CARPG: Cross-City Knowledge Transfer for Traffic Accident Prediction via Attentive Region-Level Parameter Generation

Guang Yang
Rutgers University
Piscataway, NJ, USA
gy121@cs.rutgers.edu

Yuequn Zhang
Rutgers University
Piscataway, NJ, USA
yz1127@cs.rutgers.edu

Jinquan Hang
Rutgers University
Piscataway, NJ, USA
jh1848@cs.rutgers.edu

Xinyue Feng
Rutgers University
Piscataway, NJ, USA
xf87@cs.rutgers.edu

Zejun Xie
Rutgers University
Piscataway, NJ, USA
zx180@cs.rutgers.edu

Desheng Zhang
Rutgers University
Piscataway, NJ, USA
desheng@cs.rutgers.edu

Yu Yang*
Lehigh University
Bethlehem, PA, USA
yuyang@lehigh.edu

ABSTRACT

Traffic accident prediction is a crucial problem for public safety, emergency treatment, and urban management. Existing works leverage extensive data collected from city infrastructures to achieve encouraging performance based on various machine learning techniques but cannot achieve a good performance in situations with limited data (i.e., data scarcity). Recent developments in transfer learning bring a new opportunity to solve the data scarcity problem. In this paper, we design a novel cross-city transfer learning framework named CARPG for predicting traffic accidents in data-scarce cities. We address the unique challenge of predicting traffic accidents caused by its two fundamental characteristics, i.e., spatial heterogeneity and inherent rareness, which result in the biased performance of the state-of-the-art transfer learning methods. Specifically, we build cross-city region connections by jointly learning the spatial region representations for both source and target cities with an inter-city global graph knowledge transfer process. Further, we design an efficient attention-based parameter-generating mechanism to learn region-specific traffic accident patterns, while controlling the total number of parameters. Built upon that, we ensure that only relevant patterns are transferred to each target region during the knowledge transfer process and further to be fine-tuned. We conduct extensive experiments on three real-world datasets, and the evaluation results demonstrate the superiority of our framework compared with state-of-the-art baseline models.

CCS CONCEPTS

• **Computing methodologies** → **Transfer learning**; • **Applied computing** → **Transportation**; **Forecasting**.

KEYWORDS

Traffic Accident Prediction, Transfer Learning, Graph Neural Networks

*Prof. Yu Yang is the corresponding author of this paper.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614802

ACM Reference Format:
Guang Yang, Yuequn Zhang, Jinquan Hang, Xinyue Feng, Zejun Xie, Desheng Zhang, and Yu Yang. 2023. CARPG: Cross-City Knowledge Transfer for Traffic Accident Prediction via Attentive Region-Level Parameter Generation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614802

1 INTRODUCTION

Due to the fast process of urbanization, traffic accidents have become one of the most significant issues to public safety. According to the report of the World Health Organization (WHO) in 2018, around 1.35 million people are killed each year because of road traffic accidents, which are now the leading cause of death for individuals aged 5 to 29 [20]. Therefore, predicting potential traffic accidents in the future has been increasingly important to help stakeholders (e.g., police and department of transportation) with better planning to prevent accidents and mitigate the impacts.

Various methods have been proposed to enhance the accuracy of traffic accident predictions. In the early stage, conventional techniques such as decision tree, k-nearest neighbor and FP-Tree are applied to explore traffic accident risks [2, 16, 19]. Recent works [1, 3, 24, 27, 33, 34] utilize deep neural networks (e.g., Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Graph Neural Networks (GNN)) to achieve better performance by capturing complex spatial and temporal correlations of traffic accidents. However, the common assumption of these works is the availability of large-scale data, which may not be applicable in many real-world scenarios where the infrastructures are newly developed or the data collection mechanism has just started [9, 30]. This is also known as the issue of *data scarcity*.

Recent advancements in transfer learning bring a new opportunity to solve the issue of data scarcity [6, 17, 22, 23]. They generally follow a common framework with two steps that they first learn knowledge from data-rich source tasks and then transfer the knowledge to data-scarce target tasks. Although recent attempts [15, 18, 29, 31] have shown promising performance in some spatial-temporal prediction tasks, we argue that ***spatial heterogeneity*** and ***inherent rareness***, as the most prominent characteristics of traffic accidents [1, 33, 34], can introduce bias to both the knowledge learning and the knowledge transfer steps. We use New York

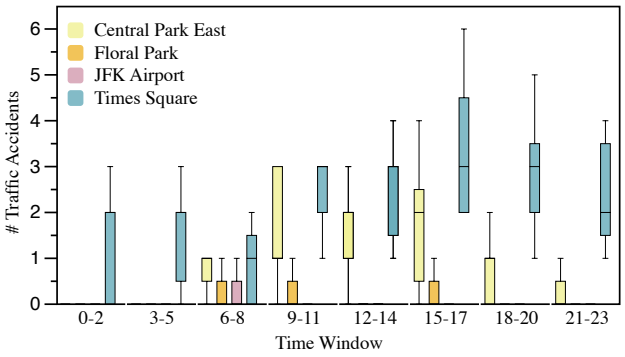


Figure 1: Distribution of Daily Traffic Accidents in New York City over a 7-Day Period Starting from March 2, 2016, Segmented by 3-Hour Time Windows.

City as an example. Fig. 1 shows the distribution of daily traffic accidents over a period of 7 days in four regions of New York City. In the knowledge learning step, we deal with *spatial heterogeneity*, a phenomenon denoting that traffic accident patterns vary from region to region. The existing methods cannot address this phenomenon because they share the patterns (i.e., parameters) among all regions in the source city to achieve the best *overall* performance, which makes the learned patterns biased to regions with major traffic accident patterns. For example, if most of the regions in New York City have similar patterns as Times Square, the patterns of regions such as Central Park East cannot be well learned. Secondly, due to *inherent rareness* of traffic accidents (e.g., only 47 traffic accidents in JFK Airport from 01/01/2016 to 06/30/2016), most regions in the target city generally only have limited accidents and even no accidents if the observed duration is short. This leads to the bias problem in knowledge transfer that the learned patterns from the source city can be fine-tuned more toward regions with more accident data than regions with less data.

Intuitively, to solve the above bias problem, the model should learn unique traffic accident patterns for each region in the source city and subsequently transfer and fine-tune the learned knowledge in a per-region manner for the target city. However, it is non-trivial to achieve this due to the following two challenges: 1) Due to the high-order and dynamic temporal correlations existing in traffic accidents [11, 12], it is challenging to build inter-city region connections with only limited data of several days; 2) Learning region-specific patterns and transferring them at region-level would result in significantly more parameters than learning shared patterns among all regions, which would highly increase the chance of overfitting during the fine-tuning process when there is only limited data in the target city.

