Hybrid Spatio-Temporal Graph Convolutional Network: Improving Traffic Prediction with Navigation Data

Rui Dai*
Alibaba Group
Beijing, China
daima.dr@alibaba-inc.com

Shenkun Xu*
Alibaba Group
Beijing, China
shenkun.xsk@alibaba-inc.com

Qian Gu
Alibaba Group
Beijing, China
hedou.gq@alibaba-inc.com

Chenguang Ji†
Alibaba Group
Beijing, China
chenguang.jcg@alibaba-inc.com

Kaikui Liu
Alibaba Group
Beijing, China
damon@alibaba-inc.com

ABSTRACT

Traffic forecasting has recently attracted increasing interest due to the popularity of online navigation services, ridesharing and smart city projects. Owing to the non-stationary nature of road traffic, forecasting accuracy is fundamentally limited by the lack of contextual information. To address this issue, we propose the Hybrid Spatio-Temporal Graph Convolutional Network (H-STGCN), which is able to "deduce" future travel time by exploiting the data of upcoming traffic volume. Specifically, we propose an algorithm to acquire the upcoming traffic volume from an online navigation engine. Taking advantage of the piecewise-linear flow-density relationship, a novel transformer structure converts the upcoming volume into its equivalent in travel time. We combine this signal with the commonly-utilized travel-time signal, and then apply graph convolution to capture the spatial dependency. Particularly, we construct a compound adjacency matrix which reflects the innate traffic proximity. We conduct extensive experiments on real-world datasets. The results show that H-STGCN remarkably outperforms state-of-the-art methods in various metrics, especially for the prediction of non-recurring congestion.

CCS CONCEPTS

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

KEYWORDS

Traffic forecasting; Spatio-temporal dependency; Graph convolution; Deep learning; Traffic simulation; Navigation

ACM Reference Format:
Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. 2020. Hybrid Spatio-Temporal Graph Convolutional Network: Improving Traffic Prediction with Navigation Data. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403358

1 INTRODUCTION

Spatio-temporal forecasting has important applications such as weather prediction, transportation planning, etc. Traffic prediction is one classic example. Successful deployment of advanced technologies such as time-dependent routing [1], intelligent traffic light control [23], and proactive traffic management [25] rely considerably on the robust performance of traffic prediction.

Forecasting traffic (travel time) is a challenging task as a diverse spectrum of events affect travel demand. While daily commute is relatively predictable, events including festivals, casual entertainment activities, and adverse weather conditions are subject to strong stochasticity and hard to foretell. The absence of such contextual information renders the evolution of road traffic non-stationary [20]. As a consequence, prior data-driven approaches [15, 18, 24] that use state variable (travel time) as the main input generally performed suboptimally. Several studies [10, 16] incorporated event-relevant features, for example tweet counts or crowd map queries in the model to handle this issue. Its efficacy, however, is restricted to neighborhood of hot spots.

To overcome this problem, we augment machine learning models with intended traffic flow acquired from an online navigation engine. Presently, navigation services including smart route recommendations, audio maneuver guidance, etc., are substantially relied upon by drivers in their daily travel. For instance, according to a third-party[1] report, Amap, the top-tier LBS-service provider in China, served more than 115 million users on the National Day of the People's Republic in 2018. The vast number of planned routes offered by a navigation engine comprehensively reflect live travel demand and provide even greater detail than numerous event-level features. More specifically, aggregating the planned routes produces intended traffic volume, which in turn offers strong clues as to future travel time. Figure 1 illustrates this process.

[*] Rui Dai and Shenkun Xu contributed equally to this paper.
[†] Chenguang Ji is the corresponding author.

[1] https://www.questmobile.com.cn/en

KDD '20, August 23–27, 2020, Virtual Event, CA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403358

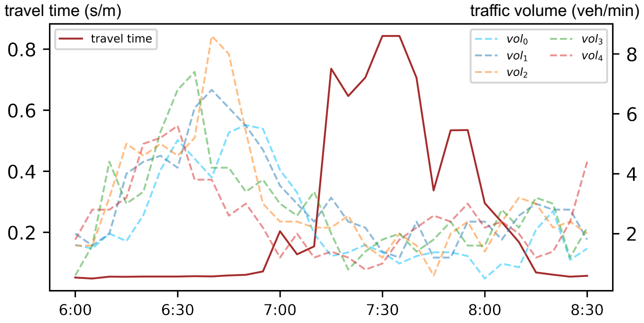
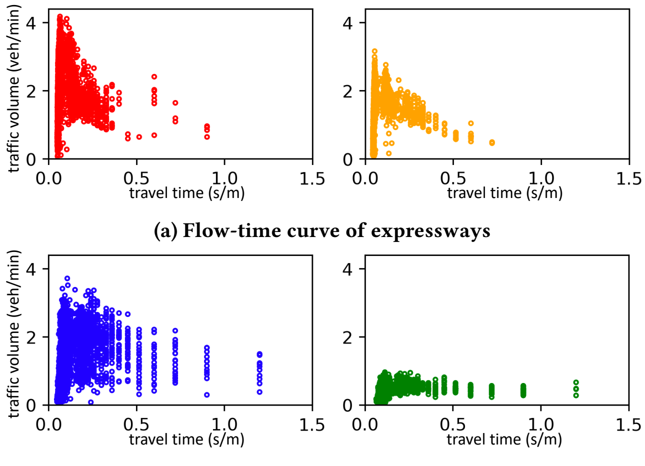


Figure 1: The travel time and intended traffic volume of a road segment in Beijing during morning rush hour on October 28, 2019. vol_f represents the intended traffic volume at a time slot f-step ahead, acquired from the navigation data. The quick rise of intended traffic volume indicates potential traffic congestion.

To integrate this heterogeneous modality into a travel time forecasting model, we design a novel domain transformer to convert traffic volume into its equivalent in travel time. Traffic flow theory [11] establishes that traffic flow and the vehicle density of a road segment satisfies a universal triangular relationship, and specifics of the flow-density diagram such as peak capacity is segment-varying. Figure 2 depicts several real-world examples. To utilize this knowledge, we engineer a flow-to-time transformer with two cascaded mapping components, which are separately responsible for capturing the shared geometric shape and segment-specific characteristics.



(a) Flow-time curve of expressways

(b) Flow-time curve of major roads

Figure 2: Flow-time curve of four different road segments.

Furthermore, owing to the non-Euclidean spatial dependency of road traffic, we adopt graph convolution to extract the shared pattern, and propose a novel adjacency matrix that better reflects innate traffic proximity. In prior scholarship [15], the adjacency matrix generally assumed a node proximity with simple exponential distance-decay, which does not comport with on-the-ground realities. For example, congestion on a major thoroughfare rarely propagates to an intersecting private road where only authorized person can travel, even though the two segments are contiguous. To solve this issue, we propose a compound adjacency matrix which, in addition to the aforementioned spatial attenuation term, incorporates the covariance matrix of road segment travel time.

