Early Forecasting of the Impact of Traffic Accidents Using a Single Shot Observation

Guangyu Meng[*] Qisheng Jiang[†] Kaiqun Fu[‡] Beiyu Lin[§] Chang-Tien Lu[¶] Zhqian Chen[‖]

Abstract

Predicting and measuring the impact of traffic collisions is crucial for Intelligent Transportation Systems (ITS). Numerous works in this field have successfully applied graph neural networks to ITS. Existing research on graph neural networks mainly relies on the graph Fourier transform, assuming neighborhood homophily. The homophily assumption, on the other hand, makes it difficult to define abrupt signals such as traffic accidents. Our research proposes an abrupt graph wavelet network (AGWN) for forecasting the durations of traffic incidents using a single shot. To begin, graph wavelet (GW) is theoretically examined in terms of linear separability in comparison to graph Fourier (GF), demonstrating its advantage in modeling abrupt graph signals. Sensitivity analysis and admissibility conditions are utilized to further study the behavior of GW in abrupt graph signals, justifying the use of zero sum function as wavelet kernel. The synthetic data results support our proposed wavelet kernel's effectiveness in modeling a variety of abrupt signals, while real-world trials demonstrate that our method significantly outperforms baseline models in forecasting the duration of an accident impact.

1 Introduction

A one-minute increase in the duration of traffic collisions resulted in a loss of $65 [1]. Due to the magnitude of the social and economic consequences connected with traffic accidents, the focus of research has turned to Intelligent Transportation Systems (ITS), with the impact of traffic accidents being a main research priority [2]. In general, traffic accidents result in slower speeds, longer travel times, and higher vehicular congestion. Failure to anticipate the impact of traffic accidents may have more serious repercussions. A massive deployment of traffic speed sensors have been widely deployed during the previous decade, giving ubiquitous access to traffic speed data and accident reports. In this paper, we focus on the *early forecast of traffic accident impact* utilizing a single shot immediately following the accidents. More precisely, time duration is the metric used to determine the degree of accident impact. Early forecasting with a single shot is desirable because (1) without early forecasting, drivers may spend excessive time in traffic jams or in secondary accidents while primary accidents are not cleared, and (2) single shot significantly reduces the time for data collection and model prediction. On the other hand, it is challenging since only one shot can be utilized for this task.

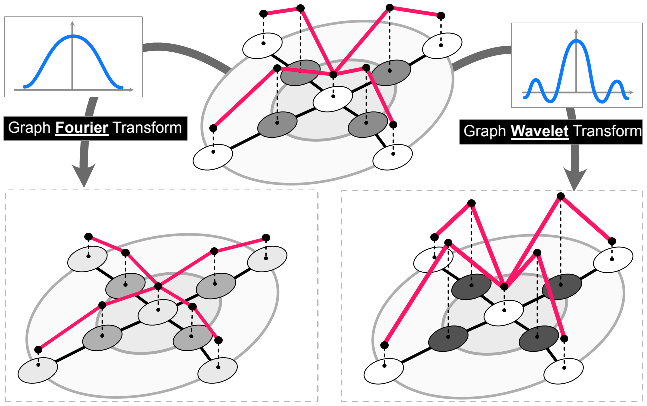


Figure 1: The graph Fourier transform smooths out an abrupt signal, whereas the graph wavelet transform accentuates it.

With the latest success of extending deep learning approaches from regular grids to structured data, graph neural networks (GNN) [3, 4, 5] has become a dominating research methodology in network modeling and has numerous applications in traffic [6, 7, 8]. However, current studies suffer from several shortcomings: **(1) Existing graph Fourier-based methods cannot handle abrupt signal:** Most existing graph neural networks rely on graph Fourier, which result in low-pass filtering [9, 10, 11, 12]. When a graph has an abrupt signal in a few local places, the graph Fourier basis averages the neighborhood pattern throughout the graph, ignoring the uncommon and abrupt signal. **(2) Existing neural networks based on graph wavelets are incapable of recognizing the abrupt signal:** In theory, the graph wavelet transform can better identify abrupt signals then graph Fourier [13], as illustrated in Figure 1. However, existing graph wavelet neural network [14, 15] violate the admissibility condition, making abrupt graph signals difficult to model [16]. **(3) Insufficient theoretical and empirical**

[*] University of Notre Dame, gmeng@nd.edu
[†] Shanghai Tech University, jiangqsh@shanghaitech.edu.cn
[‡] South Dakota State University, kaiqun.fu@sdstate.edu
[§] University of Nevada, beiyu.lin@unlv.edu
[¶] Virginia Tech, ctlu@vt.edu
[‖] Mississippi State University, zchen@cse.msstate.edu

evidence to justify graph wavelet design on abrupt graph signals: Although graph wavelet can handle abrupt graph signals, theoretical research on kernel selection is not explicitly studied with graph neural networks, particularly in traffic accident scenarios.

