers in traffic forecasting as a benchmark dataset. This dataset could be downloaded at <https://github.com/liyaguang/DCRNN>.

- PeMS Datasets

Performance Measurement Systems (PeMS) Dataset collected by California Transportation Agencies (<https://pems.dot.ca.gov>) is another popular public dataset in traffic forecasting studies. PeMS data (volume, speed, occupancy, vehicle miles traveled, delay, etc.)

are gathered via more than over 44000 detectors that report data every 30 seconds. The traffic flow (volume data) is aggregated in different time resolutions (i.e. 5 min, hourly, daily, and monthly) and the speed data is aggregated in 5-minute intervals. Different subsets of PeMS dataset have been used by different researchers, and in the following, the most frequently-cited sub-datasets are introduced:

- PeMS-BAY

This dataset which was used by [6] is the other benchmark dataset in many studies. The dataset includes traffic data from 325 sensors in the Bay Area in California. The time slot used by [6] includes a six months interval from 1 January 2017 to 30 June 2017. The dataset is accessible at <https://github.com/liyaguang/DCRNN> and <https://zenodo.org/record/4263971>.

- PeMSD3

This dataset is collected in North Central Area in California and the subset has been used in [216], [217], and [218]. The dataset used in [217] includes the data from 358 sensors from 1 September to 30 November 2018 and is accessible at <https://github.com/Davidham3/STSGCN>, and the dataset used in [218] is from July 10th to August 9th for years 2011 to 2017 and could be downloaded at <https://github.com/AprLie/TrafficStream>.

- PeMSD4

The dataset contains the data in San Francisco Bay Area. The dataset has been used in [219] (325 sensors from 1 January 2017 to 31 March 2017), The time period used in most studies includes 307 sensors from January 1st to February 28th, 2018 [220], [221] and is downloadable at <https://github.com/Davidham3/ASTGCN>.

- PeMSD7, PeMSD7(M), and PeMSD7(L)

The dataset contains the traffic data from district 7 of California (Los Angeles) and different time slots of the dataset have been utilized by researchers. the dataset used in [27], [221], and [220] includes the data for 228 sensors on the weekdays of May to June 2012. The dataset could be accessed via <https://github.com/Davidham3/STGCN>. Ge et al. [219] used the data from selected 204 sensors and the time slot between 1 January to 31 March 2018. Choi et al. [222] employed the data from May and August 2018. The dataset used in [217] includes the data obtained from 1047 sensors between 1 July 2019 and 30 September 2019.

- PeMSD8

This dataset depicts San Bernardino, California. The dataset used in [221], [222], and [220] contains the data from July to August 2016 obtained from 170 sensors on eight roads. The data is accessible at <https://github.com/Davidham3/ASTGCN>.

- PeMS-SF

This dataset includes 440 daily records that describe the occupancy rate (between 0 and 1)

for different lanes in San Francisco Bay area in the USA. The dataset has been downloaded from the California Department of Transportation PEMS website, <http://pems.dot.ca.gov>, and includes the period from 1 January 2008 to 30 March 2009. The samples are aggregated at 10-minute intervals. This dataset is accessible via UC Irvine Machine Learning Repository at <https://archive.ics.uci.edu/ml/machine-learning-databases/00204>.

- **LOOP - Seattle Loop Dataset**
This data is collected via inductive loop detectors in Seattle. The dataset contains the speed data aggregated at 5-minute intervals from freeways I-5, I-405, I-90, and SR-520 for 2015. This dataset has been introduced and used by Cui et al. [4] and could be downloaded at <https://github.com/zhiyongc/Seattle-Loop-Data>.
- **Q-Traffic Dataset**
The Q-Traffic dataset was collected by Baidu and includes a query sub-dataset, traffic speed sub-dataset, and road network sub-dataset. The query sub-dataset is a travel time dataset and contains the starting time-stamp, coordinates of the starting location, coordinates of the destination, and estimated travel time (minutes). The data is collected in Beijing, China between April 1, 2017, and May 31, 2017. The traffic speed sub-dataset contains the speed data for the same time interval for 15,073 road segments covering approximately 738.91 km [223]. This dataset could be downloaded at <https://ai.baidu.com/broad/introduction?dataset=traffic>.
- **Shanghai Speed Data**
This dataset is the speed data-driven from taxi trajectories in Shanghai, China from 1 April 2015 to 30 April 2015. The speed data has been aggregated in 10-minute intervals and is accessible via <https://github.com/xxArbiter/grnn>.
- **Virginia Traffic**
This dataset includes the traffic volume measured every 15 minutes at 36 sensor locations along two major highways in Northern Virginia/Washington D.C. capital region. The dataset is accessible via the UC Irvine Machine Learning Repository at <https://archive-beta.ics.uci.edu/api/static/ml/datasets/608>.
- **Los-loop**
The dataset that is collected in Los Angeles contains the traffic speed data from 207 sensors from 3/1/2012 to 3/7/2012. Traffic speeds are aggregated in 5-minute intervals and the dataset is downloadable from <https://github.com/lehaifeng/T-GCN>.
- **Guangzhou Traffic Dataset**
The dataset consists of 214 anonymous road segments within two months from August 1, 2016, to September 30, 2016, at 10-minute intervals, collected in Guangzhou, China [224]. The dataset is accessible at <https://zenodo.org/record/1205229#.YbfMOVmhW3A>.
- **UVDS**
UVDS dataset includes 104 sensors in Daejeon City, South Korea. The dataset includes three months and the data are aggregated in 5-minute intervals [21].

2) Taxi and Ride-Hailing Systems:

• NYC TLC Trip Record Data (NYC Taxi)

The NYC TLC Trip Record Data is one of the most popular datasets in travel demand studies and has been utilized by many researchers [63], [72], [93], [225], [226]. This dataset includes three sub-datasets:

- The yellow and green taxi trip records including pick-up and drop-off dates/times and locations, and driver-reported passenger counts. The data has been collected and provided by the NYC Taxi and Limousine Commission (TLC). The yellow taxi trip records include the period from 2009 to the present, and the green taxi trip records start from July 2012.
- For-Hire Vehicle (“FHV”) trip records that include pick-up and drop-off date/time and taxi zone location ID. This dataset includes the trip records from 2015 to the present.
- High Volume FHV trip records include the trip records (pick-up/drop-off location and time) for High Volume For-Hire Vehicles from February 2019 in New York City (NYC). The data also indicate whether the trip has been a shared trip or not. TLC-licensed FHV businesses that currently dispatch or plan to dispatch more than 10,000 FHV trips in New York City per day under a single brand, trade, or operating name, referred to as High-Volume For-Hire Services (such as Uber Pool, Lyft Line, etc.). The NYC TLC Trip Record Dataset is accessible at <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>.