As a coherent combination of the above-proposed techniques, we develop a novel multi-modal learning architecture for traffic forecasting: the Hybrid Spatio-Temporal Graph Convolutional Network (H-STGCN). In H-STGCN, the transformer first converts intended traffic volume acquired from Amap into its equivalent in travel time. Then shared convolution is applied on individual segments along the temporal dimension to extract high-level patterns. Graph convolution with the compound adjacency matrix further processes the concatenated temporal signal to capture the intrinsic traffic dynamics. The integrated structure is trained end-to-end and capable of foreseeing future congestion based on upcoming traffic flux. We evaluate the proposed model using real-world datasets. Extensive experiments demonstrate that our model has shown remarkable improvements over various state-of-the-art benchmarks.

To summarize, the primary contributions of the paper are as follows:

- We propose to leverage the data of intended traffic flow in a machine-learning model for travel time forecasting. This approach combines the strengths of the recently emerged data-driven approach and the traditional traffic simulation approach [4].
- We design the domain transformer to integrate the heterogeneous modality of traffic flow. This universal coupler naturally adapts to all neural-network based architectures for travel time forecasting.
- We propose the compound adjacency matrix, which encodes innate traffic proximity.
- We construct H-STGCN, a multi-modal learning architecture that significantly outperforms state-of-the-art benchmarks in real world datasets.

The rest of the paper is organized as follows. Section 2 outlines the preliminary concepts and formulates the traffic prediction problem. Section 3 details the structure of the proposed H-STGCN. Section 4 describes the experimental results. Related works are reviewed in Section 5. Finally, Section 6 concludes the paper.

2 PRELIMINARIES

In this section, we provide definitions and outline the forecasting problem. Given a regional network, intersections split it into n directional road segments. We further split time into 5-minute intervals, and denote the time range of training set by $[0, S_{\text{train}})$, test set by $[S_{\text{train}}, S_{\text{train}} + S_{\text{test}})$. We format the data as a tensor $\mathbf{X} \in \mathbb{R}^{n \times (S_{\text{train}}+S_{\text{test}}) \times C^{(\text{in})}}$, where $C^{(\text{in})}$ is the number of input features.

Travel Time / Traffic Volume. Travel time $\tau_{i,t}$ is defined as the average traversing time (per unit length) on segment s_i over time slot t. Similarly, traffic volume $v_{i,t}$ denotes the number of vehicles entering segment s_i within time slot t.

Ideal Future Volume. Given a time slot t_0, ideal future volume $v_{i,t_0,f}$ $(f \geq 0)$ is the counterpart of v_{i,t_0+f} under two conceptual assumptions: 1) only vehicles using a navigation service at t_0 are considered; 2) each vehicle follows the exact planned path and travels at a speed consistent with ETA (estimated time of arrival).

Historical Average (HA). Let L denote the number of time slots in a week. Then the historical average of variable $\omega_{i,t}$ (ideal future volume or travel time) is given by

$$\omega^{(h)}_{i,t} = \frac{1}{W} \sum_{r \equiv t \pmod{L},\, r \neq t,\, r \in [0, S_{\text{train}})} \omega_{i,r}, \tag{1}$$

where W is the number of weeks in the training set.

Traffic Forecasting. Given time t and all available data, traffic forecasting aims to predict future travel time for the whole network. More specifically, provided the sequence of previous traffic features $\{\mathbf{X}_{:,t-P+1,:}, \dots, \mathbf{X}_{:,t,:}\}$, model $\mathscr{H}$ estimates travel time for the next few slots $\mathscr{H}(\mathbf{X}_{:,t-P+1,:}, \dots, \mathbf{X}_{:,t,:}) = \{\hat{\boldsymbol{\tau}}_{:,t+1}, \dots, \hat{\boldsymbol{\tau}}_{:,t+F}\}$, where P denotes the length of input time series, and F the forecasting horizon.

3 METHODOLOGY

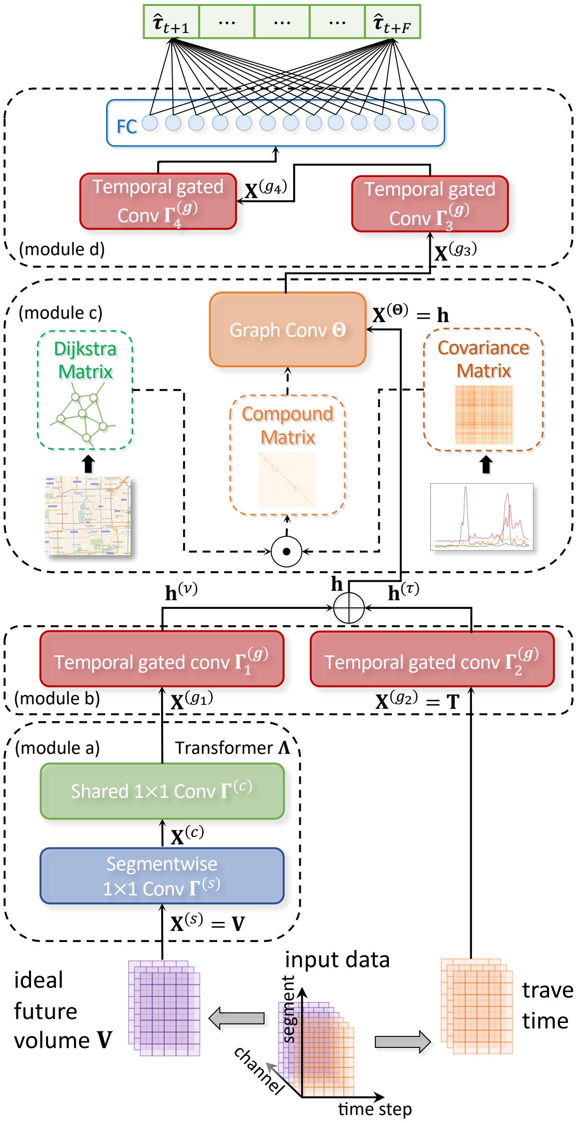
3.1 Overall Architecture

In this section, we describe the overall architecture of H-STGCN, as illustrated in Figure 3. The model input consists of two feature tensors, the ideal-future-volume tensor $\mathbf{V}$ and travel-time tensor $\mathbf{T}$. Specifically, both $\mathbf{V}$ and $\mathbf{T}$ have three dimensions: the spatial dimension, temporal dimension, and channel dimension, which corresponds respectively to the road segments, previous time slots utilized, and features. A domain transformer (module a) first converts each element of $\mathbf{V}$ into its equivalent in travel time, outputting the so-called future-travel-time tensor $\mathbf{X}^{(g_1)}$. Then separate gated convolutions along the temporal dimension (module b) are applied on $\mathbf{X}^{(g_1)}$ and $\mathbf{T}$ to extract the high-level temporal patterns. Treating each segment as a node, a graph convolution with compound adjacency matrix (module c) processes the mixed signal $\mathbf{h} = \mathbf{h}^{\nu} \oplus \mathbf{h}^{\tau}$ to capture the interaction mechanism between traffic volume and travel time ("$\oplus$" stands for the concatenation operator). Next, two additional gated convolutions are applied sequentially to further enlarge the temporal receptive field. Finally, a fully connected (FC) layer outputs the forecasting results. We elaborate each of the modules in subsequent sections.