To address these challenges, we propose CARPG for traffic accident risk prediction via cross-city knowledge transfer. In response to the first challenge, we use multi-source static information as input and build cross-city region connections via an intra-city region representation learning module followed by an inter-city global graph knowledge transfer module. To tackle the second challenge, we develop an attention-based region-level traffic accident pattern learning module. The parameters for this module are generated through a lightweight parameter pool queried by the learned region representations. This ensures that similar patterns are learned for similar regions across the source and the target city. On this basis, we further adopt a freezing mechanism to ensure each target region will have only relevant patterns transferred from the source city during the knowledge transfer process, which can be subsequently fine-tuned. We summarize our contributions as follows:

- To the best of our knowledge, this is the first work to address the existing transfer learning bias problem in traffic accident risk prediction for data-scarce cities. This is achieved by first learning unique traffic accident patterns for each region in the source city, and then transferring and fine-tuning this knowledge in a per-region manner in the target city.
- Despite the complexity introduced by the high-order dynamics of traffic accidents, we build an inter-city global graph to facilitate knowledge transfer and establish cross-city region similarity connections. An efficient attention-based parameter-generating mechanism is designed to learn region-specific traffic accident patterns, simultaneously controlling the total number of parameters to prevent overfitting.
- We validate the effectiveness of our proposed framework through extensive experiments on 6-month real-world public datasets of New York City, Chicago, and Nashville. The results show CARPG outperforms state-of-the-art methods by up to 12.0% and 15.2% in terms of the regression and ranking perspectives, respectively.

2 RELATED WORK

2.1 Traffic Accident Prediction

There have been many efforts focusing on the prediction of traffic accidents. Early works apply classical techniques such as tree-based methods [2, 16] and K-nearest-neighbor [19] to this problem but have dissatisfying performance due to neglecting spatial and temporal correlations of traffic accidents. Recent works [4, 26, 33] have utilized deep neural networks to capture such correlations and achieved better performance. For example, Chen *et al.* [4] utilize stack denoise autoencoder (SDAE) to predict traffic accident risks at the city level, and Yuan *et al.* [33] employ ConvLSTM with the spatial model ensemble to make predictions in the entire Iowa state. These days, compared to the traditional CNN, GNN has shown better performance in spatial-temporal tasks by taking non-Euclidean correlations among regions into consideration. Zhou *et al.* [34] propose RiskOracle with dynamic graphs to predict traffic accident risk at the minute level. Wang *et al.* [27] build multi-view graphs to model spatial correlations of traffic accidents from different semantic perspectives. An *et al.* [1] transfer knowledge across regions at different risk levels to improve performance.

However, all these models are built upon the assumption that a large amount of data is available to train the model, which may not be applicable in cities with limited data. Instead, our goal is to address the data scarcity issue for traffic accident prediction.

2.2 Knowledge Transfer and Reuse

Transfer learning is a promising direction to solve the data scarcity issue by transferring knowledge from a data-rich source domain to a data-scarce target domain [21, 36]. A lot of methods [5, 13, 17, 35] are proposed following this direction, but most of them focus on transferring knowledge on independent and identically distributed

data, which is unsuitable for traffic accident data with complex spatial and temporal correlations. A few attempts have been made for spatial-temporal knowledge transfer. Wang *et al.* [29] design a deep spatial-temporal transfer learning framework based on the similarity between regions in the source city and target city. To mitigate the risk of model instability and negative transfer, Yao first *et al.* [31] propose to transfer spatial-temporal correlations in a meta-learning paradigm. Moreover, Jin *et al.* [15] assign weights to source regions and conduct selective source training before the fine-tuning stage to rule out irrelevant knowledge. Lu *et al.* [18] conduct spatial-temporal graph learning in the few-shot scenario and propose to learn node-level metaknowledge from multiple cities to enhance feature extraction and knowledge transfer.

However, these models are tailored for common urban events such as crowd flow, bike volume, and traffic demand. As a result, they cannot address the transfer learning bias problem caused by the inherent rareness and spatial heterogeneity of traffic accidents. In this work, we propose a new framework to address this issue.

3 PROBLEM FORMULATION

Definition 1 (Region). Following prior work [31, 32], we partition each city into $I_c \times J_c$ equal-sized grids based on longitude and latitude coordinates. Each grid represents a region and $c \in \{sc, tc\}$ denotes either the source city (sc) or the target city (tc). Note that according to the road network coverage, we only predict traffic accident risks of N_c regions, where N_c is the total number of regions that contain road segments within the border of city c.

Definition 2 (Features). As one of the most challenging traffic problems, traffic accidents have complex correlations with multiple factors. We conduct traffic accident prediction with two groups of features, i.e., dynamic traffic features and static traffic features. Depending on the data availability, the dynamic information might include traffic accident risk, traffic flow, weather, calendar, etc. We denote dynamic features of all regions within the city c at time interval t as $\boldsymbol{X}_{c,t} \in \mathbb{R}^{N_c \times D}$, where N_c is the number of regions with road segments within city c and D is the dimension of dynamic features. Considering that we only have dynamic features available for a few days in the target city, we propose the use of commonly available static information (e.g., points of interest (POI), road network data) as a complementary component to dynamic features. We denote the static features of all regions in the city c as $\boldsymbol{Z}_c \in \mathbb{R}^{I_c \times J_c \times S}$, where S is the dimension of static features.

Definition 3 (Traffic Accident Risk). Following [4, 27, 34], we first classify traffic accidents into three types according to the number of casualties, i.e., $\mathcal{P} = \{minor, injurious, major\}$. Each type is assigned a risk weight w_{tp} to represent its severity. The traffic accident risk $Y_{i,t}$ for a specific region i at a given time interval t is then calculated using the following formula:

$$Y_{i,t} = \sum_{tp \in \mathcal{P}} w_{tp} \times a_{i,t}^{tp}, \tag{1}$$

where w_{tp} is the risk weight, and $a_{i,t}^{tp}$ indicates the number of traffic accidents in region i during time interval t that belong to accident type tp.

Problem Definition. We denote the set of all the time intervals in a city c as

$$\mathcal{T}_c = \{T - |\mathcal{T}_c| + 1, \cdots, T\}, \tag{2}$$

where T is the current/last time interval, $|\mathcal{T}_c|$ is the total number of time intervals in city c. Given a source city sc with rich data and a target city tc with limited data (i.e., $|\mathcal{T}_{sc}| \gg |\mathcal{T}_{tc}|$), our goal is to learn a function $f_{\theta_{tc}}$ to predict the traffic accident risk for all regions in target city tc at the future time interval $T+1$. Here θ_{tc} is the model parameter for target city tc.

4 METHODOLOGY

4.1 Overview

As shown in Fig. 2, we design CARPG to address the transfer learning bias issue in traffic accident prediction. Through the Cross-city Region Representation Learning module and the Region-specific Traffic Accident Pattern Learning module, we build cross-city region connections and learn region-specific traffic accident patterns for the source city during the source training stage. Specifically, with the multi-source static information as the input, we first utilize a CNN layer followed by an intra-city graph learning layer to learn the intra-city region representations. Subsequently, we build an inter-city global graph to transfer region-level knowledge across the source and target cities. This graph also helps to adapt the domain differences. After that, to capture region-specific traffic accident patterns, we construct a lightweight attention-based parameter generation mechanism. This mechanism generates parameters in a per-region manner based on the learned region representations. At last, a Gated Recurrent Unit (GRU) layer followed by a fully connected (FC) layer is used to capture temporal traffic accident patterns and make the final prediction.

Built upon the first two modules, during the knowledge transfer stage, we freeze the learned region representations and part of the parameters in our Region-aware Knowledge Transfer and Fine-tuning module. This ensures that each target region has only its relevant patterns transferred from the source city, based on the established cross-city region similarity connections. These transferred patterns are then further fine-tuned in a region-specific manner.