To solve these issues, we examine the linear separability of graph wavelet (GW) and graph Fourier (GF) transforms, demonstrating that GW outperforms GF in recognizing abrupt signals. Then, using the admissibility condition, graph wavelets with different kernels are compared, separating abrupt from non-abrupt kernels. As a result, we propose a graph neural network with an appropriately sized kernel for modeling abrupt graph signals. Finally, the early forecasting of traffic accident impacts is investigated using both synthetic and real-world data. Our contributions are summarized below:

- **Compare graph wavelet and graph Fourier in theory for the purpose of detecting abrupt graph signals.** Fisher score is used to evaluate graph Fourier and graph wavelet transforms on abrupt graph signals.
- **Analyze graph wavelet kernels for identifying abrupt graph signals.** By admissibility condition, graph wavelets are divided into abrupt and non-abrupt kernels. Then, based on our theoretical analysis, we deduce the advantage of abrupt kernels for abrupt signal modeling.
- **Develop a graph wavelet neural network and undertake extensive experimentation.** To characterize the abrupt graph signal, multi-scale graph wavelets are combined with graph neural networks. Extensive experiments are conducted on synthetic data and real-world traffic accident datasets.

2 Related Work

2.1 Analysis of the Effects of Traffic Accidents: The uses of traditional statistical approaches have demonstrated their usefulness in predicting the duration of traffic accidents. In the last decade, traffic control centers in numerous cities and roads have adopted Traffic Accident Management Systems (TIMS) to mitigate the impact of traffic accidents on traffic conditions [17]. Numerous data mining and machine learning techniques have been used to quantify and predict the duration of traffic accidents. The majority of traffic evaluations using graph neural networks (GNNs) focus on traffic prediction, whereas GNNs disregard accident impact modeling [6, 7, 8]. The use of graph neural networks in conjunction with recurrent neural networks to categorize traffic has increased significantly in recent years [8, 7, 18, 6]. They do, however, require an input sequence to forecast the destination at a specific time point. Our research establishes an efficient graph neural network with a single input shot.

2.2 Graph Neural Networks: Numerous graphs and geometric convolution approaches for modeling graph data have been proposed recently [19, 4, 3]. Their theory is based on Fourier analysis of graphs[20]. As a result, graph convolutional neural networks (GNNs) successfully adapt the highly effective convolutional neural networks used to represent Euclidean data to graph-structured data [4, 3]. However, those studies mainly rely on the graph Fourier transform, which may result in significant estimation errors when dealing with abrupt signals. Another effective approach for modeling complex signals on a graph is the graph wavelet. Existing graph convolutional networks with wavelet kernels, on the other hand, are unable to identify abrupt signals due to their kernel choices [14, 15]. To solve this issue, we conduct a theoretical analysis of the various kernels and suggest a model with the most appropriate kernel.

3 Problem Formulation: Single Shot based Prediction

Our objective is to provide an early estimate of the impact duration in traffic accidents using a single shot taken immediately after it occurs. The transportation sensor network is defined as $\mathbf{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{A}\}$, where $\mathbf{V}$ stands for a set of N nodes which are the traffic sensors, $\mathbf{E}$ represents connectivity among the sensors (e.g., geographical proximity), and $\mathbf{A} = \{a_{ij}\}^{N\times N}$ is an adjacent weighted matrix of these connectivity. The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}, \mathbf{L} \in \mathrm{R}^{N\times N}$ ($\mathbf{D}$ is degree matrix with entries $d_{ii} = \sum_k a_{ik}$) Therefore, **traffic accidents** can be formulated as **abrupt graph signals** which is defined as below:

DEFINITION 3.1. (ABRUPT GRAPH SIGNAL IN TRAFFIC) *Let a graph signal* $\mathbf{X} = \{\mathbf{X}_a, \mathbf{X}_s\}$. $\mathbf{X}_s$ *is a smooth neighborhood, where all nodes have similar value. While* $\mathbf{X}_a$ *is an abrupt neighborhood, where most nodes are similar and contains a few nodes that have significantly different values (i.e., signals of a traffic accident). Let* σ *denote the variance,* κ *represents node amount, and* μ *being mean value.* $\mathbf{X}$ *is abrupt graph signal as along as it satisfies (1)* $\sigma_a \gg \sigma_s \approx 0$, *(2)* $\kappa_a \ll \kappa_s$, *and (3)* $|\mu_a - \mu_s| \gg 0$.

Remark: (2) and (3) in Theorem 3.1 are assumed for transportation networks, as regular traffic speeds are constrained by traffic flow and regulation. As a result, when an accident occurs, its measures, such as speed and volume (e.g., 0 miles per hour for severe incidents or a huge volume of vehicles in traffic jams), are significantly different from those of normal traffic flow. Additionally, the number of sensors that are impacted by accidents is always quite small.