3.2 Model Input and Data Processing

To forecast future traffic states, H-STGCN uses an input tensor $\mathbf{X}$ with features from all P previous time slots. Each slice of $\mathbf{X}$ that corresponds to a single time slot t ($\leq t_0$) further comprises two categories of features: ideal future volume and travel time. These features and associated data processing techniques are described as follows.

Ideal Future Volume. As an approximation of the unavailable actual future traffic volume, the ideal future volume $v_{i,t_0,f}$ defined in Section 2 can be acquired from an online navigation engine. To employ this feature, we use data from Amap, a leading LBS solution provider in China with over 700 millions users[2]. The architecture of Amap's navigation system is depicted in Figure 4. During navigation, a vehicle synchronizes its location with the cloud server every second to ensure the timely detection of potential deviation and follow-up rerouting. Meanwhile, to keep users posted of the latest traffic condition, cloud servers update the ETA in a near-real-time fashion. In this way, the navigation engine is able to collect the planned path and live ETA of a vehicle, with at most a one-second delay when a detour happens.



$\hat{\boldsymbol{\tau}}_{t+1}$
…
…
…
$\hat{\boldsymbol{\tau}}_{t+F}$
FC
Temporal gated Conv $\Gamma_4^{(g)}$
$\mathbf{X}^{(g_4)}$
Temporal gated Conv $\Gamma_3^{(g)}$
(module d)
$\mathbf{X}^{(g_3)}$
(module c)
Graph Conv $\boldsymbol{\Theta}$
$\mathbf{X}^{(\boldsymbol{\Theta})} = \mathbf{h}$
Dijkstra Matrix
Covariance Matrix
Compound Matrix
$\odot$
$\mathbf{h}^{(\nu)}$
$\mathbf{h}$
$\mathbf{h}^{(\tau)}$
Temporal gated conv $\Gamma_1^{(g)}$
Temporal gated conv $\Gamma_2^{(g)}$
(module b)
$\mathbf{X}^{(g_1)}$
$\mathbf{X}^{(g_2)} = \mathbf{T}$
(module a)
Transformer $\boldsymbol{\Lambda}$
Shared 1×1 Conv $\Gamma^{(c)}$
$\mathbf{X}^{(c)}$
Segmentwise 1×1 Conv $\Gamma^{(s)}$
$\mathbf{X}^{(s)} = \mathbf{V}$
input data
ideal future volume $\mathbf{V}$
segment
channel
time step
travel time $\mathbf{T}$

Figure 3: Architecture of H-STGCN.

Original data acquired from Amap is formally organized as

$$\mathscr{L} = \{(r, \{(\rho_{r,l}, \delta_{r,l}, \psi_r) | l \in [0, M_r)\} | r \in [0, N_{\mathscr{L}})\}, \tag{2}$$

where r is the navigation identifier, ψ_r is the launch time of navigation r, $\rho_{r,l}$ denotes the l-th road segment along the planned route, $\delta_{r,l}$ is the estimated time to arrive at $\rho_{r,l}$, M_r is the total number of road segments on the route, and $N_{\mathscr{L}}$ is the total number of navigation processes. Specifically, routes in Amap are planned with Dijkstra-like algorithms [1], and ETA is forecasted using a machine learning model inferred from historical trajectories. Each

[2] https://www.iresearch.com.cn/

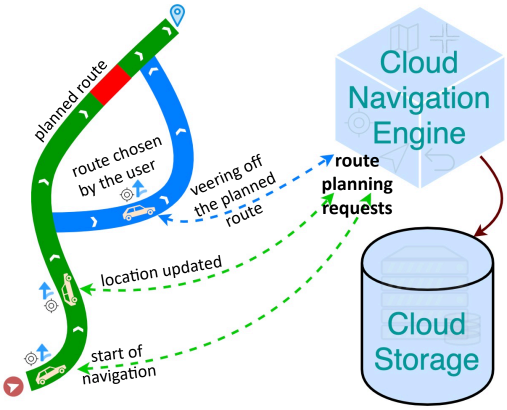


Figure 4: Architecture of Amap's navigation system.

record in $\mathscr{L}$ corresponds exactly to one planned route. Algorithm 1 demonstrates the procedure to obtain the ideal future volume.

In H-STGCN, ideal future volume within the prediction window and the corresponding historical average are both taken as input:

$$\mathbf{V}_{i,t} = \left[v_{i,t,0}, v_{i,t,1}, \ldots, v_{i,t,F}, v^{(h)}_{i,t,0}, v^{(h)}_{i,t+1,0}, \ldots, v^{(h)}_{i,t+F,0}\right], \tag{3}$$

where i is the segment index.

Travel Time. Travel time $\tau_{i,t}$ is calculated using the map-matched [17] GPS data from Amap. In H-STGCN, travel time and its historical average within the prediction window are also both taken as input:

$$\mathbf{T}_{i,t} = \left[\tau_{i,t}, \tau^{(h)}_{i,t}, \tau^{(h)}_{i,t+1}, \ldots, \tau^{(h)}_{i,t+F}\right], \tag{4}$$

where i is the segment index.

3.3 Domain Transformer

Transformer Λ is proposed to convert ideal future volume into its travel time equivalent. In this manner, any network structure originally designed to process the signal of travel time is equally applicable for volume, easing the process of modality integration. Λ consists of two cascaded layers, the shared 1×1 convolution and segmentwise 1×1 convolution, as shown in Figure 3.