4.2 Region Representation Learning with Cross-city Graph Knowledge Transfer

To capture the complex regional traffic conditions and establish similarity connections across regions in the source and target cities, we design the Region Representation Learning Module as shown in Fig 2. It contains two sub-modules: 1) the Intra-city Region Representation Learning Module, which captures intra-city region proximity and similarity connections, and 2) the Inter-city Global Graph Knowledge Transfer Module, which facilitates knowledge transfer across cities based on global region similarity. It should be noted that, as different cities usually have different feature spaces, we use different parameter sets for the source and target cities in the Intra-city Region Representation Learning Module to align the domain shift. This adjustment is further refined by the inter-city graph knowledge transfer.

4.2.1 Intra-city Region Representation Learning. The traffic condition and functionality of a region are often closely correlated with

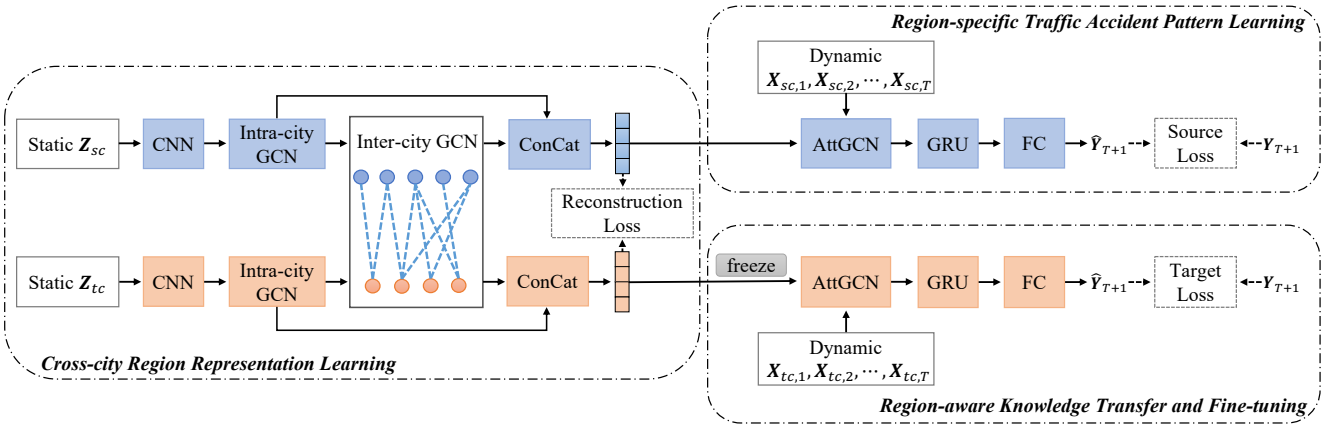


Figure 2: Overview of CARPG. CARPG contains three components: (1) Cross-city Region Representation Learning, (2) Region-specific Traffic Accident Pattern Learning, and (3) Region-aware Knowledge Transfer and Fine-tuning. The first two components establish cross-city region connections based on the region representations and learn region-specific traffic accident patterns for the source city during the source training stage. The third component adopts a freezing mechanism to ensure that only relevant patterns are transferred from the source city to each target region and then fine-tuned.

surrounding regions. For example, the traffic volume of a region highly depends on the road connections with surrounding regions, and several adjacent regions can connect together to serve as a cohesive business area. CNN has achieved great success in capturing local correlations. Thus we leverage these spatial convolutions to capture the local static proximities of regions, which can be formulated as follows:

$$Z^{Conv}_{c,k} = \text{ReLU}(\boldsymbol{W}^{Conv}_{c,k} * Z^{Conv}_{c,k-1} + \boldsymbol{b}^{Conv}_{c,k}), \tag{3}$$

where $*$ is the convolution operation, $\boldsymbol{W}^{Conv}_{c,k}$ and $\boldsymbol{b}^{Conv}_{c,k}$ are learnable parameters of the k-th convolutional layer for city c. $Z^{Conv}_{c,k}$ is the output of k-th convolutional layer. Initially, $Z^{Conv}_{c,0} = Z_c$, and we adopt padding to ensure $Z^{Conv}_{c,k}$ has the same dimension as $Z^{Conv}_{c,0}$ at each layer. At last, we denote the final output of CNN as $Z^{Conv}_c \in \mathbb{R}^{I \times J \times O}$, where O is the number of kernels at the last convolutional layer.

Besides local proximity, traffic conditions also exhibit global similarities across distant regions. In this work, we capture these intra-city global region similarities by the Graph Convolutional Network (GCN), which has shown great potential in capturing such non-Euclidean correlations and achieved great success in spatial-temporal prediction tasks [8, 14, 27, 34]. First, we construct the city graph as $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c, \mathcal{A}_c)$, where $\mathcal{V}_c$ is the set of nodes representing all the regions with road segments in the city c. Each node corresponds to one region and $|\mathcal{V}_c| = N_c$ is the number of regions with road segments. $\mathcal{E}_c$ is the set of edges, while $\mathcal{A}_c$ is the adjacent matrix that denotes the similarity among all nodes.

To construct $\mathcal{A}_c$, we first calculate the cosine similarity $S^{i,j}_c$ between node i and j based on their representations output by the last convolutional layer. Using S_c, we then build $\mathcal{A}_c$ as follows:

$$\mathcal{A}^{i,j}_c = \begin{cases} S^{i,j}_c, & \text{if } j \text{ is top-}L \text{ similar to } i \\ 0, & \text{otherwise.} \end{cases}$$

Then the intra-city GCN layer can be formulated as:

$$Z^{A}_{c,k} = \text{ReLU}(\mathcal{A}_c Z^{A}_{c,k-1} \boldsymbol{W}^{A}_{c,k} + \boldsymbol{b}^{A}_{c,k}), \tag{4}$$

where $\boldsymbol{W}^{A}_{c,k}$ and $\boldsymbol{b}^{A}_{c,k}$ are learnable parameters of the k-th graph convolutional layer for city c. $Z^{A}_{c,k}$ is the output of k-th graph convolutional layer, and $Z^{A}_{c,0} = Z^{Conv}_c$. At last, we denote the final output of the intra-city GCN module as $Z^{A}_c \in \mathbb{R}^{N_c \times G^A}$, where G^A is the number of kernels in the last graph convolutional layer.

4.2.2 Inter-city Global Graph Knowledge Transfer. To transfer only relevant knowledge to the target city at the regional level, we first build similarity connections across regions in the source and target cities. This is based on our proposition that the intra-city *global* similarity connections can be extended across different cities. For example, two regions in different cities can still have similar traffic conditions and traffic accident trends because they are both in school areas with similar surrounding conditions. Therefore, we first construct the inter-city bipartite graph $\mathcal{G}_b = (\mathcal{V}_b, \mathcal{E}_b, \mathcal{A}_b)$ to build region similarity connections across the source and target cities. Then leveraging GCN, we capture inter-city correlations and conduct knowledge transfer.