- **TaxiBJ** This dataset contains the inflow and outflow demands for taxis collected in Beijing, China. Beijing has been divided into 32*32 grids and the flows have been aggregated into 30-minuted time intervals. The data are provided in four different time slots: 1) 1st Jul. 2013 to 30th Oct. 2013, 2) 1st Mar. 2014 to 30th Jun. 2014, 3) 1st Mar. 2015 to 30th Jun. 2015, 4) 1st Nov. 2015 to 10th Apr. 2016, and is openly accessible at <https://github.com/TolicWang/DeepST>. This dataset has been used for model evaluation by Bai et al. [227].

• TaxiSZ

The dataset that consists of speed data is derived from taxi trajectories in Shenzhen, China from 1 January 2015 to 31 January 2015. The data is aggregated in 15-minute intervals and was introduced by Zhao et al. [30]. The dataset is available at <https://github.com/lehaifeng/T-GCN>.

3) Ride-Hailing Services:

• DiDi GAIA Open Data

This dataset is provided by the Chinese corporation DiDi Chuxing and contains different types of data (for instance, travel time and trajectories) for different cities and time slots. The data can be openly accessible at <https://outreach.didichuxing.com/research/opendata>.

• NYC Citi Bike

The Citi Bike program in New York City, USA, which is operated by NYC Bike Share system generates data, including trip records, a real-time

TABLE V
MOST POPULAR PUBLIC DATASETS UTILIZED FOR GNN STUDIES

Transportation Domain	Dataset	Type of Data	Relevant Studies	Temporal Range	No. Locations	Link to the Dataset
Traffic Forecasting	METR-LA	Speed and Volume	[6]	Mar - Jun 2012	207	https://github.com/liyaguang/DCRNN
	Seattle Loop Dataset	Speed and Volume	[4]	Jan - Dec 2015	323	https://github.com/zhuyongce/Seattle-Loop-Data
	PeMS-BAY	Speed and Volume	[6]	Jan - Jun 2017	325	https://github.com/liyaguang/DCRNN
	PeMSD3	Speed and Volume	[216]	Sep - Nov 2018	358	https://github.com/Davidham3/STSGCN
	PeMSD4	Speed and Volume	[217], [218]	Jul 2011 - Aug 2017		https://github.com/AprLie/TrafficStream
			[219]	Jan - Mar 2017	325	...
	PeMSD7, 7M, 7L	Speed and Volume	[220], [221]	Jan - Feb 2018	307	https://github.com/Davidham3/ASTGCN
			[221], [27], [220]	May - Jun 2012	228	https://github.com/Davidham3/STGCN
			[219]	Jan - Mar 2018	204	...
			[222]	May - Aug 2018		...
	PeMSD8	Speed and Volume	[217]	Jul - Sep 2019	1047	...
			[221], [27], [220]	May - Jun 2012	170	https://github.com/Davidham3/ASTGCN
	PeMS-SF	Occupancy	...	Jan 2008 - Mar 2009	-	archive.ics.uci.edu/ml/machine-learning-databases/00204
	Q-Traffic	Speed	-	Apr - May 2017	15073	ai.baidu.com/broad/introduction?dataset=traffic
Travel Demand	Shanghai	Speed	-	Apr 2015		https://github.com/xxArbiter/grnn
	Virginia Traffic	Volume	-	-	36	archive-beta.ics.uci.edu/api/static/ml/datasets/608
	Los-loop	Speed	-	Jan - July 2012	207	https://github.com/lehaifeng/T-GCN
	Guangzhou	-	[224]	Aug - Sep 2016	214	https://zenodo.org/record/1205229#.YbfMOVmhW3A
	NYC Taxi	Pick-up Drop-off location and time	[225], [93], [63], [226]	2013 - Present	-	www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
	TaxiBJ	Passenger flows	[227]	several time slots	-	https://github.com/TolicWang/DeepST
	TaxiSZ	Passenger flows	[227]	Jan 2015	-	https://github.com/lehaifeng/T-GCN
Autonomous Vehicles	NYC CitiBike	Trip Records	-	2013 - Present	-	https://s3.amazonaws.com/tripdata/index.html
	SHMetro	Inflow/Outflows	-	Jul - Sep 2016	288	https://github.com/ivechan/PVCGN
	HZMetro	Inflow/Outflows	-	Jan 2019	80	https://github.com/ivechan/PVCGN
	Hangzhou Metro	Inflow/Outflows	-	Jan 2019	81	tianchi.aliyun.com/competition/entrance/231708/information
	Waymo	Perception, Motion Planning	Many Studies	Updating	103,354	https://waymo.com
	KITTI	Perception, Motion Planning	Many Studies	2012	-	http://www.cvlibs.net/datasets/kitti
	Lyft Level 5	Perception, Motion Planning	Many Studies	Updating	-	https://level-5.global/data
	Berkeley BDD Data	Perception, Motion Planning	Many Studies	2018	-	https://www.bdd100k.com
	PandaSet	Perception	Many Studies	2021	-	https://scale.com/resources/download/pandaset

feed of station status, and monthly reports. The trip history data is available from 2013 at <https://s3.amazonaws.com/tripdata/index.html>.

4) Public Transport Data:

• SHMetro

This dataset is derived from transaction records of the

TABLE VI
A LIST OF OPEN-SOURCE CODES THAT COULD BE UTILIZED AS BENCHMARKS FOR GNN STUDIES

Transportation Domain	Study	Open-source Code	Framework	Link to the Code
Traffic Forecasting	[6]	DCRNN	Python - TensorFlow	https://github.com/liyaguang/DCRNN
	[4]	TGC-LSTM	Python - PyTorch	https://github.com/zhiyongc/Graph_Convolutional_LSTM
	[218]	TrafficStream	Python - PyTorch	https://github.com/AprLie/TrafficStream
	[30]	T-GCN	Python - PyTorch and TensorFlow	https://github.com/lehaifeng/T-GCN
	[32]	ASTGCN	Python - PyTorch	https://github.com/Davidham3/ASTGCN
Travel Demand Modeling	[85]	PVCGN	Python - PyTorch	https://github.com/ivechan/PVCGN
	[30]	T-GCN	Python - PyTorch and TensorFlow	https://github.com/lehaifeng/T-GCN
	[54]	DeepST	Python - TensorFlow	https://github.com/TolicWang/DeepST

Shanghai metro from 1 July 2016 to 30 September 2016. The aggregated inflow and outflows for each metro station at provided at 15-minute time intervals for 288 metro stations. The dataset is accessible via <https://github.com/ivechan/PVCGN>.