Shared 1×1 **Convolution**. A 1×1 convolution $\mathbf{\Gamma}^{(c)}$ shared across all segments and time slots is used as the top layer, aiming to capture the universal triangular-shaped mapping. A schematic of the convolution is shown in Figure 5a. Let $\mathbf{X}^{(c)}_{i,t,:} \in \mathbb{R}^{C^{(c_{\text{in}})}}$, $\mathbf{Y}^{(c)}_{i,t,:} \in \mathbb{R}^{C^{(c_{\text{out}})}}$ denote the input and output, then this layer works as

$$\mathbf{Y}^{(c)}_{i,t,:} = \mathbf{\Gamma}^{(c)}\left(\mathbf{X}^{(c)}_{i,t,:}\right) = \sigma\left(\mathbf{X}^{(c)}_{i,t,:} \cdot \mathbf{F}^{(c)} + \mathbf{b}^{(c)}\right), \tag{5}$$

where $\mathbf{F}^{(c)} \in \mathbb{R}^{C^{(c_{\text{in}})} \times C^{(c_{\text{out}})}}$ is the weight, $\mathbf{b}^{(c)} \in \mathbb{R}^{C^{(c_{\text{out}})}}$ is the bias, and σ is the Exponential Linear Unit (ELU) [5].

Segmentwise 1×1 **Convolution**. To ensure sufficient model capacity to extract the segment-level features, a 1×1 convolution $\mathbf{\Gamma}^{(s)}$ with segment-specific parameters is used as the bottom layer. A schematic of the convolution is shown in Figure 5b. Let $\mathbf{X}^{(s)}_{i,t,:} \in \mathbb{R}^{C^{(s_{\text{in}})}}$, $\mathbf{Y}^{(s)}_{i,t,:} \in \mathbb{R}^{C^{(s_{\text{out}})}}$ denote the input and output, then this layer works as

$$\mathbf{Y}^{(s)}_{i,t,:} = \mathbf{\Gamma}^{(s)}\left(\mathbf{X}^{(s)}_{i,t,:}\right) = \sigma\left(\mathbf{X}^{(s)}_{i,t,:} \cdot \mathbf{F}^{(s)}_{i,:,:} + \mathbf{b}^{(s)}_{i,:}\right), \tag{6}$$

where $\mathbf{F}^{(s)} \in \mathbb{R}^{n \times C^{(s_{\text{in}})} \times C^{(s_{\text{out}})}}$ is the weight, $\mathbf{b}^{(s)} \in \mathbb{R}^{n \times C^{(s_{\text{out}})}}$ is the bias, and σ is an ELU.

Algorithm 1 Route aggregation algorithm to obtain the ideal future volume

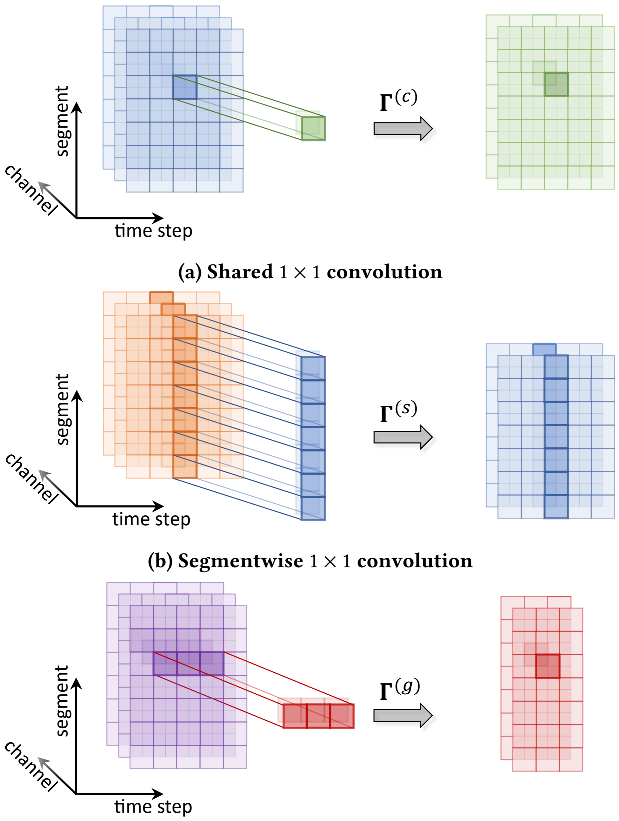
```
Input: The list of route records from the dataset $\mathscr{L}$
Output: Ideal future volume $\boldsymbol{v}$
Initialize $Z$ as an empty set
for each $r \leftarrow 0, 1, \ldots, N_{\mathscr{L}} - 1$ do
    for each $l \leftarrow 0, 1, \ldots, M_r - 1$ do
        $s \leftarrow \rho_{r,l}$    {$s$ is the id of a road segment}
        $t \leftarrow \delta_{r,l}$    {$t$ is a time slot}
        for $f \leftarrow 0, 1, 2, \cdots, F$ do
            if $t \geq \psi_r$ then
                $\zeta \leftarrow (s, t, f)$
                Add $\zeta$ to $Z$
            else
                break
            end if
            $t \leftarrow t - \Delta t$    {$\Delta t$ stands for the length of a single time slot}
        end for
    end for
end for
for each $s_0 \leftarrow 0, 1, \ldots, n - 1$ do
    for each $t_0 \leftarrow 0, 1, \ldots, S_{\text{train}} + S_{\text{test}} - 1$ do
        for each $f_0 \leftarrow 0, 1, \ldots, F$ do
            $\boldsymbol{v}_{s_0,t_0,f_0}$ = cardinality($\{\zeta \mid \zeta.s = s_0, \zeta.t = t_0, \zeta.f = f_0, \forall \zeta \in Z\}$)
        end for
    end for
end for
return $\boldsymbol{v}$
```

3.4 Graph Convolution with Compound Adjacency Matrix

Graph convolution has been utilized as a key building block in many existing architectures [8, 15, 24] to model the non-Euclidean spatial dependency of road traffic. At the core of graph convolution is the weighted adjacency matrix [22], which as a node proximity measure, determines the spectral modes that are amplified or attenuated by the learnable parameters. Our proposed compound adjacency matrix and the formulation of graph convolution are elaborated as follows.