Accordingly, the early forecast task can be written as a regression task:

DEFINITION 3.2. (EARLY FORECAST OF ACCIDENTS)

$$\mathbf{Y} = f(\mathbf{G}, \mathbf{X}), \tag{3.1}$$

where $\mathbf{G}$ *represent the sensor network,* $\mathbf{X} \in R^{N\times F}$ *represent an one shot observation on* $\mathbf{G}$ *immediately following the* *accident. F denotes the signal dimension, and Y is the duration time of traffic accidents, i.e., from when a traffic accident happens to when it is cleared. f is the function to learn.*

4 Abrupt Graph Wavelet Networks (AGWN)

In this section, the task of early forecast on the impact of traffic accidents is described. The superiority of graph wavelet transform is analyzed theoretically. Then a graph neural network with a graph wavelet is presented.

4.1 Analysis on Graph Fourier and Graph Wavelet.

The majority of GNNs employ the graph Fourier transform with a low-pass filtering flavor, and its over-smoothing issue is well-known. [11, 9, 21, 22, 23, 24, 25, 26, 27, 28, 29]. This precludes the use of graph Fourier-based techniques for modeling traffic accidents. The rationale for this is that when low frequencies dominate the data (e.g., smooth traffic speed and volume), the model readily fits the low frequencies and ignores the high frequencies (e.g., traffic accidents), which aids the model in achieving overall accuracy. On the contrary, graph wavelet is highly sensitive to high frequency, particularly when a kernel satisfying the admissibility condition is used [16]. In the following analysis, graph Fourier and graph wavelet will be theoretically contrasted in the presence of an abrupt signal. Specifically, linear separability is used to determine their capacity in distinguishing abrupt signals. The phrase ”linear separability” refers to a group's capacity to be split by linear lines in planes. Linear discriminant analysis (LDA) [30] is used to learn the optimal transform. Similarly, linear separability can be used to assess the ability of the GW or GF transform to distinguish abrupt signals.

Algorithm 1: Wavelet Transform Function

```
  Input: adjacent matrix of a graph $\mathbf{A} \in \mathrm{R}^{N\times N}$,
         predefined distance threshold $\tau \in \mathrm{R}$, a
         continuous 1D function $h$
  Output: Wavelet $\Psi \in \mathrm{R}^{N\times N}$
1 $S \leftarrow$ Shortest Path Matrix $\in \mathrm{R}^{N\times N}$, each entry $s_{i,j}$
    being the shortest path between node $i$ and $j$
2 // discretize h
3 $\bar{h} \in \mathrm{R}^{\tau\times 1} \leftarrow$ Divide h into $\tau$ groups along x-axis, and
    take mean of each group
4 for $v_i \in V$ do
5     $G_V \in \mathrm{R}^{\tau\times 1} \leftarrow$ Group the other nodes by distances
        to $v_i$ and filtered by distance threshold $\tau$ (groups
        whose distances are larger than $\tau$ will set to be zero)
6     // element-wise division
7     $\Psi[i] \in \mathrm{R}^{N\times 1} \leftarrow \bar{h}/G_V$
```

The graph spectral domain calculation is based on the Laplacian matrix ($\mathbf{L} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}$) and it has a complete set of orthonormal eigenvectors $\mathbf{U} = \{u_i\}, i \in \{1, 2, ...n\}$, with corresponding non-negative real eigenvalues $\mathbf{\Lambda} = \{\lambda_i\}, i \in \{1, 2, ...n\}$. The graph wavelet theory is derived from the graph Fourier transform, which uses spectral graph theory [16] to define Fourier modes and frequencies on graphs. Graph Fourier transform is defined as $\hat{\mathbf{X}} = \mathbf{U}^\top\,\mathbf{X}$, while inverse graph Fourier transform is $\mathbf{X} = \mathbf{U}\,\mathbf{X}$ [16]. Similarly, with a graph wavelet kernel $\Psi \in \mathrm{R}^{N\times N}$ Graph wavelet transform: $\hat{\mathbf{X}} = \Psi^{-1}\mathbf{X}$, while graph wavelet inverse transform: $\mathbf{X} = \Psi\hat{\mathbf{X}}$, where the design of Ψ is based on the geodesic or shortest path distance $d_{\mathbf{G}}(i, j)$ between node i and j. Define $j \in \partial\mathcal{N}(i, \delta)$ to be a node set such that $d_{\mathbf{G}}(i, j) \le \delta$ where δ is a predefined threshold. Then the wavelet function $\Psi_{s,i} : \mathbf{V} \to R$ at scale s and center vertex $i \in \mathbf{V}$ can be written as:

$$\Psi_{s,i}(j) = \frac{\phi_{s,\delta}}{|\partial\mathcal{N}(i,\delta)|}, \forall j \in \partial\mathcal{N}(i,\delta), \tag{4.2}$$

where $\phi_{s,\delta}$ is a wavelet kernel function, and $\phi_{s,\delta} \neq 0$ iff $\delta \le s$. Thus, each wavelet is constant across all vertices $j \in \partial\mathcal{N}(i, \delta)$ that are the same distance from the center vertex i, and the value of the wavelet at the vertices in $\partial\mathcal{N}(i, \delta)$ depends on the distance δ (see details in IV-C of [20]). The wavelet calculation is described in Algorithm 1: Given a center vertex v_i, a given kernel function h is discretized into τ groups and the mean is calculated for each group. The wavelet value at each neighbor vertex equals to the ratio of the mean of its group to the volume of the group (Line 7). The Mexican hat wavelet, which has an integral of zero, is a typical wavelet kernel that will be employed in our design [31]. As a result, a theorem is presented below.