- **HZMetro**

This dataset is again from the transaction data from the Hangzhou metro system in China. The dataset contains the inflow and outflows for 80 metro stations aggregated at 15-minute intervals in January 2019. The dataset is accessible via <https://github.com/ivechan/PVCGN>.

- **Hangzhou Metro** The dataset is 25-day subway credit card data records from 1 January 2019 to 25 January 2019 belonging to 81 subway stations of the Hangzhou Metro system in China. This dataset could be downloaded at tianchi.aliyun.com/competition/entrance/231708/information.

5) *Autonomous Vehicles:*

- **Waymo Open Dataset**

Waymo open dataset includes a Motion and a Perception dataset. The motion dataset is comprised of 103,354 segments each containing 20 seconds of object tracks at 10Hz, and the HD map data for the covered area. The perception dataset contains independently-generated labels for data coming from the Waymo autonomous car's lidar and camera sensors. This dataset can be downloaded at <https://waymo.com>.

- **KITTI**

This dataset has been gathered in and around the city of Karlsruhe, Germany using a vehicle equipped with several RGB and monochrome cameras, a Velodyne HDL 64 laser scanner, and an accurate RTK-corrected GPS/IMU localization unit. The dataset can be downloaded at <http://www.cvlibs.net/datasets/kitti>.

- **nuScenes**

The nuScenes dataset has 3D bounding boxes for 1000 scenes (the 20s long at 2Hz frequency) collected in Boston, the US, and Singapore. This dataset includes 28130 samples for training, 6019 samples for validation, and 6008 samples for testing. The data comes from 32-beam LiDAR, 6 cameras, and radars with complete

360° coverage. The dataset can be downloaded at <https://www.nuscenes.org>

- **Lyft Level 5**

This dataset includes two sub-datasets for motion planning and perception. The motion dataset includes the logs of over 1,000 hours of movement of various traffic agents, and the perception dataset includes human-labeled 3D bounding boxes of traffic agents and an underlying HD spatial semantic map. The data can be downloaded at <https://level-5.global/data>.

- **Berkeley BDD Data**

This large dataset consists of 100K driving videos collected from more than 50K rides. Each video is 40-second long and 30fps. More than 100 million frames in total. The dataset could be used for object detection and tracking. The link to the dataset is <https://www.bdd100k.com>.

- **PandaSet**

This dataset combines the data from different sources (LiDAR, Camera, and On-board GPS/IMU) for object detection and segmentation purposes. The data can be downloaded at <https://scale.com/resources/download/pandaset>.

6) *Other Datasets:*

- **NGSIM** This NGSIM dataset was collected under the Next Generation Simulation (NGSIM) project by Federal Highway Administration at the US Department of Transportation. The whole dataset consists of four sub-datasets: I-80 and I-101 datasets gathered from freeways, and Lankershin and Leachtree gathered from two arterial corridors). The data was collected with the aim to support the development of algorithms for driver behavior at microscopic levels and has been extensively used by researchers in driver behavior modeling and traffic microsimulation studies [228]. It includes the trajectories of vehicles

A summary of the above-mentioned datasets is presented in Table V.

B. *Open-Source Codes*

In this section, we introduce the open-source codes that have been provided by researchers in the reviewed studies. Also,

we provide some information about the codes in Table VI such as the transportation domain, programming framework, and link to the code). This could be helpful either for model development or performance comparison for future studies.

REFERENCES

- [1] J. Guerrero-Ibañez, J. Contreras-Castillo, and S. Zeadally, "Deep learning support for intelligent transportation systems," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 3, Mar. 2021.
- [2] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3152–3168, Aug. 2020.
- [3] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 99, pp. 144–163, Jan. 2019.
- [4] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.
- [5] B. Shahsavari and P. Abbeel, "Short-term traffic forecasting: Modeling and learning spatio-temporal relations in transportation networks using graph neural networks," Univ. California at Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/ECS-2015-243, 2015.
- [6] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.
- [7] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3904–3924, May 2022.
- [8] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117921.
- [9] W. L. Hamilton, "Graph representation learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 14, no. 3, pp. 1–159, 2020.
- [10] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, "Hypergraph learning: Methods and practices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2548–2566, May 2022.
- [11] C. Stamile, A. Marzullo, and E. Deusebio, *Graph Machine Learning: Take Graph Data to the Next Level by Applying Machine Learning Techniques and Algorithms*. Birmingham, U.K.: Packt, 2021.
- [12] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
- [13] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.
- [14] S. Abadal, A. Jain, R. Guirado, J. López-Alonso, and E. Alarcón, "Computing graph neural networks: A survey from algorithms to accelerators," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–38, Dec. 2022.
- [15] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 249–270, Jan. 2022.
- [16] L. Sun et al., "Adversarial attack and defense on graph data: A survey," 2018, *arXiv:1812.10528*.
- [17] V.-A. Darvari, S. Hailes, and M. Musolesi, "Goal-directed graph construction using reinforcement learning," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 477, no. 2254, Oct. 2021, Art. no. 20210168.
- [18] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: A comprehensive review," *Comput. Social Netw.*, vol. 6, no. 1, pp. 1–23, Dec. 2019.
- [19] J. Rico, J. Barateiro, and A. Oliveira, "Graph neural networks for traffic forecasting," 2021, *arXiv:2104.13096*.
- [20] K.-H. N. Bui, J. Cho, and H. Yi, "Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues," *Appl. Intell.*, vol. 52, pp. 2763–2774, Jun. 2021.
- [21] K.-H. N. Bui, H. Yi, and J. Cho, "UVDS: A new dataset for traffic forecasting with spatial-temporal correlation," in *Proc. Asian Conf. Intell. Inf. Database Syst.* Cham, Switzerland: Springer, 2021, pp. 66–77.
- [22] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: Recent advances and new challenges," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 2, pp. 93–109, Apr. 2018.
- [23] E. L. Manibardo, I. Lana, and J. D. Ser, "Deep learning for road traffic forecasting: Does it make a difference?" *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6164–6188, Jul. 2022.
- [24] A. Ermagun and D. Levinson, "Spatiotemporal traffic forecasting: Review and proposed directions," *Transp. Rev.*, vol. 38, no. 6, pp. 786–814, 2016.
- [25] A. M. Avila and I. Mezić, "Data-driven analysis and forecasting of highway traffic dynamics," *Nature Commun.*, vol. 11, no. 1, pp. 1–16, Apr. 2020.
- [26] N. G. Polson and O. S. Vadim, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Dec. 2017.
- [27] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*.
- [28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
- [29] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "GaAN: Gated attention networks for learning on large and spatiotemporal graphs," 2018, *arXiv:1803.07294*.
- [30] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.
- [31] Y. Shin and Y. Yoon, "Incorporating dynamicity of transportation network with multi-weight traffic graph convolutional network for traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2082–2092, Mar. 2022.
- [32] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.
- [33] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1720–1730.
- [34] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," 2019, *arXiv:1906.00121*.
- [35] A. V. D. Oord et al., "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
- [36] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1234–1241.
- [37] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 753–763.
- [38] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003.
- [39] S. Roberts, M. Osborne, M. Ebdon, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philos. Trans. Roy. Soc. London A, Math. Phys. Sci.*, vol. 371, no. 1984, 2013, Art. no. 20110550.
- [40] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 95–104.
- [41] S.-Y. Shih, F.-K. Sun, and H.-Y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 108, nos. 8–9, pp. 1421–1441, Sep. 2019.
- [42] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3529–3536.
- [43] L. Chen, W. Shao, M. Lv, W. Chen, Y. Zhang, and C. Yang, "AARGNN: An attentive attributed recurrent graph neural network for traffic flow prediction considering multiple dynamic factors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17201–17211, Oct. 2022.
- [44] J. L. Toole, S. Colak, B. Sturt, L. P. Alexander, A. Evisukoff, and M. C. González, "The path most traveled: Travel demand estimation using big data resources," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 162–177, Sep. 2015.
- [45] R. Abduljabbar, H. Dia, S. Liyanage, and S. A. Bagloee, "Applications of artificial intelligence in transport: An overview," *Sustainability*, vol. 11, no. 1, p. 189, 2019.
- [46] G. Guo and T. Zhang, "A residual spatio-temporal architecture for travel demand forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102639.