Compound Adjacency Matrix. Adjacency matrix in prior works [15, 24] assumed a node-proximity with simple exponential distance-decay:

$$w^{(d)}_{ij} = \begin{cases} \exp\left(-\frac{d^2_{ij}}{\sigma^2}\right) & , \exp\left(-\frac{d^2_{ij}}{\sigma^2}\right) \geq \epsilon \\ 0 & , \text{otherwise} \end{cases}, \tag{7}$$



(a) Shared 1×1 convolution

(b) Segmentwise 1×1 convolution

(c) Temporal gated convolution (with the gate ommited)

Figure 5: Illustration of convolution operators in H-STGCN.

where d_{ij} is the shortest-path distance between segment s_i and s_j, σ denotes the spatial attenuation length, and ϵ is a cutoff controlling the matrix sparsity. We call $\mathbf{W}^{(d)}$ the Dijkstra matrix in the following. This pure spatial closeness fails to reflect the actual traffic proximity in many scenarios. To be specific, the effect of an occurrence of congestion on traffic diversion depends on several attributes of the adjacent road segment, including the functional class, pavement condition, etc. Thus, congestion propagation is often not spatially uniform, contradicting the aforementioned assumption. To overcome this problem, we propose the compound adjacency matrix $\mathbf{W}^{(c)}$ as follows:

$$\begin{aligned} w_{ij}^{(c)} &= \sigma_{ij} \cdot w_{ij}^{(d)}, 1 \le i \le n, 1 \le j \le n, \\ \sigma_{ij} &= \sum_{t \in [0, S_{\text{train}})} \left(\tau_{i,t} - \bar{\tau}_i\right)_+ \left(\tau_{j,t} - \bar{\tau}_j\right)_+, \end{aligned} \tag{8}$$

where $(\cdot)_+ = \max\{0, \cdot\}$, $\bar{\tau}_i = \sum_{t \in [0, S_{\text{train}})} \tau_{i,t} / S_{\text{train}}$. Term σ_{ij} is the equivalent of the travel time correlation between segment i and j subtracting the $(\cdot)_+$ operation. This operation is added to remove the "correlation floor" derived from the common free-flow periods. In this paper, Σ is referred to by covariance matrix for convenience.

As shown in Eqn. 8, the compound adjacency matrix is the Hadamard product of the covariance matrix and the Dijkstra adjacency matrix. The incorporation of the covariance term is inspired by the connection between graph convolution and the standard convolutional neural network (CNN) widely utilized in computer vision tasks. As pointed out in [3], when applied to natural images, a graph convolution using the covariance of pixel intensity as proximity measure recovers a standard CNN without any prior knowledge. The covariance term Σ is therefore analogously presumed to offer a more intrinsic measure for traffic proximity. Meanwhile, the Dijkstra matrix is retained to eliminate the unphysical long-range correlations in Σ, such as those induced by citywide daily rush-hour congestion.

Graph Convolution. The regional road network is considered as an undirected graph, with each node representing a particular road segment. As in the prior work [24], a shared graph convolutional network (GCN) is applied on each individual time slice to extract common spatial patterns, and we implement GCN with the spectral formulation [7]. Specifically, we have the normalized graph Laplacian $\mathbf{L}$ and scaled graph Laplacian $\tilde{\mathbf{L}}$ as

$$\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}} \mathbf{W}^{(c)} \mathbf{D}^{-\frac{1}{2}}, \tag{9}$$

$$\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{\max} - \mathbf{I}_n, \tag{10}$$

where $\mathbf{I}_n$ is the identity matrix, $\mathbf{W}^{(c)}$ is the compound adjacency matrix, $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{W}^{(c)}$ with $D_{ii} = \sum_{j=1}^{n} w_{ij}^{(c)}$, and $\lambda_{\max}$ is the greatest eigenvalue of $\mathbf{L}$. The GCN $\mathbf{\Theta}$ is parametrized with Chebyshev polynomials of the scaled graph Laplacian $\tilde{\mathbf{L}}$. Let $\mathbf{X}_{:,t,:}^{(\Theta)} \in \mathbb{R}^{n \times C^{(\Theta_{\text{in}})}}$, $\mathbf{Y}_{:,t,:}^{(\Theta)} \in \mathbb{R}^{n \times C^{(\Theta_{\text{out}})}}$ denote the input and output, then GCN $\mathbf{\Theta}$ works as:

$$\mathbf{Y}_{:,t,j}^{(\Theta)} = \sigma\left(\sum_{m=1}^{C^{(\Theta_{\text{in}})}} \sum_{k=0}^{K-1} \mathbf{\Theta}_{k,m,j} T_k(\tilde{\mathbf{L}}) \mathbf{X}_{:,t,m}^{(\Theta)} + \mathbf{b}_j^{(\Theta)} \right) \in \mathbb{R}^n, \qquad \forall j = 1, 2, \ldots, C^{(\Theta_{\text{out}})} \tag{11}$$

where $T_k(\tilde{\mathbf{L}})$ is the k-th order Chebyshev polynomial, K is the kernel size, $\mathbf{\Theta} \in \mathbb{R}^{K \times C^{(\Theta_{\text{in}})} \times C^{(\Theta_{\text{out}})}}$ denotes the parameter tensor, $\mathbf{b}_j$ is the bias, and σ is an ELU.

3.5 Temporal Gated Convolution

To extract common temporal features, we take advantage of the temporal gated convolution $\mathbf{\Gamma}^{(g)}$ proposed in [24]. As illustrated in Figure 5c, a shared gated 1D convolution is applied on each road segment along the temporal dimension. The 1D convolution maps the input $\mathbf{X}^{(g)} \in \mathbb{R}^{n \times P \times C^{(g_{\text{in}})}}$ to a tensor:

$$[\mathbf{A}\ \mathbf{B}] \in \mathbb{R}^{n \times (P - K_t + 1) \times (2C^{(g_{\text{out}})})} = \mathbf{F}^{(g)} * \mathbf{X}^{(g)} + \mathbf{b}^{(g)}, \tag{12}$$

where $*$ is the 1D-convolution operator, $\mathbf{F}^{(g)} \in \mathbb{R}^{K_t \times C^{(g_{\text{in}})} \times 2C^{(g_{\text{out}})}}$ is the convolution kernel, K_t is the kernel size, P is the length of input temporal sequence, $\mathbf{b}^{(g)}$ is the bias, and $\mathbf{A}$ and $\mathbf{B}$ are of equal size with $C^{(g_{\text{out}})}$ channel. A gated linear unit (GLU) with $\mathbf{A}$ and $\mathbf{B}$ as inputs further adds non-linearity to obtain this layer's output: $\mathbf{\Gamma}^{(g)}\left(\mathbf{X}^{(g)}\right) = \mathbf{A} \odot \sigma(\mathbf{B}) \in \mathbb{R}^{n \times (P - K_t + 1) \times C^{(g_{\text{out}})}}$. "$\odot$" stands for the operator of element-wise multiplication.

3.6 Connection to STGCN

Proposed by Yu et al. [24], the Spatio-Temporal Graph Convolutional Network (STGCN) stacks the spatial graph convolutional layer and temporal gated convolutional layer multiple times in an alternating fashion to jointly capture spatio-temporal dependency. When dropping the volume-feature processing branch (the hybrid branch) and the covariance term in the adjacency matrix, our proposed model reduces to a STGCN model with a single ST-Conv block.

3.7 Model Training

Data Augmentation. Since traffic volume is discrete in nature, in situations of low traffic, even a small fluctuation in the volume channel would considerably affect the model output, making it hard to generalize. To solve this problem, Gaussian noise is added [9] on all volume channels with values below a threshold ϵ_n. Experiment results show that this data augmentation approach significantly mitigates the overfitting.