THEOREM 4.1. (LINEAR SEPARABILITY COMPARISON) *Let $\mathbf{X}$ be an abrupt graph signal by Def. 3.1, and J is the linear separability measured by the Fisher score, then we have:*

$$J^{GW}(\mathbf{X}) > J^{GF}(\mathbf{X}),$$

where $J^{GW}(\mathbf{X})$ and $J^{GF}(\mathbf{X})$ denote linear separability level of graph wavelet and graph Fourier transform on $\mathbf{X}$, respectively.

Proof. Linear discriminant analysis is used to examine graph Fourier (GF) and graph wavelet (GW) data (LDA). As a result, the improved transform should make it easier to distinguish between node representations of different classes (i.e., $\mathbf{X}_a$, $\mathbf{X}_s$). The effectiveness of a transform can be evaluated by linear separability with the Fisher score [30]:

$$J(\mathbf{X}) = S\left(\mathbf{X}_a, \mathbf{X}_s\right) = \frac{\left(\mu_a - \mu_s\right)^2}{\sigma_a + \sigma_s}, \tag{4.3}$$

where S denotes the Fisher score. Equation 4.3 can be treated as linear separability on raw data without any transform. The value of abrupt neighborhood is significantly different from that of smooth neighborhood, i.e., $\mu_a \ll \mu_s$ or $\mu_a \gg \mu_s$. By Def. 3.1, $\sigma_s \ll \sigma_a$. Formally, GF's Fisher score can be written as:

$$J^{GF}(\mathbf{X}) = S\left(\mathbf{X}_a^{GF}, \mathbf{X}_s^{GF}\right) = \frac{\left(\mu_a^{GF} - \mu_s^{GF}\right)^2}{\sigma_a^{GF} + \sigma_s^{GF}},$$

where superscript $(\cdot)^{GF}$ means applying GF. Typical graph Fourier methods with low-pass filtering equals to average one node and its neighbors [10], so the mean values of abrupt and smooth groups move to each other, and their different become smaller after applying GF, i.e., $\left(\mu_a^{GF} - \mu_s^{GF}\right)^2 < \left(\mu_a - \mu_s\right)^2$. On average, part of each group move closer to each other, so scatter range of both of two group increases and thereby their variances increase, i.e., $\sigma_a < \sigma_s^{GF}$, and $\sigma_s < \sigma_s^{GF}$. Therefore, $\sigma_a^{GF} + \sigma_s^{GF} > \sigma_a + \sigma_s$, and we have:

$$J^{GF} = \frac{\left(\mu_a^{GF} - \mu_s^{GF}\right)^2}{\sigma_a^{GF} + \sigma_s^{GF}} < \frac{\left(\mu_a - \mu_s\right)^2}{\sigma_a + \sigma_s} = J \tag{4.4}$$

Similarly, graph wavelet satisfying admissibility condition [16] is evaluated with Fisher score (see more about admissibility in 4.2). According to admissibility condition, integral of the kernel function (e.g., Mexico hat) is zero, and abrupt and smooth neighborhood respectively: **(1) For smooth neighborhood:** filtered smooth neighborhood by GW is equal to the product of the kernel's integral and a constant vector (smooth signal), which leads to near zero. Therefore, the mean of the smooth neighborhood is approximately zero: $\mu_s^{GW} \approx 0$. We also have $\sigma_s^{GW} \approx 0$ by Def. 3.1. **(2) For abrupt neighborhood:** graph wavelet highlights the difference between abrupt and smooth neighborhood, so $\mu_a^{GW} = \alpha|\mu_a - \mu_s|$, where $\alpha \in \mathrm{R}^+$ is a weight and varies with the position of abrupt value in the neighborhood. It is easy to get $\sigma^{GW} = \alpha^2\sigma$. So we have the following transformtion for both (1) smooth and (2) abrupt neighborhood:

$$\begin{aligned}
J^{GW} =& \frac{\left(\mu_a^{GW} - \mu_s^{GW}\right)^2}{\sigma_a^{GW} + \sigma_s^{GW}} && \text{(definition)}\\
\approx& \frac{\left(\mu_a^{GW}\right)^2}{\sigma_a^{GW}} && (\mu_s^{GW} \approx 0, \sigma_s^{GW} \approx 0)\\
=& \frac{\left[\alpha(\mu_a - \mu_s)\right]^2}{\alpha^2\sigma_a} && \text{(apply graph wavelet)}\\
=& \frac{\left(\mu_a - \mu_s\right)^2}{\sigma_a} && (\text{cancel out } \alpha^2)\\
>& \frac{\left(\mu_a - \mu_s\right)^2}{\sigma_a + \sigma_s} = J && (\sigma_s > 0).
\end{aligned}$$