- [47] D. Wang, Y. Yang, and S. Ning, "DeepSTCL: A deep spatio-temporal ConvLSTM for travel demand prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
- [48] I. Saadi, M. Wong, B. Farooq, J. Teller, and M. Cools, "An investigation into machine learning approaches for forecasting spatio-temporal demand in ride-hailing service," 2017, *arXiv:1703.02433*.
- [49] G. Jin, Y. Cui, L. Zeng, H. Tang, Y. Feng, and J. Huang, "Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102665.
- [50] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 3656–3663.
- [51] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Q. Z. Sheng, "STG2Seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," 2019, *arXiv:1905.10069*.
- [52] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
- [53] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Oct. 2016, pp. 1–4.
- [54] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.
- [55] H. Yao et al., "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.
- [56] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.
- [57] J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *J. Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.
- [58] J. Chu et al., "Passenger demand prediction with cellular footprints," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 252–263, Jan. 2022.
- [59] R. K. Pace, R. Barry, J. M. Clapp, and M. Rodriguez, "Spatiotemporal autoregressive models of neighborhood effects," *J. Real Estate Finance Econ.*, vol. 17, no. 1, pp. 15–33, 1998.
- [60] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3146–3154.
- [61] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 1227–1235.
- [62] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 362–373.
- [63] J. Ke, X. Qin, H. Yang, Z. Zheng, Z. Zhu, and J. Ye, "Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network," *Transp. Res. C, Emerg. Technol.*, vol. 122, Jan. 2021, Art. no. 102858.
- [64] Z. Li et al., "A multi-stream feature fusion approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1456–1466, Feb. 2022.
- [65] N. Hu, D. Zhang, K. Xie, W. Liang, and M.-Y. Hsieh, "Graph learning-based spatial-temporal graph convolutional neural networks for traffic forecasting," *Connection Sci.*, vol. 34, no. 1, pp. 429–448, Dec. 2022.
- [66] H. Yang, X. Zhang, Z. Li, and J. Cui, "Region-level traffic prediction based on temporal multi-spatial dependence graph convolutional network from GPS data," *Remote Sens.*, vol. 14, no. 2, p. 303, Jan. 2022.
- [67] K. Guo, Y. Hu, Z. Qian, Y. Sun, J. Gao, and B. Yin, "Dynamic graph convolution network for traffic forecasting based on latent network of Laplace matrix estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1009–1018, Feb. 2022.
- [68] G. Jin, Z. Xi, H. Sha, Y. Feng, and J. Huang, "Deep multi-view graph-based network for citywide ride-hailing demand prediction," *Neurocomputing*, vol. 510, pp. 79–94, Oct. 2022.
- [69] Z. Huang, W. Zhang, D. Wang, and Y. Yin, "A GAN framework-based dynamic multi-graph convolutional network for origin-destination-based ride-hailing demand prediction," *Inf. Sci.*, vol. 601, pp. 129–146, Jul. 2022.
- [70] S. Feng, J. Ke, H. Yang, and J. Ye, "A multi-task matrix factorized graph neural network for co-prediction of zone-based and OD-based ride-hailing demand," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5704–5716, Jun. 2022.
- [71] E. Eren and V. E. Uz, "A review on bike-sharing: The factors affecting bike-sharing demand," *Sustain. Cities Soc.*, vol. 54, Mar. 2020, Art. no. 101882.
- [72] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," *Transp. Res. C, Emerg. Technol.*, vol. 97, pp. 258–276, Dec. 2018.
- [73] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 397–400.
- [74] G. Xiao, R. Wang, C. Zhang, and A. Ni, "Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks," *Multimedia Tools Appl.*, vol. 80, no. 15, pp. 22907–22925, Jun. 2021.
- [75] X. Ma, Y. Yin, Y. Jin, M. He, and M. Zhu, "Short-term prediction of bike-sharing demand using multi-source data: A spatial-temporal graph attentional LSTM approach," *Appl. Sci.*, vol. 12, no. 3, p. 1161, Jan. 2022.
- [76] G. Li et al., "A data-driven spatial-temporal graph neural network for docked bike prediction," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, May 2022, pp. 713–726.
- [77] L. Liu and R.-C. Chen, "A novel passenger flow prediction model using deep learning methods," *Transp. Res. C, Emerg. Technol.*, vol. 84, pp. 74–91, Nov. 2017.
- [78] Y. Liu, Z. Liu, and R. Jia, "DeepPF: A deep learning based architecture for metro passenger flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 18–34, Apr. 2019.
- [79] D. Luo et al., "Fine-grained service-level passenger flow prediction for bus transit systems based on multitask deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7184–7199, Nov. 2021.
- [80] X.-L. Ma, J.-Y. Zhang, B.-W. Du, C. Ding, and L.-L. Sun, "Parallel architecture of convolutional bi-directional LSTM neural networks for network-wide metro ridership prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2278–2288, Jun. 2019.
- [81] J. Li et al., "Graph CNNs for urban traffic passenger flows prediction," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct. 2018, pp. 29–36.
- [82] Y. Han, S. Wang, Y. Ren, C. Wang, P. Gao, and G. Chen, "Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 6, p. 243, May 2019.
- [83] H. Peng et al., "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Inf. Sci.*, vol. 521, pp. 277–290, 2020.
- [84] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 736–744.
- [85] L. Liu, J. Chen, H. Wu, J. Zhen, G. Li, and L. Lin, "Physical-virtual collaboration modeling for intra- and inter-station metro ridership prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3377–3391, Apr. 2022.
- [86] S. Chen, J. Dong, P. Ha, Y. Li, and S. Labi, "Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 7, pp. 838–857, 2021.
- [87] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, Jun. 2017.
- [88] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv:1801.02143*.