Optimization. For the multistep traffic forecasting task in this paper, we use the L1 loss function:

$$\mathcal{L} = \frac{1}{n \times S_{\text{train}} \times F} \sum_{\substack{i \in [0,n) \\ t \in [0, S_{\text{train}}) \\ f \in (0,F]}} |\hat{\tau}_{i,t+f} - \tau_{i,t+f}|, \tag{13}$$

where $\hat{\tau}_{i,t+f}$ is the model output and $\tau_{i,t+f}$ is the ground truth.

4 EXPERIMENTS

In this section, we first describe the datasets, compared methods, implementation details, and evaluation metrics. Then we show the effectiveness of the compound adjacency matrix, future-volume feature, and domain transformer. At last, we discuss the model scalability.

4.1 Datasets

Using anonymous user data from Amap, we conduct experiments on two regional networks in the Beijing area as shown in Figure 6: one is around the West 3rd Ring Road with 715 segments, and the other around the East 5th Ring Road with 2907 segments. The respective datasets are denoted by W3-715 and E5-2907. Table 1 depicts the statistics of road segments in the two networks.

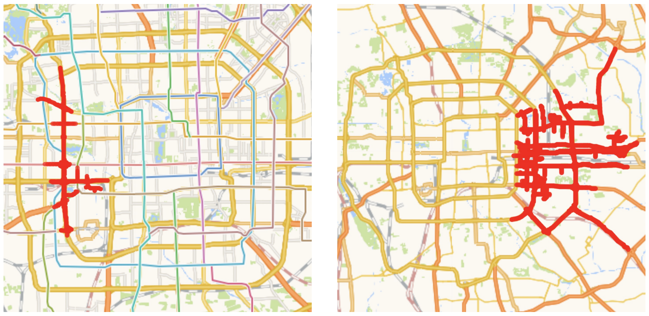
Each dataset contains traffic condition and navigation records from 06:00 to 22:00, and the time span is from December 24, 2018 to April 21, 2019 with holidays removed (ten weeks in total). The previous eight weeks are used as training data, and the remaining two weeks as testing data.

4.2 Compared Methods

We compare our proposed architecture with the following two methodological categories:

Benchmark Models.

- Historical Average (HA): Historical average predicts travel time with mean value over time slots at the same previous relative position.
- Linear Regression (LR): Linear regression is a basic regression model.
- Gradient Boosting Regression Tree (GBRT): GBRT is a widely-used boosting model. We set the number of trees at 50, with a maximum depth of 6.



(a) Road segments of W3-715 (b) Road segments of E5-2907

Figure 6: Spatial distribution of the regional road networks.

Table 1: Statistics of road segments in the network, including the total road segment number, the average length (meter) of road segments, and the average traffic volume (veh/min) of road segments corresponding to each road class.

Dataset	Road class	Num.	Avg. len. (meter)	Avg. vol.
W3-715	Freeway	7	132	7.0
	Highway	34	176	3.2
	Expressway	186	163	11.8
	Major Road	488	80	2.4
	Total	715	107	4.9
E5-2907	Freeway	135	334	9.9
	Highway	163	178	2.6
	Expressway	427	348	12.4
	Major Road	2182	97	2.3
	Total	2907	150	4.1

Table 2: Statistics of high volume road segments, including the number of road segments, the percentage of congested periods (C), the percentage of non-recurring congested periods (NRC), and the average traffic volume (veh/min).

Dataset	Road class	Num.	Pct. (C)	Pct. (NRC)	Avg. vol.
W3-715	Freeway	0	/	/	/
	Highway	0	/	/	/
	Expressway	138	19.6%	6.5%	14.0
	Major Road	1	22.9%	5.4%	10.2
E5-2907	Freeway	70	7.8%	3.4%	12.1
	Highway	5	11.5%	8.5%	15.7
	Expressway	235	15.7%	7.3%	18.0
	Major Road	1	39.3%	21.3%	10.2

- Multi-Layer Perceptron (MLP): MLP is a fully connected multi-layer neural network. We use three layers, and the hidden unit of each layer is 64.
- Sequence-to-Sequence (Seq2Seq): Seq2Seq models use the encoder-decoder architecture and have been widely applied

to language modeling and time-series forecasting. We use two layers, and the hidden unit of each layer is 200.

- STGCN: Original STGCN used multiple ST-Conv blocks to boost the performance. On our dataset, however, one block is found sufficient to achieve a similar level of accuracy. We thus use STCGN with a single ST-Conv block as the baseline.

Variant Models for Ablation Study.

- STGCN (Im): Improved STGCN uses a compound adjacency matrix as opposed to the Dijkstra matrix.
- H-STGCN (1): H-STGCN (1) uses an input volume tensor $\mathbf{V}$ with all elements set to one (1).

4.3 Implementation Details

In all models, we use data from the previous six time slots (30 min) as input, and predict travel time for the next hour. The shared and segmentwise 1×1 convolutions in domain transformer both have 16 filters. The graph convolution has 64 filters. The temporal gated convolutional layers $\mathbf{\Gamma}_1^{(g)}$, $\mathbf{\Gamma}_2^{(g)}$, $\mathbf{\Gamma}_3^{(g)}$, $\mathbf{\Gamma}_4^{(g)}$ have 64, 128, 64, and 64 filters. The last fully connected layer outputs 12 values, corresponding to the forecasting period. We set $\sigma^2 = 3$ km^2, $\epsilon = 0$ (no spatial cutoff) in Equation (7), and the threshold of noise injection $\epsilon_n = 3$. We use Adam optimizer [13] with initial learning rate 0.001 and decay rate 0.98. We implement the traditional benchmark models with scikit-learn, and the neural network models on TensorFlow. The training and inference of neural networks are conducted on 4 NVIDIA GPUs with 16 GB memory.

4.4 Evaluation Metrics

To better verify the effectiveness of H-STGCN, we select two additional subtestsets based on the following considerations. First, to showcase the extra predictive power brought by ideal future volume, we consider only segments with average historical traffic volume above 10 veh/min (high-volume segments). Secondly, we focus only on congested time periods, the non-trivial part of traffic forecasting. The congestion speed threshold is set according to road class: 30 km/h for freeway, 20 km/h for highway and expressway, and 12 km/h for major road. We further define non-recurring congestion as the one with travel speed constantly below half of its historical average. Thirdly, to examine forecasting performance over the full lifecycle of congestion, we extend a (non-recurring) congested period by an hour in each direction to include both the formation and dissipation stages of congestion. To summarize, we have three types of test sets in the experiment:

- Full test set as described in Section 4.1.
- Test set comprising data from high-volume segments in the congested periods, denoted by suffix (C).
- Test set comprising data from high-volume segments in the non-recurring congested periods, denoted by suffix (NRC).