In the bipartite graph $\mathcal{G}_b$, $\mathcal{V}_b$ represents the set of nodes comprising all regions with road segments from both source and target cities, resulting in $|\mathcal{V}_b| = N_{sc} + N_{tc}$, where N_{sc}, N_{tc} is the number of regions in the source and target cities, respectively. $\mathcal{E}_b$ is the set of edges connecting the source and target regions, and $\mathcal{A}_b$ is the adjacent matrix of the graph denoting the corresponding inter-city region similarity. The construction of $\mathcal{A}_b$ is similar to that of $\mathcal{A}_c$. The only difference is that for each source region in $\mathcal{A}_b$, we only select the top-L most similar regions from the target city to build edges, and the same is done vice versa. The reason we build a distinct bipartite graph, rather than a unified cross-city graph containing $\mathcal{A}_{sc}$ and $\mathcal{A}_{tc}$, is to accommodate the domain and feature distribution differences between the source and target cities. If we were to merge the intra-city and inter-city graphs into a unified

graph based on the region representations, the most similar regions - the top-L - to a given region would likely be within the same city. As a result, most edges would only connect intra-city regions, which would impair the knowledge transfer process across the source and target cities. To address this issue, we construct a bipartite graph in our framework, allowing for more effective knowledge transfer. The inter-city GCN layer can be formulated as follows:

$$Z_k^E = \text{ReLU}(\mathcal{A}_b Z_{k-1}^E W_k^E + b_k^E), \tag{5}$$

where W_k^E and b_k^E are learnable parameters of the k-th graph convolutional layer. Z_k^E is the output of k-th graph convolutional layer, and $Z_0^E = [Z_{sc}^A; Z_{tc}^A]$. At last, we denote the final output of the inter-city global GCN layer as $Z^E = [Z_{sc}^E; Z_{tc}^E] \in \mathbb{R}^{(N_{sc}+N_{tc})\times G^E}$, where G^E is the number of kernels at the last graph convolutional layer. To enhance the region representation, we add a Concatenate layer after the inter-city global GCN layer to concatenate both the intra-city and inter-city representations as $Z_c^{AE} = [Z_c^A, Z_c^E] \in \mathbb{R}^{N_c\times(G^A+G^E)}$, where Z_c^{AE} is the final output of the Region Representation Learning Module for city c.

It is important to note that domain adaptation is typically required for cross-city knowledge transfer because the features of the source and target cities are from different domains and follow different distributions. It is challenging for a prediction model trained on the source city to be directly applied to the target city. Our message-passing process via inter-city GCN not only facilitates the construction of inter-city region connections but also serves as a smoothing process of features across source and target regions, which aligns the domains without the need for an additional domain adaptation module, contributing to the effectiveness and simplicity of our approach.

4.2.3 Reconstruction Loss. To preserve the city-specific properties within the learned region representation, we introduce a reconstruction loss to the output of the Region Representation Learning Module as follows:

$$\mathcal{L}_R = \sum_{c\in\{sc,tc\}} \frac{1}{N_c} ||Z_c^{AE} W^{Rec} - Z_c||^2, \tag{6}$$

where Z_c^{AE} final output of the Region Representation Learning Module for city c, Z_c is the corresponding static features, and W^{Rec} is the learnable parameter that reconstruct Z_c from Z_c^{AE}. The region representations for both the source and target cities are jointly learned during the source training stage.

4.3 Attention-based Region-specific Traffic Accident Pattern Learning

4.3.1 Attention-based Region-specific Graph Convolutions. We propose Attention-based Region-specific GCN (AttGCN) as the cornerstone to tackle the bias issue in transfer learning for traffic accident prediction. This module ensures each region learns unique traffic accident patterns while regions with similarities develop similar patterns. As shown in Fig. 3, AttGCN generates region-specific traffic accident patterns through an attention-based parameter-generating mechanism. This mechanism operates based on a lightweight parameter pool, which contains a set of elementary lightweight parameters that can be considered as the basis of the parameter space.

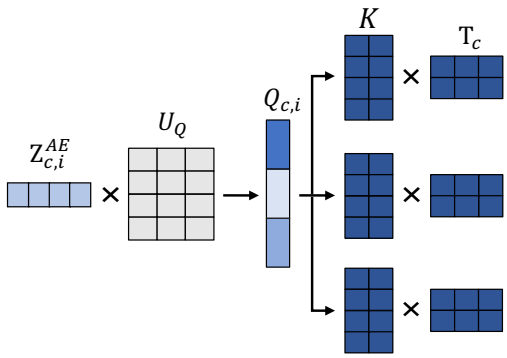


Figure 3: Attention-based Region-specific Parameter Generation in AttGCN for Region i.

We use the same city graph $\mathcal{G}_c$ as defined in Section 4.2.1. A straightforward approach to learning region-specific traffic accident patterns is to assign each region a unique set of parameters. However, this approach could result in an excessive number of parameters, significantly increasing the risk of overfitting during source training. This is particularly problematic when dealing with a large number of regions and a small number of traffic accidents. Moreover, the scarcity of data in the target regions would further increase the risk of overfitting during parameter fine-tuning.

To address the aforementioned challenge, we employ an attention mechanism in AttGCN to generate region-specific parameters by querying a lightweight parameter pool, hence limiting the overall number of parameters. The output of the Region Representation Learning module can be interpreted as a representation of a region's traffic conditions and functionality, which is an essential indicator for traffic accident patterns. Therefore, to generate the region-specific parameters, we first map the region representations into queries, which would query the keys in the lightweight parameter pool. Formally, the query is generated as follows:

$$Q_c = Z_c^{AE} U_Q, \tag{7}$$

where $Z_c^{AE} \in \mathbb{R}^{N_c\times(G^A+G^E)}$ is the region representation of city c, $U_Q \in \mathbb{R}^{(G^A+G^E)\times C}$ is the learnable parameter, $Q_c \in \mathbb{R}^{N_c\times C}$ is the query matrix.

Next, we design a lightweight parameter pool as $W_c^{Pool} = [W_c^{Pool,1}, W_c^{Pool,2}, \cdots, W_c^{Pool,P}] \in \mathbb{R}^{P\times D\times F}$. Each $W_c^{Pool,m}, m \in \{1, 2, \cdots, P\}$, serves as a basis of the parameter space for learning traffic accident patterns. The parameters for each region are generated by the weighted sum of the parameter matrices in W_c^{Pool}. Specifically, we utilize the attention mechanism to generate the weights of the parameter matrices for each region. To generate the key of each weight matrix $W_c^{Pool,m}$, instead of mapping it into a new latent space, we decompose it as follows:

$$W_c^{Pool,m} = K^m T_c^m, \tag{8}$$

where $K^m \in \mathbb{R}^{D\times h}$ is the key matrix and $T_c^m \in \mathbb{R}^{h\times F}$. To enable dot production between Q_c and K^m, we flatten K^m into $\hat{K}^m \in \mathbb{R}^{D\cdot h}$ and set $C = D \cdot h$. Let $\hat{K} = [\hat{K}^1; \hat{K}^2; \cdots; \hat{K}^P]$, the parameters of all regions in city c can then be generated as:

$$W_c^{Att} = \mathcal{A}(Q_c, W_c^{Pool}) = \text{softmax}(\frac{Q_c \hat{K}^T}{\sqrt{P}}) W_c^{Pool}. \tag{9}$$