- [89] Y. He, L. Li, X. Zhu, and K. L. Tsui, "Multi-graph convolutional-recurrent neural network (MGC-RNN) for short-term forecasting of transit passenger flow," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18155–18174, Oct. 2022.
- [90] J. Wang, Y. Zhang, Y. Wei, Y. Hu, X. Piao, and B. Yin, "Metro passenger flow prediction via dynamic hypergraph convolution networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7891–7903, Dec. 2021.
- [91] J. Ke, S. Feng, Z. Zhu, H. Yang, and J. Ye, "Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach," *Transp. Res. C, Emerg. Technol.*, vol. 127, Jun. 2021, Art. no. 103063.
- [92] Y. Liang, G. Huang, and Z. Zhao, "Joint demand prediction for multimodal systems: A multi-task multi-relational spatiotemporal graph neural network approach," *Transp. Res. C, Emerg. Technol.*, vol. 140, Jul. 2022, Art. no. 103731.
- [93] J. Hu, B. Yang, C. Guo, C. S. Jensen, and H. Xiong, "Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Apr. 2020, pp. 1417–1428.
- [94] F. Toque, E. Come, M. K. El Mahrsi, and L. Oukhellou, "Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1071–1076.
- [95] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1695–1704.
- [96] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Cham, Switzerland: Springer, 2003, pp. 63–71.
- [97] C. A. Sims, "Macroeconomics and reality," *Econometrica: J. Econ. Soc.*, vol. 48, no. 1, pp. 1–48, 1980.
- [98] S. E. Shladover, "Connected and automated vehicle systems: Introduction and overview," *J. Intell. Transp. Syst.*, vol. 22, pp. 190–200, May 2018.
- [99] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56. Cham, Switzerland: Springer, 2009.
- [100] H.-H. Jebamkious and R. Kashef, "Autonomous vehicles perception (AVP) using deep learning: Modeling, assessment, and challenges," *IEEE Access*, vol. 10, pp. 10523–10535, 2022.
- [101] G. Te, W. Hu, A. Zheng, and Z. Guo, "RGCNN: Regularized graph CNN for point cloud segmentation," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 746–754.
- [102] L. Yi et al., "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.
- [103] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.
- [104] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
- [105] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.
- [106] L. Yi, H. Su, X. Guo, and L. Guibas, "SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2282–2290.
- [107] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.
- [108] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1711–1719.
- [109] M. Liang et al., "Learning lane graph representations for motion forecasting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 541–556.
- [110] W. Jakub, N. Patryk, S. Rafal, and W. Adam, "CVA-GNN: Convolutional vicinity aggregation graph neural network for point cloud classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
- [111] S. Jin, C. Jing, Y. Wang, and X. Lv, "Spatiotemporal graph convolutional neural networks for metro flow prediction," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 403–409, Jun. 2022.
- [112] A. Baghbani, N. Bouguila, and Z. Patterson, "Short-term passenger flow prediction using a bus network graph convolutional long short-term memory neural network model," *Transp. Res. Rec.*, vol. 2677, no. 2, pp. 1331–1340, 2023.
- [113] C. Zhao, X. Li, Z. Shao, H. Yang, and F. Wang, "Multi-featured spatial-temporal and dynamic multi-graph convolutional network for metro passenger flow prediction," *Connection Sci.*, vol. 34, no. 1, pp. 1252–1272, Dec. 2022.
- [114] X. Zou, K. Li, Y. Li, W. Wei, and C. Chen, "Multi-task Y-shaped graph neural network for point cloud learning in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9568–9579, Jul. 2022.
- [115] K. Zhang, M. Hao, J. Wang, C. W. de Silva, and C. Fu, "Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features," 2019, *arXiv:1904.10014*.
- [116] W. Jing, W. Zhang, L. Li, D. Di, G. Chen, and J. Wang, "AGNet: An attention-based graph network for point cloud classification and segmentation," *Remote Sens.*, vol. 14, no. 4, p. 1036, Feb. 2022.
- [117] M. Meyer, G. Kusch, and S. Tomforde, "Graph convolutional networks for 3D object detection on radar data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 3060–3069.
- [118] J. Zhang, X. Zhao, Z. Chen, and Z. Lu, "A review of deep learning-based semantic segmentation for point cloud," *IEEE Access*, vol. 7, pp. 179118–179133, 2019.
- [119] P. Ghorai, A. Eskandarian, Y.-K. Kim, and G. Mehr, "State estimation and motion prediction of vehicles and vulnerable road users for cooperative autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 16983–17002, Oct. 2022.
- [120] X. Li, X. Ying, and M. C. Chuah, "GRIP: Graph-based interaction-aware trajectory prediction," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3960–3966.
- [121] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "SpAGNN: Spatially-aware graph neural networks for relational behavior forecasting from sensor data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9491–9497.
- [122] H. Jeon, J. Choi, and D. Kum, "SCALE-Net: Scalable vehicle trajectory prediction network under random number of interacting vehicles via edge-enhanced graph convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2095–2102.
- [123] L. Gong and Q. Cheng, "Exploiting edge features for graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9211–9219.
- [124] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, "Implicit latent variable model for scene-consistent motion forecasting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 624–641.
- [125] X. Mo, Y. Xing, and C. Lv, "Graph and recurrent neural network-based vehicle trajectory prediction for highway driving," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 1934–1939.
- [126] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17654–17665, Oct. 2022.
- [127] D. Singh and R. Srivastava, "Graph Neural Network with RNNs based trajectory prediction of dynamic agents for autonomous vehicle," *Appl. Intell.*, vol. 52, pp. 12801–12816, Feb. 2022.
- [128] J. Chen, G. Chen, Z. Li, Y. Wu, and A. Knoll, "Attention based graph convolutional networks for trajectory prediction," in *Proc. 6th IEEE Int. Conf. Adv. Robot. Mechatronics (ICARM)*, Jul. 2021, pp. 852–857.
- [129] S. Carrasco, D. F. Llorca, and M. A. Sotelo, "SCOUT: Socially-Consistent and Understandable graph attention network for trajectory prediction of vehicles and VRUs," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 1501–1508.
- [130] K. Li, S. Eiffert, M. Shan, F. Gomez-Donoso, S. Worrall, and E. Nebot, "Attentional-GCNN: Adaptive pedestrian trajectory prediction towards generic autonomous vehicle use cases," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 14241–14247.
- [131] H. Zhou, D. Ren, H. Xia, M. Fan, X. Yang, and H. Huang, "AST-GNN: An attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction," *Neurocomputing*, vol. 445, pp. 298–308, Jul. 2021.