Therefore, combining Equation 4.4, we have:

$$J^{GW} > J > J^{GF},$$

which means abrupt graph signals filtered by graph wavelet are more linear separable than that by graph Fourier. □

4.2 Configuration of Graph Wavelet. This subsection will compare abrupt graph wavelets (AGW) with non-abrupt graph wavelets (N-AGW) theoretically, as well as evaluate the essential condition that provides abruptness awareness. Then, we suggest using a Mexican hat kernel in graph wavelet to enable the model to recognize a adrupt signal. This is accomplished by ensuring that the integral of the function is zero, in contrast to existing graph neural networks with N-AGW (e.g., exponential function) [14, 15]. We will use a typical wavelet kernel function, i.e., the Mexican hat wavelet [31], to illustrate the necessity of the admissibility condition. The Mexican hat is implemented with a mother wavelet:

$$\phi(t) = \frac{2}{\sqrt{3\sigma}\pi^{\frac{1}{4}}}(1 - (\frac{t}{\sigma})^2)e^{-\frac{t^2}{2\sigma^2}}, \quad \int_0^\infty \phi(t) = 0, \tag{4.5}$$

where σ means standard deviation and t is the input distance from the center. There are two reasons that support this design:

Reason 1: Sensitive for Abrupt Signal. Take Mexico hat as an example, AGW has distinguishable behaviors on smooth signal and abrupt signals.

- **For smooth neighborhood**: For any neighbor of node i, i.e., $\mathbf{X}(j) \in \partial\mathcal{N}(i,\delta)$ satisfies:

  $$|\,\mathbf{X}(i) - \mathbf{X}(j)| < \epsilon, \forall j \in \partial\mathcal{N}(i,\delta),$$

  where ϵ is a small value, and node j is in the neighborhood of node i. Therefore, we have the result after applying AGW transform in the neighborhood of node i:

  $$\Psi_{s,i}^{AGW}\,\mathbf{X}(i) = \sum_{\delta=0}^{s}\sum_{j\in\partial\mathcal{N}(i,\delta)} \frac{\phi_{s,\delta}}{|\partial\mathcal{N}(i,\delta)|}\mathbf{X}(j) \approx \mathbf{X}_j \sum_{\delta=0}^{s}\phi_{s,\delta} = 0,$$

  which means that the outcome of AGW transform is zero.

- **For abrupt neighborhood**: By Def. 3.1, the neighborhood consists of a few abrupt part and majority is smooth, satisfying:

  $$\begin{aligned}
  &abrupt: |\,\mathbf{X}(i) - \mathbf{X}(j\prime)| \gg \epsilon, && \text{if } \exists j\prime \in \partial\mathcal{N}(i,\delta)\\
  &smooth: |\,\mathbf{X}(i) - \mathbf{X}(j)| < \epsilon, && \text{if } \forall j \neq j\prime \cap j \in \partial\mathcal{N}(i,\delta)
  \end{aligned}$$

  Therefore, applying graph wavelet on smooth part can be written as:

  $$\begin{aligned}
  &\Psi_{s,i}^{AGW}\,\mathbf{X}(i) && \text{(smooth part)}\\
  =&\sum_{\delta=0}^{s}\sum_{j\in\partial\mathcal{N}(i,\delta)} \frac{\phi_{s,\delta}}{|\partial\mathcal{N}(i,\delta)|}\mathbf{X}(j) && \text{(traverse scales \& neighbors)}\\
  \approx&\,\mathbf{X}(j)\sum_{\delta=0}^{s}\sum_{j\in\partial\mathcal{N}(i,\delta)} \frac{\phi_{s,\delta}}{|\partial\mathcal{N}(i,\delta)|} && (\mathbf{X}(j) \approx \mathbf{X}_i)\\
  =&\,\mathbf{X}(j)\sum_{\delta=0}^{s} \frac{\phi_{s,\delta}}{|\partial\mathcal{N}(i,\delta)|}|\partial\mathcal{N}(i,\delta)|\\
  =&\,\mathbf{X}(j)\sum_{\delta=0}^{s}\phi_{s,\delta} = \mathbf{X}(j)\cdot 0 = 0 && \text{(Equation 4.5)}.
  \end{aligned}$$

By combining smooth and abrupt parts, the following result is obtained:

$$\begin{aligned}
&|\Psi_{s,i}^{AGW}\,\mathbf{X}(i)|\\
=&\sum_{\delta=0}^{s}\sum_{j}\left|\overbrace{\frac{\phi_{s,\delta}}{|\partial\mathcal{N}(i,\delta)|}\mathbf{X}(j)}^{\text{smooth part}} + \overbrace{\sum_{j\prime}\frac{\phi_{s,\delta}}{|\partial\mathcal{N}(i,\delta)|}\mathbf{X}(j\prime)}^{\text{abrupt part}}\right|\\
\approx&\sum_{\delta=0}^{s}\left|0 + \sum_{j\prime}|\,\mathbf{X}(i) - \mathbf{X}(j\prime)|\right| \gg \epsilon > 0,
\end{aligned}$$

This result shows that the abrupt and smooth neighborhoods can be distinguished by checking their absolute values, i.e., close to zero or significantly larger than zero.