Similarly, $\boldsymbol{b}_c^{Att}$ can be generated through a comparable mechanism. Finally, the AttGCN layer can be formulated as:

$$\boldsymbol{X}_{c,t,k}^{Att} = \text{ReLU}(\mathcal{A}_c \boldsymbol{X}_{c,t,k-1}^{Att} \boldsymbol{W}_{c,k}^{Att} + \boldsymbol{b}_{c,k}^{Att}), \tag{10}$$

where $\boldsymbol{W}_{c,k}^{Att} \in \mathbb{R}^{N_c \times D \times F}$ and $\boldsymbol{b}_{c,k}^{Att} \in \mathbb{R}^{N_c \times F}$ are learnable parameters of the k-th graph convolutional layer for city c, which are shared among all the time intervals. $\boldsymbol{X}_{c,t,k}^{Att}$ is the output of k-th graph convolutional layer at time interval t in city c, and $\boldsymbol{X}_{c,t,0}^{Att} = \boldsymbol{X}_{c,t}$. Note that, we can reshape $\mathcal{A}_c \boldsymbol{X}_{c,t,k-1}^{Att}$ from $\mathbb{R}^{N_c \times D}$ to $\mathbb{R}^{N_c \times 1 \times D}$ to conduct $3D$ matrix multiplication with $\boldsymbol{W}_{c,k}^{Att}$. At last, we denote the final output of the inter-city global GCN module as $\boldsymbol{X}_{c,t}^{Att} \in \mathbb{R}^{N_c \times G^{Att}}$, where G^{Att} is the number of kernels at the last graph convolutional layer. It is worth mentioning that during the source training stage, we only train AttGCN for the source city. And it will be fine-tuned using the limited data of the target city during the following stage. Moreover, $\boldsymbol{U}_Q$ and $\boldsymbol{K}$ are frozen during the knowledge transfer and fine-tuning stage. More details regarding this process will be provided in a subsequent section.

4.3.2 Temporal Pattern Learning and Prediction. Traffic accidents exhibit not only spatial but also temporal patterns. For example, the traffic accidents that happened in the previous several hours might have long-lasting influences on the nearby traffic conditions, thereby affecting the likelihood of accidents in the current time interval. To capture such temporal correlations, we introduce a GRU layer [28] after the AttGCN layer, which could be formulated as:

$$\boldsymbol{h}_{c,i,t} = \text{GRU}(\boldsymbol{X}_{c,i,t}^{Att}, \boldsymbol{h}_{c,i,t-1}), \tag{11}$$

where $\boldsymbol{X}_{c,i,t}^{Att}$ is the output of the AttGCN layer for region i at time interval t of city c. $\boldsymbol{h}_{c,i,t}$ and $\boldsymbol{h}_{c,i,t-1}$ are the hidden states of GRU layer at time interval t and $t-1$, respectively. We denote the final output of the GRU layer as $\boldsymbol{H}_{c,t} \in \mathbb{R}^{N_c \times R}$, where R is the number of hidden units of the GRU layer. At last, we get the final prediction by projecting $\boldsymbol{H}_{c,t}$ onto $\hat{\boldsymbol{Y}}_{c,t}$ using a fully connected layer.

4.4 Region-aware Knowledge Transfer and Fine-tuning

To address the transfer learning bias problem caused by spatial heterogeneity and the inherent rareness of traffic accidents, we ensure that only relevant patterns are transferred for each target region and further to be fine-tuned at region-level, effectively filtering out irrelevant patterns. This is built upon our attention-based region-specific traffic accident patterns learning mechanism. Given that region representations for both source and target cities are jointly learned, and we create an inter-city global graph for knowledge transfer and domain adaptation, regions with similar traffic conditions and urban functionalities are assured of having similar representations across cities. Consequently, since the parameters for each region are generated using the attention mechanism queried by the region representations, we freeze the query projection matrix $\boldsymbol{U}_Q$ and key matrix $\boldsymbol{K}$ to ensure that regions with similar representations will have similar parameters (i.e., traffic accident patterns) transferred from the source city to the target city as the initialization. Moreover, due to the freezing of the region representations and both the query projection matrix $\boldsymbol{U}_Q$ and key matrix $\boldsymbol{K}$, the attention scores for each parameter matrix in the parameter pool $\boldsymbol{W}_c^{Pool}$ for each target region remain constant. This means that during the fine-tuning process, each target region is assigned different levels of attention to fine-tune different parameter matrices. This region-specific fine-tuning process further mitigates the bias problem.

4.5 Loss Function

Inspired by [27], we employ a weighted loss to assign greater importance to regions with higher traffic accident risk levels. Consequently, the network pays more attention to high-risk regions, thereby mitigating the zero-inflation issue. Formally, the weighted prediction loss for city c is formulated as follows:

$$\mathcal{L}_{c,W} = \frac{1}{N_c} \sum_{rl} w_{rl} ||\boldsymbol{Y}_c(rl) - \hat{\boldsymbol{Y}}_c(rl)||^2, \tag{12}$$

where $\boldsymbol{Y}_c(rl)$ and $\hat{\boldsymbol{Y}}_c(rl)$ are all the samples of ground truth and prediction results with risk weight level rl, respectively. w_{rl} is the weight for risk level rl.

Finally, we combine both the reconstruction loss $\mathcal{L}_R$ and the weighted prediction loss $\mathcal{L}_{sc,W}$ on the source city as the final source training loss:

$$\mathcal{L} = \lambda \mathcal{L}_R + (1-\lambda)\mathcal{L}_{sc,W}, \tag{13}$$

where λ is a hyperparameter. During the fine-tuning stage, we utilize the weighted prediction loss on the target city $\mathcal{L}_{tc,W}$ as the fine-tuning loss.

Table 1: Datasets Statistics.

Dataset	NYC	Chicago	Nashville
Time Span	01/01/2016-06/30/2016		
# Traffic Accidents	83,321	16,971	16,432
# Taxi Orders	77,411,325	13,575,563	-
# PoIs	27,004	17,127	3,234
Road Lengths (km)	16,408	17,744	11,346

5 EXPERIMENT

To validate the effectiveness of CARPG, we conduct extensive experiments on public real-world datasets from three cities. Through our evaluation, we intend to answer the following research questions:

- What is the overall performance of CARPG compared to baseline models when tested across different cities?
- How does each main component contribute to the performance of CARPG?
- How do the key hyperparameters influence the performance of CARPG?
- What are the real-world impacts and potential applications of CARPG?

5.1 Data Description

We conduct the evaluation based on real-world public datasets from three cities: NYC[1], Chicago[2], and Nashville[3]. The data statistics are shown in Table 1.

[1]https://opendata.cityofnewyork.us
[2]https://data.cityofchicago.org
[3]https://data.nashville.gov

5.1.1 Dynamic data. The dynamic data includes traffic accident, taxi trip[4], weather [5], and calendar data from the first half of 2016. We use traffic accident data to measure traffic accident risks, and each record contains traffic accident details including longitude, latitude, timestamp, and the number of casualties. Taxi trip data is used to depict the traffic density of the regions, and each record includes the pick-up and drop-off locations and the timestamp. Besides, we use 6 attributes of hourly weather data (i.e., dew point temperature, dry bulb temperature, precipitation, relative humidity, visibility, and wind speed) as another indicator of the regional traffic conditions. At last, we use the calendar data (i.e., the time of day, the day of week, and holiday) to capture the temporal dynamics.