- [132] A. Monti, A. Bertugli, S. Calderara, and R. Cucchiara, "DAG-Net: Double attentive graph neural network for trajectory forecasting," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 2551–2558.
- [133] J. Li, H. Ma, Z. Zhang, J. Li, and M. Tomizuka, "Spatio-temporal graph dual-attention network for multi-agent prediction and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10556–10569, Aug. 2022.
- [134] S. Kumar, Y. Gu, J. Hoang, G. C. Haynes, and M. Marchetti-Bowick, "Interaction-based trajectory prediction over a hybrid traffic graph," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 5530–5535.
- [135] E. Jo, M. Sunwoo, and M. Lee, "Vehicle trajectory prediction using hierarchical graph neural network for considering interaction among multimodal maneuvers," *Sensors*, vol. 21, no. 16, p. 5354, 2021.
- [136] Y. Lu, W. Wang, X. Hu, P. Xu, S. Zhou, and M. Cai, "Vehicle trajectory prediction in connected environments via heterogeneous context-aware graph convolutional networks," *IEEE Trans. Intell. Transp. Syst.*, early access, May 24, 2022, doi: [10.1109/TITS.2022.3173944](https://doi.org/10.1109/TITS.2022.3173944).
- [137] L. Tang, F. Yan, B. Zou, W. Li, C. Lv, and K. Wang, "Trajectory prediction for autonomous driving based on multiscale spatial-temporal graph," *IET Intell. Transp. Syst.*, vol. 17, no. 2, pp. 386–399, 2023.
- [138] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.
- [139] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Feb. 2022.
- [140] A. Khan, A. Ribeiro, V. Kumar, and A. G. Francis, "Graph neural networks for motion planning," 2020, *arXiv:2006.06248*.
- [141] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 11785–11792.
- [142] P. Bachiller, D. Rodriguez-Criado, R. R. Jorvekar, P. Bustos, D. R. Faria, and L. J. Manso, "A graph neural network to model disruption in human-aware robot navigation," *Multimedia Tools Appl.*, vol. 81, no. 3, pp. 3277–3295, Jan. 2022.
- [143] M. Huegle, G. Kalweit, M. Werling, and J. Boedecker, "Dynamic interaction-aware scene understanding for reinforcement learning in autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4329–4335.
- [144] P. Hart and A. Knoll, "Graph neural networks and reinforcement learning for behavior generation in semantic environments," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1589–1594.
- [145] K. Jin and X. Han, "Conquering ghosts: Relation learning for information reliability representation and end-to-end robust navigation," 2022, *arXiv:2203.09952*.
- [146] P. Cai, H. Wang, Y. Sun, and M. Liu, "DiGNet: Learning scalable self-driving policies for generic traffic scenarios with graph neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 8979–8984.
- [147] P. Cai, H. Wang, Y. Sun, and M. Liu, "DQ-GAT: Towards safe and efficient autonomous driving with deep Q-learning and graph attention networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21102–21112, Nov. 2022.
- [148] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.
- [149] M. Klimke, B. Völz, and M. Buchholz, "Cooperative behavior planning for automated driving using graph neural networks," 2022, *arXiv:2202.11376*.
- [150] H. Zhou et al., "Review of learning-based longitudinal motion planning for autonomous vehicles: Research gaps between self-driving and traffic congestion," *Transp. Res. Rec.*, vol. 2676, no. 1, pp. 324–341, 2022.
- [151] S. S. M. Qadri, M. A. Gökçe, and E. Öner, "State-of-art review of traffic signal control methods: Challenges and opportunities," *Eur. Transp. Res. Rev.*, vol. 12, no. 1, pp. 1–23, Dec. 2020.
- [152] P. W. Shaikh, M. El-Abd, M. Khanafer, and K. Gao, "A review on swarm intelligence and evolutionary algorithms for solving the traffic signal control problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 48–63, Jan. 2022.
- [153] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*. Cham, Switzerland: Birkhäuser, 2016, pp. 47–66.
- [154] H. Wei, G. Zheng, V. Gayah, and Z. Li, "Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation," *ACM SIGKDD Explor. Newslett.*, vol. 22, no. 2, pp. 12–18, 2021.
- [155] T. Nishi, K. Otaki, K. Hayakawa, and T. Yoshimura, "Traffic signal control based on reinforcement learning with graph convolutional neural nets," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 877–883.
- [156] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. V. D. Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. Eur. Semantic Web Conf. Cham, Switzerland: Springer*, 2018, pp. 593–607.
- [157] H. Wei et al., "CoLight: Learning network-level cooperation for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 1913–1922.
- [158] P. Koonce and L. Rodegerdts, "Traffic signal timing manual," Federal Highway Admin., Washington, DC, USA, Tech. Rep. FHWA-HOP-08-024, 2008.
- [159] P. Varaiya, "The max-pressure controller for arbitrary networks of signalized intersections," in *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Cham, Switzerland: Springer, 2013, pp. 27–66.
- [160] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proc. Learn., Inference Control Multi-Agent Syst. (NIPS)*, 2016, pp. 1–8.
- [161] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 2496–2505.
- [162] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2019.
- [163] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 128–135, 2010.
- [164] X. Li et al., "Deep imitation learning for traffic signal control and operations based on graph convolutional neural networks," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.
- [165] X. Hu, C. Zhao, and G. Wang, "A traffic light dynamic control algorithm with deep reinforcement learning based on GNN prediction," 2020, *arXiv:2009.14627*.
- [166] H. Wei et al., "PressLight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 1290–1298.
- [167] S. Yang, B. Yang, Z. Kang, and L. Deng, "IHG-MA: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control," *Neural Netw.*, vol. 139, pp. 265–277, Jul. 2021.
- [168] P. Mercader, W. Uwayid, and J. Haddad, "Max-pressure traffic controller based on travel times: An experimental analysis," *Transp. Res. C, Emerg. Technol.*, vol. 110, pp. 275–290, Jan. 2020.
- [169] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "MetaLight: Value-based meta-reinforcement learning for traffic signal control," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1153–1160.
- [170] F.-X. Devailly, D. Larocque, and L. Charlin, "IG-RL: Inductive graph reinforcement learning for massive-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7496–7507, Jul. 2022.
- [171] T. Zhong, Z. Xu, and F. Zhou, "Probabilistic graph neural networks for traffic signal control," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4085–4089.
- [172] J. Yoon, K. Ahn, J. Park, and H. Yeo, "Transferable traffic signal control: Reinforcement learning with graph centric state representation," *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103321.
- [173] G. Kim and K. Sohn, "Area-wide traffic signal control based on a deep graph Q-network (DGQN) trained in an asynchronous manner," *Appl. Soft Comput.*, vol. 119, Apr. 2022, Art. no. 108497.
- [174] M. Wang, L. Wu, M. Li, D. Wu, X. Shi, and C. Ma, "Meta-learning based spatial-temporal graph attention network for traffic signal control," *Knowl.-Based Syst.*, vol. 250, Aug. 2022, Art. no. 109166.
- [175] S. Yang, W. Ma, X. Pi, and S. Qian, "A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources," *Transp. Res. C, Emerg. Technol.*, vol. 107, pp. 248–265, Oct. 2019.