Reason 2: Guarantee for Lossless Recovering. In the classical continuous wavelet (CWT) transform, the admissibility condition [16] is a key that determines whether the CWT can be inverted, and a wavelet ψ satisfies the admissibility condition [16] if Therefore, mean value of ψ must be 0 in discrete version:

$$C_{\psi} = \int_{0}^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} dw < \infty,$$

which means that as for the continuously differentiable ψ, it follows $\hat{\psi}(0) = \int \psi(x)dx = 0$ [16]. Therefore, mean value of ψ must be 0 in discrete version:

$$\hat{\psi}(0) = \sum_{x} \psi(x) = 0.$$

Therefore, only zero sum function satisfies the admissibility condition and ensures information lossless during transformation. This could be proved: when $\omega \to 0$, then $\hat{\psi}(\omega)$ also needs to $\to 0$ otherwise $\frac{|\hat{\psi}(\omega)|^2}{\omega}$ would be ∞ and violates the admissibility condition.

The proposed algorithm makes use of a Mexican hat wavelet, which is a continuous and zero-integral function that fulfills the admissibility condition. Theoretically, the admissibility condition is required because it satisfies two fundamental properties: **(1)** it is localized and has zero integral [32], which can help AGW be more effective at identifying the abrupt graph signal. **(2)** Theoretical guarantee of precise recovery implies that AGW is information-loss-free. By contrast, previous research such as Graph Wavelet Neural Network (GWNN) [14] and GraphWave [15] employs an exponential kernel with integrals greater than 0, which degenerates into a low-pass filter. As a result, GWNN violate the admissibility condition and are incapable of adequately characterizing abrupt graph signals.

4.3 Implementation. Based on the analysis above, AGW kernel, $\Psi^{AGW} \in \mathrm{R}^{N \times N}$, is applied with neural network, and the model is formulated as below:

$$\mathbf{Z} = \Psi^{AGW} \otimes \mathbf{X}^{(l)} \Theta^{(l)}, \tag{4.6}$$

$$\mathbf{X}^{(l+1)} = \sigma(\mathbf{Z}), \tag{4.7}$$

where l is the layer number, $\mathbf{X} \in \mathrm{R}^{N \times F}$ indicates the graph signal, $\Theta \in \mathrm{R}^{F \times C}$ is the learnable parameter of network, and σ is sigmoid activation function. $\otimes$ means each wavelet in Ψ^{AGW} times $\mathbf{X}^{(l)}$ and then apply concatenation on the outcome. Specifically, Ψ^{AGW} consists of multi-scale wavelet by concatenation:

$$\Psi^{AGW} = \Psi^{AGW}_{s=1} \oplus \Psi^{AGW}_{s=2} \oplus \ldots \oplus \Psi^{AGW}_{s=\tau}.$$

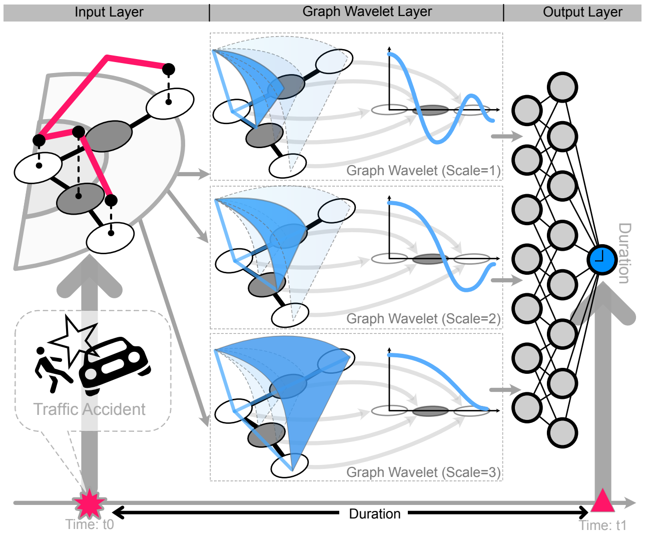


Figure 2: Illustration of AGWN's Architecture

where $\oplus$ denotes concatenation. Therefore Equation 4.6 can be rewritten as concatenation:

$$\Psi^{AGW} \otimes \mathbf{X}^{(l)} = \Psi^{AGW}_{s=1}\, \mathbf{X}^{(l)} \oplus \Psi^{AGW}_{s=2}\, \mathbf{X}^{(l)} \oplus \ldots \oplus \Psi^{AGW}_{s=\tau}\, \mathbf{X}^{(l)}.$$