5.1.2 Static data. This contains the point of interest (PoI) and road network data, which depict the relatively stable regional traffic conditions. Both of these two datasets are obtained from OpenStreetMap[6]. We choose 11 types of common PoIs as the features of each region. And the road segment data include the number of crossings, stop signs, traffic lights, and road lengths.

5.2 Implementation Details

Following the common setting of traffic accident prediction [27, 33, 34] and spatial-temporal transfer learning [15, 29, 31], we partition all three cities (i.e., NYC, Chicago, Nashville) into $2km$ by $2km$ grids, a granularity commonly used by stakeholders (e.g., police and department of transportation) to take reactions. We assign risk weights to different traffic accident types (i.e., minor, injurious, and major) as 1, 2, and 3, respectively. Moreover, we choose New York City which has more public resources available as the source city, and assume Chicago and Nashville are data-scarce target cities for the evaluation purpose. Specifically, following [15], we use the last month for testing and one month before for validation. And we assume all the remaining data are available for training for the source city, but for the target city, we only use the data of a limited period (i.e., 1, 7, 15 days) before the last two months for training.

We implement CARPG in Python with PyTorch, running it on a server with a NVIDIA RTX A4000 GPU. We set the time interval length as 1 hour and use historical data from the preceding 4 hours to predict traffic accident risks for the next 1 hour. For the CNN layer, we set $k = 2$ and the convolution kernel size as 3×3. For all the GCN layers (i.e., Intra-city GCN, Inter-city GCN, AttGCN), we set the number of graph filters to 64 and $k = 1$. When building the graphs, we set $L = 10$. We stack 3 GRU layers with the number of hidden units R set to 64. For the source-training loss, we set λ to 0.5. And we use *Adam* for optimization with a learning rate of $1e^{-4}$.

5.3 Metrics

Following the previous study of traffic accident prediction [27, 34], we conduct the evaluation from both the regression and ranking perspectives. For the regression perspective, we use Root Mean Squared Error (RMSE) to measure the model's ability to gauge the overall city traffic accident risk condition:

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(Y_t - \hat{Y}_t)^2}. \tag{14}$$

For the ranking perspective, we use Recall to evaluate the model's ability to identify regions with high traffic accident risk levels, which is helpful for stakeholders (e.g., police) to allocate resources. More specifically, Recall measures the percentage of predicted regions with top accident risks that have intersections with the regions where traffic accidents really happened as follows:

$$\text{Recall} = \frac{1}{T}\sum_{t=1}^{T}\frac{|\hat{\mathcal{R}}_t^{top} \cap \mathcal{R}_t|}{|\mathcal{R}_t|}, \tag{15}$$

where $\mathcal{R}_t$ is the set of all regions where traffic accidents really happened at time interval t, $\hat{\mathcal{R}}_t^{top}$ is the set of regions with top $|\mathcal{R}_t|$ highest traffic accident risks.

5.4 Baselines

Following common transfer learning settings [15, 29, 31], we compare CARPG with both baselines without knowledge transfer and baselines with knowledge transfer. For baselines without knowledge transfer, we only train the model using the short-term data of the target city. For baselines with knowledge transfer, we first train the model on the data-rich source city and then transfer the learned knowledge and fine-tune it on the data-scarce target city.

(1) Baselines without knowledge transfer:

- **Historical Average (HA)**: The average traffic accident risks of the same time intervals in the training set.
- **Multiple Layer Perception (MLP) [7]**: The fully connected feed-forward neural network.
- **Long Short Term Memory (LSTM) [10]**: A particular type of recurrent neural network used to capture long-term dependencies of time series.
- **Convolutional LSTM (ConvLSTM) [25]**: A deep neural network that combines both CNN and LSTM to capture spatial-temporal patterns.
- **GSNet [27]**: The state-of-the-art GCN-based traffic accident prediction model which captures spatial correlations of traffic accidents from both geographical and semantic perspectives.

(2) Baselines with knowledge transfer:

- **ConvLSTM (FT)**: Train ConvLSTM on the source city and fine-tune it for the target city.
- **GSNet (FT)**: Train GSNet on the source city and fine-tune it for the target city.
- **RegionTrans [29]**: The first deep spatial-temporal transfer learning framework which builds cross-city region connections by directly calculating the similarity between the service or auxiliary data of the regions.
- **CrossTReS [15]**: The state-of-the-art spatial-temporal transfer learning framework for traffic prediction. It conducts selective knowledge transfer by source region re-weighting to mitigate the risk of negative knowledge transfer.

[4]https://www1.nyc.gov
[5]https://www.ncei.noaa.gov/cdo-web/
[6]https://www.openstreetmap.org

Table 2: Evaluation Results for Traffic Accident Risk Prediction in Chicago and Nashville. In each column, the best result is bolded and the second best is underlined.

Baselines	Chicago						Nashville					
	1-day		7-day		15-day		1-day		7-day		15-day	
	RMSE	Recall	RMSE	Recall	RMSE	Recall	RMSE	Recall	RMSE	Recall	RMSE	Recall
Non-transfer												
HA	11.77	6.55%	11.64	9.53%	11.68	11.36%	20.45	4.10%	20.04	8.15%	20.02	11.01%
MLP	22.95	9.34%	11.53	13.22%	11.60	14.22%	26.28	10.11%	18.78	13.68%	19.54	14.84%
LSTM	11.67	12.51%	10.66	13.25%	10.74	14.33%	20.66	12.25%	18.63	16.95%	18.96	17.93%
ConvLSTM	11.70	10.31%	9.39	14.56%	9.58	15.64%	20.64	11.88%	**17.17**	16.99%	17.31	17.96%
GSNet	10.80	8.60%	9.77	16.01%	9.32	16.60%	19.26	8.12%	17.32	17.93%	17.79	18.11%
Transfer												
ConvLSTM (FT)	10.53	12.99%	9.97	14.78%	9.66	15.38%	19.29	14.02%	17.45	18.30%	17.77	18.64%
GSNet (FT)	10.90	12.21%	9.78	15.75%	9.80	16.31%	18.87	14.69%	17.39	18.04%	17.88	18.26%
RegionTrans	11.82	14.21%	11.73	16.08%	11.56	16.30%	20.13	15.33%	20.04	18.69%	19.99	18.70%
CrossTReS	12.22	6.33%	12.04	10.71%	11.88	10.74%	20.72	8.53%	20.08	11.57%	20.30	12.74%
CARPG (ours)	**9.27**	**15.45%**	**9.14**	**16.83%**	**9.01**	**16.72%**	**17.98**	**17.66%**	17.28	**19.20%**	**17.22**	**19.32%**

5.5 Overall Performance

Table 2 shows the overall performance of CARPG for traffic accident risk prediction on the target city with limited data. Generally, CARPG achieves the best performance over all baselines in terms of both RMSE and Recall. It is worth noting that, in the most extreme case for transfer learning (i.e., there is only 1-day data in the target city for training), CARPG can get a significant improvement compared to the second best baseline, which is stable on both the regression and ranking perspectives for the two cities.