- [176] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res. C Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, 2011.
- [177] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 1186–1193.
- [178] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.
- [179] N. Arora et al., "Hard to park? Estimating parking difficulty at scale," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2296–2304.
- [180] Y. Rong, Z. Xu, R. Yan, and X. Ma, "Du-parking: Spatio-temporal big data tells you realtime parking availability," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 646–654.
- [181] W. Wu and L. Peng, "G-RGAN: A spatiotemporal graph generative adversarial model of parking data recovery," in *Proc. 4th Int. Conf. Intell. Auto. Syst. (ICoIAS)*, May 2021, pp. 181–186.
- [182] D. Zhao et al., "MePark: Using meters as sensors for citywide on-street parking availability prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7244–7257, Jul. 2022.
- [183] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.
- [184] W.-C. Hong, Y. Dong, F. Zheng, and S. Y. Wei, "Hybrid evolutionary algorithms in a SVR traffic flow forecasting model," *Appl. Math. Comput.*, vol. 217, no. 15, pp. 6733–6747, 2011.
- [185] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Dec. 2015.
- [186] X. Xiao, Z. Jin, Y. Hui, Y. Xu, and W. Shao, "Hybrid spatial-temporal graph convolutional networks for on-street parking availability prediction," *Remote Sens.*, vol. 13, no. 16, p. 3338, 2021.
- [187] N. Zhang et al., "Structured knowledge base as prior knowledge to improve urban data analysis," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 7, p. 264, 2018.
- [188] D. Zhu et al., "Understanding place characteristics in geographic contexts through graph convolutional neural networks," *Ann. Amer. Assoc. Geographers*, vol. 110, no. 2, pp. 408–420, 2020.
- [189] H.-W. Wang et al., "Evaluation and prediction of transportation resilience under extreme weather events: A diffusion graph convolutional approach," *Transp. Res. C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102619.
- [190] S. He et al., "RoadTagger: Robust road attribute inference with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10965–10972.
- [191] S. Hu et al., "Urban function classification at road segment level using taxi trajectory data: A graph convolutional neural network approach," *Comput., Environ. Urban Syst.*, vol. 87, May 2021, Art. no. 101619.
- [192] M. Li, S. Gao, F. Lu, K. Liu, H. Zhang, and W. Tu, "Prediction of human activity intensity using the interactions in physical and social spaces through graph convolutional networks," *Int. J. Geographical Inf. Sci.*, vol. 35, no. 12, pp. 2489–2516, Dec. 2021.
- [193] W. R. Tobler, "A computer movie simulating urban growth in the Detroit region," *Econ. Geogr.*, vol. 46, pp. 234–240, Jun. 1970.
- [194] P. Liu and F. Biljecki, "A review of spatially-explicit GeoAI applications in urban geography," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102936.
- [195] M. Graham, B. Hogan, R. K. Straumann, and A. Medhat, "Uneven geographies of user-generated information: Patterns of increasing informational poverty," *Ann. Assoc. Amer. Geographers*, vol. 104, no. 4, pp. 746–764, Jul. 2014.
- [196] L. Thakali, L. Fu, and T. Chen, "Model-based versus data-driven approach for road safety analysis: Do more data help?" *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2601, no. 1, pp. 33–41, Jan. 2016.
- [197] N. Arbabzadeh and M. Jafari, "A data-driven approach for driving safety risk prediction using driver behavior and roadway information data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 446–460, Jun. 2018.
- [198] G. Yannis et al., "Road traffic accident prediction modelling: A literature review," in *Proc. Inst. Civil Eng.-Transp.*, vol. 170, no. 5. London, U.K.: Thomas Telford, 2017, pp. 245–254.
- [199] Y. Zhang, X. Dong, L. Shang, D. Zhang, and D. Wang, "A multi-modal graph neural network approach to traffic risk forecasting in smart urban sensing," in *Proc. 17th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2020, pp. 1–9.
- [200] A. Alaoui and M. W. Mahoney, "Fast randomized kernel ridge regression with statistical guarantees," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
- [201] J. Drugowitsch, "Variational Bayesian inference for linear and logistic regression," 2013, *arXiv:1310.5438*.
- [202] H.-T. Cheng et al., "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.
- [203] Z. Zhou, Y. Wang, X. Xie, L. Chen, and C. Zhu, "Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3786–3799, Aug. 2022.
- [204] Z. Yuan, X. Zhou, and T. Yang, "Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 984–992.
- [205] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 914–921.
- [206] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 5668–5675.
- [207] C. Huang, C. Zhang, P. Dai, and L. Bo, "Deep dynamic fusion network for traffic accident forecasting," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2673–2681.
- [208] L.-Y. Hsu et al., "Temperature prediction and TAIEX forecasting based on fuzzy relationships and MTPSO techniques," *Expert Syst. Appl.*, vol. 37, no. 4, pp. 2756–2770, 2010.
- [209] S. Chandar, A. Reddy, M. Mansoor, and S. Jamadagni, "Road accident proneness indicator based on time, weather and location specificity using graph neural networks," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2020, pp. 1527–1533.
- [210] L. Yu, B. Du, X. Hu, L. Sun, L. Han, and W. Lv, "Deep spatio-temporal graph convolutional network for traffic accident prediction," *Neurocomputing*, vol. 423, pp. 135–147, Jan. 2021.
- [211] Y. Huang, F. Zhang, and J. Hu, "Deep spatial-temporal graph modeling of urban traffic accident prediction," in *Proc. Int. Conf. Image, Vis. Intell. Syst. (ICIVIS)*, Cham, Switzerland: Springer, 2022, pp. 445–455.
- [212] J. Ma, J. Chan, S. Rajasegarar, and C. Leckie, "Multi-attention graph neural networks for city-wide bus travel time estimation using limited data," *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117057.
- [213] C. Tian, L. Ma, Z. Yang, and Y. Dai, "PCGCN: Partition-centric processing for accelerating graph convolutional network," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2020, pp. 936–945.
- [214] M. Yan et al., "HyGCN: A GCN accelerator with hybrid architecture," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2020, pp. 15–29.
- [215] A. Holzinger, B. Malle, A. Saranti, and B. Pfeifer, "Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI," *Inf. Fusion*, vol. 71, pp. 28–37, Jul. 2021.
- [216] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," 2020, *arXiv:2012.09641*.
- [217] X. Chen et al., "TSSRGCN: Temporal spectral spatial retrieval graph convolutional network for traffic flow forecasting," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 954–959.
- [218] X. Chen, J. Wang, and K. Xie, "TrafficStream: A streaming traffic flow forecasting framework based on graph neural networks and continual learning," 2021, *arXiv:2106.06273*.
- [219] L. Ge, S. Li, Y. Wang, F. Chang, and K. Wu, "Global spatial-temporal graph convolutional network for urban traffic speed prediction," *Appl. Sci.*, vol. 10, no. 4, p. 1509, Feb. 2020.
- [220] A. Roy, K. Kumar Roy, A. Ahsan Ali, M. Ashraf Ali, and A. K. M. Mahbubur Rahman, "Unified spatio-temporal modeling for traffic forecasting using graph neural network," 2021, *arXiv:2104.12518*.
- [221] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "LSGCN: Long short-term traffic prediction with graph convolutional networks," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2355–2361.
- [222] J. Choi, H. Choi, J. Hwang, and N. Park, "Graph neural controlled differential equations for traffic forecasting," 2021, *arXiv:2112.03558*.