Equation 4.6 has a time complexity of $\mathcal{O}(N^2)$, while the graph Fourier operation, such as in GCN [3], also has a time complexity of $\mathcal{O}(N^2)$ (After adding fully connected layers, both become $\mathcal{O}(N^2FC)$, where F and C represent the dimensions in hidden and output layers, respectively). When adjacency is sparse, GCN deviates from $\mathcal{O}(N^2)$ to $\mathcal{O}(|E|)$. Similarly, the wavelet transform can be considered sparse because all entries beyond τ are zeros, and τ is often set to a small value. As a result, the sparse representation for GW can be reduced to $\mathcal{O}(N^2)$ rather than $\mathcal{O}(|E|)$. The multi-scale architecture of AGWN is illustrated in Figure 2: the input layer takes a single shot with graph, whereas the graph wavelet layer initializes multi-scale wavelets and performs concatenated wavelet transforms. The output layer is then used to forecast the duration time. In multi-scale AGW design, the outcomes of different graph wavelet transforms are concatenated. Note that there is a trade-off between scale and efficiency since more kernels incur a higher computational cost. Additionally, using several sizes of scales enables easy adoption with paralleled or distributed computation, which may save significant computational resources.

5 Experiment

This section presents synthetic data experiment for AGW, and evaluation on real-world data for AGWN [1]. All experiments are conducted with GPU 1080Ti.

[1] Code is available at https://github.com/gm3g11/AGWN

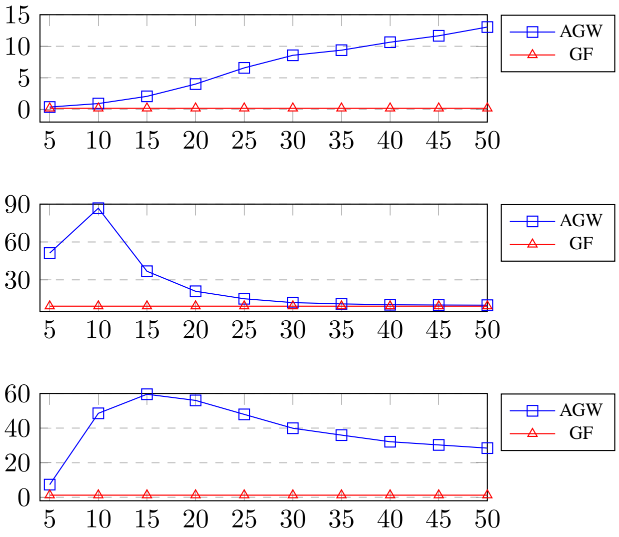


Figure 3: Fisher scores after different transforms. The X-axis denotes scale (τ), while Y-axis indicates Fisher score. **Top**: abrupt point signal; **Middle**: abrupt line signal; **Bottom**: abrupt area signal. Note that GF does not have a scale parameter, so it is constant w.r.t. scale.

5.1 Evaluation on Synthetic Data. This subsection is for the purpose of verifying the theoretical study on graph wavelets: (1) On abrupt signals, does graph wavelet yield more separable representations than graph Fourier? (2) Is the admissibility condition a requirement for the ability of a graph wavelet to deal with abrupt signals? To account for all possible scenarios, the Minnesota road network is subjected to three types of simulated traffic congestion: point, line, and area congestion (see visualization in Figure 8 in supplementary materials). All the nodes are assigned with value 0 (normal traffic), whereas a few nodes which are blocked are set to value 1 (congestion):

- The point congestion is generated by selecting 100 nodes randomly and setting them to value 1.
- The line congestion is generated by calculating the shortest path from the 1st node which is selected as the beginning of the line to the 2400th node which is selected as the end of the line, and setting the nodes in the path to value 1.
- The area congestion is generated by calculating the shortest distance between the 1900th node which is selected as the center of the area and the other nodes and setting these nodes whose distance is the shortest 10% of all to the value 1.

Different scales are sensitive to different ranges (Figure 9 illustrate scales in the supplementary materials). Notably, a larger size also entails a greater computing cost.