For non-transfer baselines, HA and MLP perform poorly due to their inability to capture the spatial-temporal patterns in traffic accidents. Advanced models (e.g., LSTM, ConvLSTM) that are designed to capture spatial or temporal patterns show improved performance. In particular, GSNet achieves the best performance when more training data becomes available in the target city because it is specifically designed to predict traffic accident risks. However, all these deep models assume that a large amount of data is available for model training and thus have suboptimal performance with limited data compared to models with knowledge transfer.

Among knowledge transfer baselines, the fine-tuned baselines (i.e., ConvLSTM(FT), GSNet(FT)) outperform their original versions, particularly when trained on 1-day data. As more data becomes available in the target city, these models do not consistently outperform their original versions due to domain differences between the source and target cities. Simply training a model on the source city and fine-tuning it on the target city may result in negative transfer that impacts the performance. Despite not being explicitly designed for traffic accident prediction, RegionTrans performs better and more consistently than GSNet, as it employs knowledge transfer to overcome data scarcity. However, it just builds cross-city region connections by simply calculating the data similarity and does not transfer region-level patterns (i.e., parameters), making it less suitable for knowledge transfer in traffic accidents. CrossTReS, while conducting selective knowledge transfer by reweighting source regions, still learns and transfers spatial-temporal patterns of all source regions as a whole. Given the spatial heterogeneity of traffic accidents, region-level knowledge transfer is required, leading to the suboptimal performance of CrossTReS.

Table 3: Evaluation Results for Traffic Accident Prediction on Rush Hour in Chicago.

	1-day		7-day		15-day	
	RMSE	Recall	RMSE	Recall	RMSE	Recall
Non-transfer						
HA	11.26	6.96%	8.90	10.19%	8.55	12.56%
MLP	19.84	9.00%	9.36	15.11%	9.44	15.96%
LSTM	9.58	13.24%	8.74	13.41%	8.78	14.18%
ConvLSTM	9.55	11.88%	7.57	15.87%	7.67	17.49%
GSNet	8.87	9.93%	8.10	16.72%	7.46	17.57%
Transfer						
ConvLSTM (FT)	8.48	14.60%	7.97	16.64%	7.67	16.72%
GSNet (FT)	8.86	13.16%	7.76	17.66%	7.83	17.74%
RegionTrans	9.67	16.28%	9.76	15.37%	9.63	17.66%
CrossTReS	10.00	7.05%	9.87	12.14%	9.71	11.64%
CARPG (ours)	**7.63**	**17.32%**	**7.51**	**18.93%**	**7.36**	**18.00%**

In addition to the overall performance, we evaluated CARPG against the baseline models during rush hours (i.e., 7 AM, 8 AM, 9 AM, 4 PM, 5 PM, 6 PM) in Chicago, an essential assessment of model effectiveness during intervals with high traffic accident risks. As shown in Table 3, CARPG continues to outperform the baseline models, which further validates the effectiveness of our framework.

5.6 Ablation Study

To demonstrate the effectiveness of the main components in CARPG, we conduct the ablation study on both Chicago and Nashville with data of only 1 day for target fine-tuning. There are three variants:

- **CARPG-Rec:** We remove the reconstruction loss during the source training process.
- **CARPG-Inter:** We remove the Inter-city Global Graph Knowledge Transfer Module.
- **CARPG-Att:** We replace the AttGCN layer with the traditional GCN layer.

As shown in Fig. 4, for Chicago, the performance suffers the most when the AttGCN layer is replaced with the traditional GCN layer. This is because there is significant spatial heterogeneity of regions in Chicago. Simply using the traditional GCN layer means that the knowledge from New York City is transferred to all regions in Chicago as a whole. This leads to a significant transfer learning

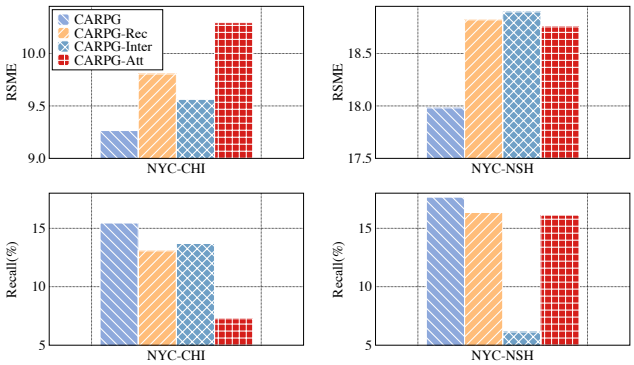


Figure 4: Component Analysis of CARPG.

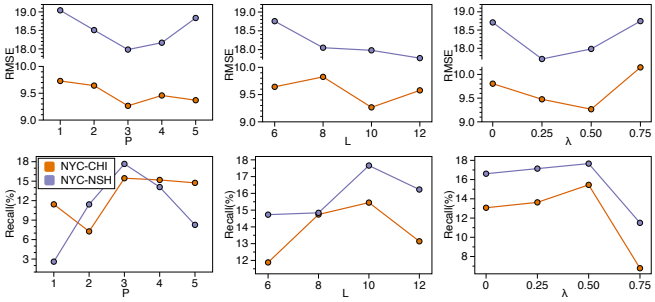


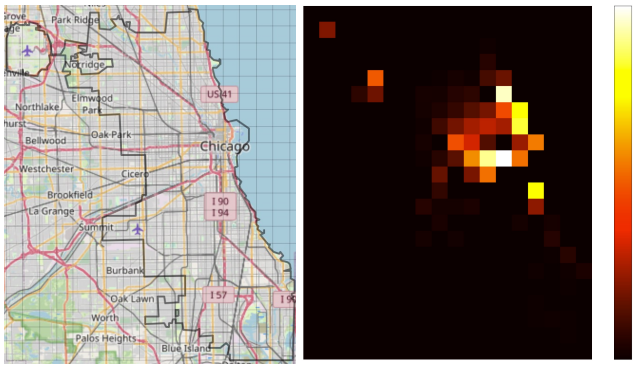
Figure 5: Analysis of Main Hyperparameters of CARPG.

bias, consequently resulting in poor performance. For Nashville, the model performance degrades most when we remove the Inter-city Global Graph Knowledge Transfer Module. The possible reason is the city difference between New York City and Nashville, which lead to a large domain shift in terms of the feature space. The model cannot overcome that issue without an explicit region-level knowledge transfer and domain adaption process. At last, when we remove the reconstruction loss, the model gets lower performance in both cities, which validates the necessity to incorporate the city-specific information into the region representations.

5.7 Hyperparameter Analysis

This section analyzes the effects of three main hyperparameters in our framework. We use 1-day data of the target city for fine-tuning, and the results are shown in Fig. 5. First, we analyze the influence of the number P of weight matrices in the parameter pool $\boldsymbol{W}_c^{Pool}$. The model achieves the best performance in both Chicago and Nashville when there are three matrices in the parameter pool. When P is too small, the parameters are insufficient to learn the region-specific patterns, and when P is too large, it increases the risk of overfitting in the target city due to the limited data. Second, we change the number L of the adjacent nodes for constructing the inter-city knowledge transfer graph. Generally, the model performs best when each region is connected with 10 similar regions across the source and target city (i.e., $L = 10$). Fewer connections diminish the efficacy of cross-city knowledge transfer, while more connections transfer more knowledge from irrelevant regions, thereby lowering performance. Moreover, we consider the trade-off λ between the reconstruction loss $\mathcal{L}_R$ and the weighted prediction loss $\mathcal{L}_{sc,W}$. Generally, the model performance improves when we increase λ from 0 to 0.5. This validates the necessity of adding the reconstruction loss to preserve the city-specific characteristics. However, when λ is too large, there is significant performance degradation due to the lack of supervision of weighted traffic accident information.