- [223] B. Liao et al., "Deep sequence learning with auxiliary information for traffic prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 537–546.
- [224] X. Chen, Y. Chen, and Z. He, "Urban traffic speed dataset of Guangzhou, China," [Data set], Zenodo, 2018. [Online]. Available: <https://zenodo.org/record/1205229#.ZBOxbOzMKUs>, doi: 10.5281/zenodo.1205229.
- [225] J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, "Coupled layer-wise graph convolution for transportation demand prediction," 2020, *arXiv:2012.08080*.
- [226] Y. Xu and D. Li, "Incorporating graph attention and recurrent architectures for city-wide taxi demand prediction," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 9, p. 414, 2019.
- [227] L. Bai, L. Yao, S. S. Kanhere, X. Wang, W. Liu, and Z. Yang, "Spatio-temporal graph convolutional and recurrent networks for citywide passenger demand prediction," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2293–2296.
- [228] B. Coifman and L. Li, "A critical evaluation of the next generation simulation (NGSIM) vehicle trajectory dataset," *Transp. Res. B, Methodol.*, vol. 105, pp. 362–377, Nov. 2017.



Saeed Rahmani (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in civil engineering, transportation, and traffic systems engineering from Iran University of Science and Technology, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Transport and Planning, Delft University of Technology. During his M.Sc., he has conducted research on traffic flow modeling in heterogeneous conditions and studied the non-lane-based driving behavior utilizing vehicle trajectories extracted from

aerial videos. Before starting his Ph.D., he was a Senior Research Assistant with the Traffic Research Laboratory and the Project Manager with the Transportation Data Analytics Center, Iran University of Science for five years. At Delft University of Technology, his research is focused on motion planning and prediction for connected and autonomous vehicles and modeling autonomous vehicles in microsimulation for traffic safety and efficiency impact assessment.



Asiye Baghbani received the Bachelor of Science degree in civil engineering from Sharif University of Technology in 2015 and the Master of Science degree in transportation and traffic systems engineering from Iran University of Science and Technology in 2018. She is currently pursuing the Ph.D. degree with the Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada. During her M.Sc., she has conducted research on transportation planning and travel behavior. Her current research focuses on

graph neural networks and their application in the prediction of passenger flows in bus networks.



Nizar Bouguila (Senior Member, IEEE) received the Engineer degree from the University of Tunis, Tunisia, in 2000, and the M.Sc. and Ph.D. degrees in computer science from Sherbrooke University, Sherbrooke, QC, Canada, in 2002 and 2006, respectively. He is currently a Professor with the Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC. He is the author or coauthor of more than 500 publications in several prestigious journals and conferences. His research interests include image processing, machine learning, data mining, computer vision, pattern recognition applied to different real-life problems across various domains, such as quality control, security, smart buildings, health informatics, finance, human resources, and energy. He received the Best Ph.D. Thesis Award in Engineering and Natural Sciences from Sherbrooke University in 2007. He was awarded the prestigious Prix d'excellence de l'association des doyens des études supérieures au Québec (Best Ph.D. Thesis Award in Engineering and Natural Sciences in Québec) and the runner-up for the prestigious NSERC Doctoral Prize. He was the holder of a Concordia University Research Chair Tier 2 from 2014 to 2019 and was named Concordia University Research Fellow in 2020. He is a regular reviewer of many international journals. He is a licensed professional engineer registered in Ontario. He is serving as an Associate Editor for several journals, such as IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, and *Engineering Applications of Artificial Intelligence*.



Zachary Patterson (Member, IEEE) received the Ph.D. degree from McGill University. He was a Post-Doctoral Researcher with the Transport and Mobility Laboratory (TRANSP-OR), Ecole Polytechnique Fédérale de Lausanne, Switzerland. He was an Assistant and Associate Professor with the Department of Geography, Planning Environment, Concordia University, in 2010. In 2021, he joined the Concordia Institute for Information Systems Engineering (CIISE). Professionally, he worked as a Transportation Modeler with the Montreal Regional

Public Transportation Planner, l'Agence Métropolitaine de Transport. His research focuses on using emerging technologies in transport-related data collection, computational geospatial data processing, data analysis and inference with statistical and artificial intelligence techniques, and discrete choice experiments. He is currently Chair of the Transportation Research Board's Travel Survey Methods Committee (AEP25), and member of the Urban Transportation Data and Information Systems Committee (AED20) and the Interuniversity Research Centre on Enterprise Networks, Logistics, and Transportation.