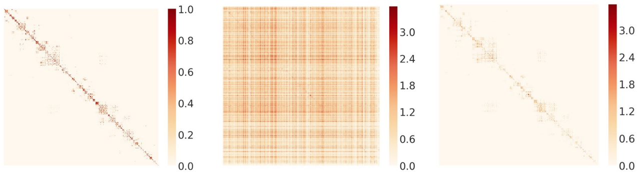
Table 2 shows statistics of the last two test sets. We use the mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE) as the evaluation metrics.

4.5 Performance Comparison

Table 3 outlines the performance of our model as compared to the competing methods. H-STGCN significantly outperforms the various benchmarks in all metrics, especially for the prediction of non-recurring congestion. In this section, we study the effectiveness of each proposed module in H-STGCN.

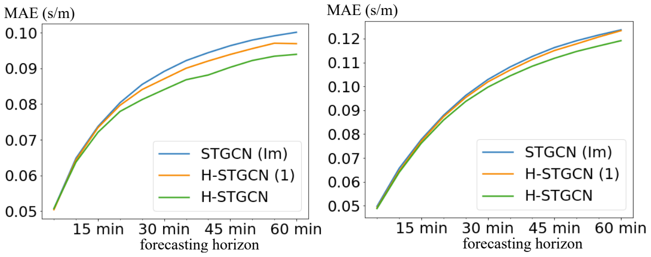
Compound Adjacency Matrix. We compare the performance of STGCN and STGCN (Im). As shown in Table 3, STGCN (Im) achieves a lower MAE and MAPE on W3-715, and a lower MAE, MAPE, and RMSE on E5-2907, validating the effectiveness of the compound adjacency matrix. Figure 7 shows an example from E5-2907, which illustrates the connections among different adjacency matrices as described in Section 3.3.

Future-Volume Feature and Domain Transformer. First, as shown in Table 3, H-STGCN delivers consistently superior performance compared to STGCN (Im), demonstrating the remarkable advantage brought by the utilization of future-volume data. Secondly, owing to the segment-wise structure of domain transformer, H-STGCN gains an edge over STGCN (Im) in terms of representation power. To eliminate this influence factor and assess the importance of the future-volume feature, we further compare H-STGCN to H-STGCN (1). As indicated by the results on test set (C) and test set (NRC), the volume feature substantially enhances model performance on congestion forecasting. Lastly, Figure 8 shows that, as the forecasting horizon lengthens, the volume feature becomes the most dominant contributor to error reduction.



(a) (b) (c)

Figure 7: Weighted adjacency matrices in E5-2907. The color represents the normalized value of $\lg(w_{ij} + 1)$. (a) Dijkstra adjacency matrix. (b) Covariance matrix. (c) Compound adjacency matrix.



(a) Dataset W3-715 **(b) Dateset E5-2907**

Figure 8: Comparison over the forecasting horizon on test set (NRC).

To explain the intuition behind H-STGCN, we show an example regarding the prediction of non-recurring congestion, as depicted in Figure 9. During the congestion formation stage between 17:30 and 18:00, the multistep-ahead travel-time prediction from H-STGCN

Table 3: Comparison with baselines on full test set, test set (C), and test set (NRC). Evaluation metrics include MAE (s/m), MAPE (%), and RMSE (s/m).

Dataset	Model	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
		Test set (Full)			Test set (C)			Test set (NRC)		
W3-715	HA	0.03886	20.73	0.09285	0.07040	34.36	0.10479	0.10303	39.39	0.16486
	LR	0.03334	16.58	0.08467	0.06469	33.52	0.10582	0.09080	39.57	0.14768
	GBRT	0.03264	16.10	0.08409	0.06236	32.08	0.10479	0.09085	39.35	0.14945
	MLP	0.03272	16.57	0.08269	0.06096	31.84	0.10190	0.08733	38.71	0.14427
	Seq2Seq	0.03231	15.81	0.08252	0.06033	28.79	0.10174	0.08599	34.04	0.14467
	STGCN	0.03219	16.01	0.08182	0.05975	30.48	0.09901	0.08599	38.72	0.14004
	STGCN (Im)	0.03200	15.83	0.08196	0.05965	29.96	0.09995	0.08539	36.71	0.14197
	H-STGCN (1)	0.03138	15.52	0.08099	0.05804	29.14	0.09806	0.08373	34.71	0.14012
	H-STGCN	**0.03114**	**15.36**	**0.08045**	**0.05711**	**28.34**	**0.09644**	**0.08124**	**33.22**	**0.13711**
E5-2907	HA	0.04615	21.22	0.11405	0.09786	44.95	0.16729	0.13161	46.96	0.21769
	LR	0.04096	17.03	0.10732	0.08229	41.69	0.14270	0.10747	47.01	0.18192
	GBRT	0.04032	16.61	0.10680	0.07997	39.51	0.14465	0.10657	44.68	0.18593
	MLP	0.04031	17.16	0.10547	0.08025	41.26	0.14229	0.10580	45.84	0.18236
	Seq2Seq	0.04087	17.52	0.10631	0.08413	41.72	0.14703	0.10981	44.81	0.18722
	STGCN	0.03984	16.95	0.10296	0.07561	38.13	0.13677	0.09966	43.28	0.17563
	STGCN (Im)	0.03957	16.85	0.10221	0.07498	37.80	0.13579	0.09843	42.74	0.17399
	H-STGCN (1)	0.03870	16.31	0.10095	0.07380	37.07	0.13455	0.09750	42.32	0.17257
	H-STGCN	**0.03861**	**16.28**	**0.10067**	**0.07254**	**36.31**	**0.13308**	**0.09528**	**40.82**	**0.17030**

(1) shows a notable time lag compared to the ground truth. In contrast, H-STGCN, when fed with the ideal-future-volume data, is able to accurately forecast the congestion even 30 minutes in advance. We understand this observation as follows. The curve of $v_{i,t,3}$, which represents an approximation of the traffic volume 15 minutes later, rapidly increases at around 17:15. Further, given the fact that a navigation engine is only aware of the trips that have started already, the actual future traffic volume would be even greater than the ideal one. Therefore, the rise of $v_{i,t,3}$ is a prominent indicator of strong upcoming traffic flux, which in turn enables H-STGCN to foresee the future congestion even without a historical reference.

4.6 Model Scalability

The model inference time for W3-715 and E5-2097 is less than 100 ms. To balance the inference efficiency and forecasting performance for real-world application, we partition a city-wide road network into sub-networks with at most a few thousand segments, by minimizing the number of congested boundary links [12]. Then a separate model is trained and deployed for each sub-network.