5.8 Case Study



(a) Map of Chicago (b) Heat Map of Prediction Results

Figure 6: Visualization of Prediction Results of Chicago for Rush Hours on Weekdays.

In this section, we visualize the prediction result of CARPG for rush hours on weekdays and compare it with the city map of Chicago. As shown in Fig. 6, CARPG successfully highlights the regions in the downtown area of Chicago, which usually have more traffic accidents during rush hours on weekdays. Note that, in this case, study, we only use 1-day data of Chicago for training. Although our model may not explicitly highlights all the high-risk regions throughout the city, it still achieves some success, especially for traffic accidents that exhibit high-order dynamics.

6 CONCLUSION

In this work, we propose a novel transfer learning framework for traffic accident risk prediction in cities with limited data. To address the transfer learning bias issue incurred by the spatial heterogeneity and inherent rareness of traffic accidents, we learn unique traffic accident patterns for each region in the source city and only transfer the relevant knowledge to the regions in the target city based on the cross-city region similarity. The transferred knowledge will further be fine-tuned on a per-region basis during the subsequent stage. To build cross-city region similarity connections, we design a joint region representation learning module with the inter-city global graph knowledge transfer. To overcome the over-fitting issue, we design a lightweight, attention-based graph learning module. This module facilitates the learning of region-specific traffic accident patterns while keeping the total number of parameters in control. We conduct extensive experiments and demonstrate that our framework outperforms state-of-the-art models based on public datasets from three cities.

7 ACKNOWLEDGMENTS

This work is partially supported by NSF 1932223, 1951890, 1952096, 2003874, 2047822. We thank all the reviewers for their insightful feedback to improve this paper.

REFERENCES

[1] Bang An, Amin Vahedian, Xun Zhou, W Nick Street, and Yanhua Li. 2022. HintNet: Hierarchical Knowledge Transfer Networks for Traffic Accident Forecasting on Heterogeneous Spatio-Temporal Data. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 334–342.

[2] Li-Yen Chang and Wen-Chieh Chen. 2005. Data mining of tree-based models to analyze freeway accident frequency. *Journal of safety research* 36, 4 (2005), 365–375.

[3] Chao Chen, Xiaoliang Fan, Chuanpan Zheng, Lujing Xiao, Ming Cheng, and Cheng Wang. 2018. Sdcae: Stack denoising convolutional autoencoder model for accident risk prediction via traffic big data. In *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*. IEEE, 328–333.

[4] Quanjun Chen, Xuan Song, Harutoshi Yamada, and Ryosuke Shibasaki. 2016. Learning deep representation from big and heterogeneous data for traffic accident inference. In *Thirtieth AAAI conference on artificial intelligence*.

[5] Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815* (2009).

[6] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

[7] Matt W Gardner and SR Dorling. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment* 32, 14-15 (1998), 2627–2636.

[8] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 922–929.

[9] Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Li Song, Hui He, and Yu Zheng. 2020. What is the human mobility in a new city: Transfer mobility knowledge across cities. In *Proceedings of The Web Conference 2020*. 1355–1365.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[11] Chao Huang, Chuxu Zhang, Peng Dai, and Liefeng Bo. 2019. Deep dynamic fusion network for traffic accident forecasting. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2673–2681.

[12] Chao Huang, Chuxu Zhang, Peng Dai, and Liefeng Bo. 2021. Cross-interaction hierarchical attention networks for urban anomaly prediction. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 4359–4365.

[13] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. 2006. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems* 19 (2006), 601–608.

[14] Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai, and Weiyang Kong. 2020. LSGCN: Long Short-Term Traffic Prediction with Graph Convolutional Networks.. In *IJCAI*, Vol. 7. 2355–2361.

[15] Yilun Jin, Kai Chen, and Qiang Yang. 2022. Selective Cross-City Transfer Learning for Traffic Prediction via Source City Region Re-Weighting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 731–741.

[16] Lei Lin, Qian Wang, and Adel W Sadek. 2015. A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction. *Transportation Research Part C: Emerging Technologies* 55 (2015), 444–459.

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. PMLR, 97–105.

[18] Bin Lu, Xiaoying Gan, Weinan Zhang, Huaxiu Yao, Luoyi Fu, and Xinbing Wang. 2022. Spatio-Temporal Graph Few-Shot Learning with Cross-City Knowledge Transfer. *arXiv preprint arXiv:2205.13947* (2022).

[19] Yisheng Lv, Shuming Tang, and Hongxia Zhao. 2009. Real-time highway traffic accident prediction based on the k-nearest neighbor method. In *2009 international conference on measuring technology and mechatronics automation*, Vol. 3. IEEE, 547–550.

[20] World Health Organization. 2018. *Global status report on road safety 2018*. World Health Organization.

[21] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[22] Weike Pan, Evan Xiang, Nathan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 24. 230–235.

[23] Weike Pan, Evan W Xiang, and Qiang Yang. 2012. Transfer learning in collaborative filtering with uncertain ratings. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[24] Honglei Ren, You Song, JingXin Liu, Yucheng Hu, and Jinzhi Lei. 2017. A deep learning approach to the prediction of short-term traffic accident risk. *arXiv preprint arXiv:1710.09543* (2017).

[25] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015).

[26] Patara Trirat and Jae-Gil Lee. 2021. DF-TAR: A Deep Fusion Network for Citywide Traffic Accident Risk Prediction with Dangerous Driving Behavior. In *Proceedings of the Web Conference 2021*. 1146–1156.

[27] Beibei Wang, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2021. GSNet: Learning Spatial-Temporal Correlations from Geographical and Semantic Aspects for Traffic Accident Risk Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4402–4409.

[28] Jun Wang, Xiaolei Zhou, Yaochang Liu, Xinrui Zhang, and Shuai Wang. 2022. Multi-scale Temporal Feature Fusion for Time-Limited Order Prediction*. In *Wireless Sensor Networks*, Huadong Ma, Xue Wang, Lianglun Cheng, Li Cui, Liang Liu, and An Zeng (Eds.). Springer Nature Singapore, Singapore, 132–144.

[29] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2019. Cross-city transfer learning for deep spatio-temporal prediction.. In *IJCAI*. 1893–1899.

[30] Ying Wei, Yu Zheng, and Qiang Yang. 2016. Transfer knowledge between cities. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1905–1914.

[31] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*. 2181–2191.

[32] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[33] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. 2018. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 984–992.

[34] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Hengchang Liu. 2020. RiskOracle: A Minute-Level Citywide Traffic Accident Forecasting Framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1258–1265.

[35] Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. 2015. Supervised representation learning: Transfer learning with deep autoencoders. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[36] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.