5 RELATED WORK

Traffic prediction has been studied for decades, and existing methods chiefly fall into two categories: the theory-driven approach and the data-driven approach. In the former category [2, 4, 21], a simulation system is built according to the theory of traffic dynamics and is composed of several interacting modules such as a routing model, driving behavior model, and queueing model. Given all origin-destination pairs, a simulator is able to forecast future traffic. For the data-driven approach, shallow machine learning models, including Bayesian network [19], support vector regression, random forest, gradient boosting regression tree etc., were thoroughly investigated. However, due to limited representation power, such elementary models cannot yield the prospective outcomes.

Recently, a host of deep-learning-based approaches have been attempted and achieved considerable improvements over traditional benchmarks. To capture the spatial dependency, graph convolution structures have been subject to experimentation and achieved notable improvements [15, 24]. To further extract global spatial correlations, Fang et al. [8] proposed the use of a non-local correlated mechanism. To model the nonlinear temporal dependency, Li et al. [15] applied the encoder-decoder architecture. Yu et al. [24] considered time series of traffic speed as a one-dimensional image and instead adopted the convolutional network.

Amongst the various traffic scenarios, non-recurring congestion is particularly difficult to predict due to the lack of contextual information. To tackle this issue, several studies suggested the use of weather status, tweets, road structure features, points of interest, or crowd map queries as auxiliary information and achieved decent performance [6, 10, 14, 16, 26]. Nonetheless, the spatial resolution of forecasting is insufficient for critical real world application.

6 CONCLUSION

In this paper, we propose a novel deep architecture for travel time forecasting, the Hybrid Spatio-Temporal Graph Convolutional Network (H-STGCN), which features the utilization of intended-traffic-volume data. We design the domain transformer to couple this

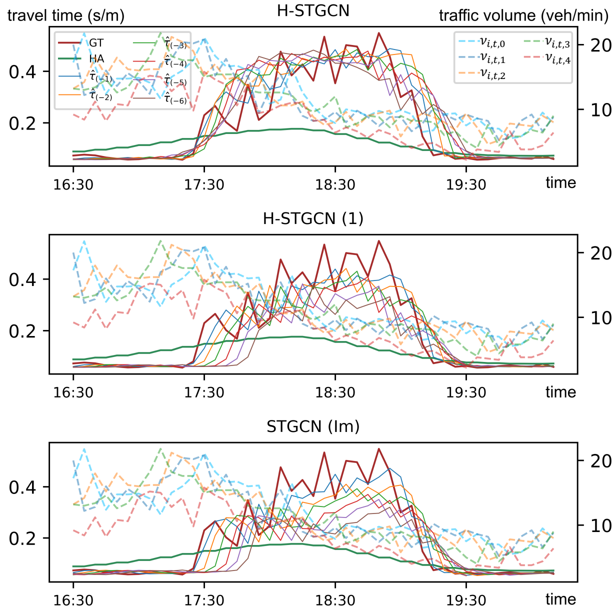


Figure 9: Example of travel time prediction on non-recurring congestion. Case studied is from a freeway segment on April 16, 2018. GT denotes the ground truth, HA denotes the historical average, $\hat{\tau}_{(-f)}$ is the f-step ahead prediction, and $v_{i,t,f}$ is the ideal future traffic volume corresponding to the time slot f-step later.

heterogeneous modality of traffic volume. We propose a compound adjacency matrix to capture the innate nature of traffic proximity. Experiments carried out on real-world datasets show that H-STGCN achieves remarkable improvement over the benchmark methods, especially for the prediction of non-recurring congestion. Finally, this architecture exemplifies a novel formalism to embed the knowledge of physics in a data-driven model, which can be readily applied to general spatio-temporal forecasting tasks.

REFERENCES

[1] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F Werneck. 2016. Route planning in transportation networks. In *Algorithm engineering*. Springer, 19–80.

[2] Moshe Ben-Akiva, Michel Bierlaire, Haris Koutsopoulos, and Rabi Mishalani. 1998. DynaMIT: A simulation-based system for traffic prediction. In *DACCORD Short Term Forecasting Workshop*. Delft, The Netherlands, 1–12.

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2014. Spectral networks and locally connected networks on graphs. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

[4] Wilco Burghout. 2004. *Hybrid microscopic-mesoscopic traffic simulation modelling*. Ph.D. Dissertation. PhD thesis, Dept of Infrastracture, Royal Institute of Technology, Stockholm, Sweden.

[5] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

[6] Daniel J Dailey and Trepanier Ted. 2006. *The use of weather data to predict non-recurring traffic congestion*. Technical Report. Technical report to Washington State Transportation Commission, Washington State Department of Transportation, University of Washington TransNow, and Federal Highway Administration.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*. 3844–3852.

[8] Shen Fang, Qi Zhang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2019. Gstnet: Global spatial-temporal network for traffic flow prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*. 10–16.

[9] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[10] Jingrui He, Wei Shen, Phani Divakaruni, Laura Wynter, and Rick Lawrence. 2013. Improving traffic prediction with tweet semantics. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. 1387–1393.

[11] Serge P Hoogendoorn and Piet HL Bovy. 2001. State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 215, 4 (2001), 283–303.

[12] George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing* 20, 1 (1998), 359–392.

[13] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

[14] Arief Koesdwiady, Ridha Soua, and Fakhreddine Karray. 2016. Improving prediction with weather information in connected cars: A deep learning approach. *IEEE Transactions on Vehicular Technology* 65, 12 (2016), 9508–9517.

[15] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

[16] Binbing Liao, Jingqing Zhang, Chao Wu, Douglas McIlwraith, Tong Chen, Shengwen Yang, Yike Guo, and Fei Wu. 2018. Deep Sequence Learning with Auxiliary Information for Traffic Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[17] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. 2009. Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. 352–361.

[18] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, Fei-Yue Wang, et al. 2015. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intelligent Transportation Systems* 16, 2 (2015), 865–873.

[19] Alessandra Pascale and Monica Nicoli. 2011. Adaptive Bayesian network for traffic flow prediction. In *2011 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 177–180.

[20] Alexey Tsymbal. 2004. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin* 106, 2 (2004), 58.

[21] Eleni I Vlahogianni. 2015. Computational intelligence and optimization for transportation big data: challenges and opportunities. In *Engineering and Applied Sciences Optimization*. Springer, 107–128.

[22] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17, 4 (2007), 395–416.

[23] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 2496–2505.

[24] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

[25] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, and Cheng Chen. 2011. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* 12, 4 (2011), 1624–1639.

[26] Chuanpan Zheng, Xiaoliang Fan, Chenglu Wen, Longbiao Chen, Cheng Wang, and Jonathan Li. 2019. DeepSTD: Mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems* (2019).