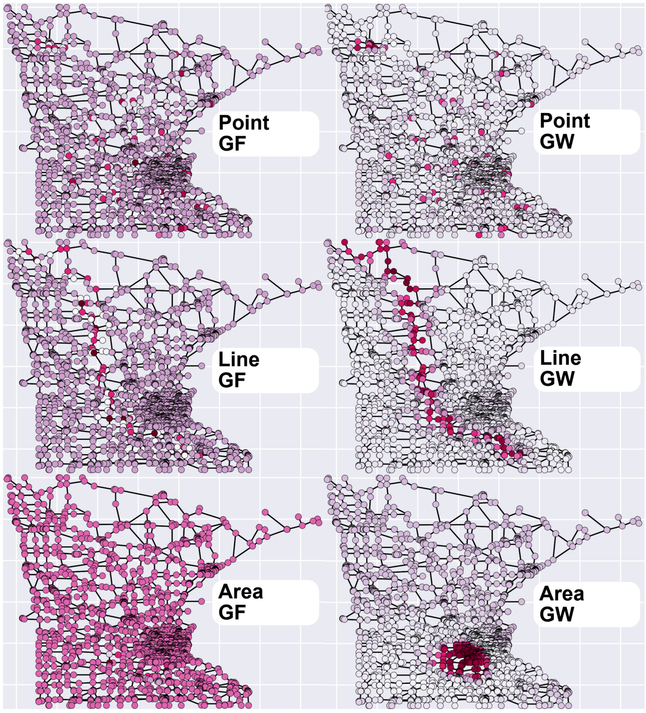


Figure 4: Examples of graph Laplacian and graph wavelet for different signals. Deeper red is close to value 1, and more white is close to value 0. **Top** row: abrupt point signal; **Middle** row: abrupt line signal; **Bottom** row: abrupt area signal; **Left** column: graph Laplacian; **Right** column: graph wavelet.

5.1.1 Graph wavelet and graph Fourier transform. Using the Fisher score, this experiment compares the linear separability of AGW and graph Fourier transform. The greater the Fisher score, the more separable the linear data is. As illustrated in Figure 3 abrupt point line area (corresponding number is listed in Table 3 in supplementary materials), Fisher scores for AGW are typically much higher than those for GF. The scales with the highest Fisher scores vary: 50 is the best scale for an abrupt point signal, 10 is the best scale for an abrupt line signal, and 15 is the best scale for an abrupt area signal. Figure 4 depicts the distinction between graph Laplacian and graph wavelet, demonstrating that GF smooths the signal while GW accentuates it. In area congestion signal (the third row in Figure 4), there is little visual difference between congestion and the others.

5.1.2 Ablation test for admissibility condition. The condition of admissibility is empirically investigated using the

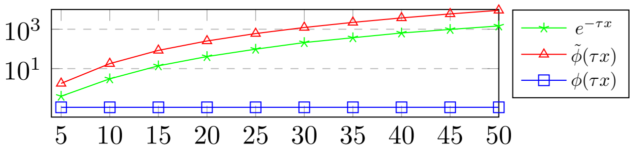


Figure 5: Ablation test of admissibility condition. X- and Y-axis denotes scale (τ) and MSE.

mean square error of recovery (transform and then inverse transform). Accurate recovery entails lossless transformation. Specifically, we compared three kernels: one is Mexican hat wavelet ($\phi(\tau x)$, where τ is the scale) [2] whose integral is 0, satisfying the admissibility condition. The second is $\tilde{\phi}(\tau x) = \phi(\tau x) + 1$, which violates the admissibility condition. The third one is heat kernel ($e^{-\tau x}$) that is often used in existing graph wavelet neural networks [14, 15]. As shown in Figure 5 (see Table 4 for numbers in supplementary materials), $\phi(\tau x)$ has the smallest error, while the error significantly increased by a upshift (i.e., $\tilde{\phi}$) or by heat kernel. Note that the error level of Mexico hat is almost constant.

5.2 Evaluation on Real-world Data. Our experiments are conducted on real-world data compiled with three sources:

- **[Feature (X)] Traffic data**: PeMS-8 by Caltrans Performance Measurement System (PeMS) [33] is used as traffic data, which contains the occupancy rate and average vehicle travel speed of the San Bernardino/Riverside and Stockton areas in California.
- **[Graph (G)] Road network data**: Corresponding road networks are extracted from Topologically Integrated Geographic Encoding and Referencing (TIGER) shapefiles by U.S. Census Bureau's Master Address File [3], which represents the topological connections among the roads and corridors.
- **[Label (Y)] Accident record**: 8574 traffic accident records are collected from the Regional Integrated Transportation Information System (RITIS)[4]. Accident duration is used as the ground truth.

This dataset aggregates traffic records over a 5-minute period, and each sensor node has two features: the occupancy rate is between 0 and 100%, and the traffic speed is between 0 and 80 (miles/hr). To eliminate the magnitude's influence on the features, normalization using the Z-Score approach is used. Each sensor is considered as a node in a graph when constructing the road network graph. Each pair of nodes has a distance between them defined as the Euclidean distance between their real latitude and longitude coordinates. Then, for each pair of nodes, the pairwise distances are calculated, and the reciprocal of the distance is utilized as the edge weight. This Euclidean-based weight is frequently utilized in similar work [20, 6] because to (1) the lack of reliable data for road connectivity in conjunction with sensors, and (2) sufficient precision for the majority of nodes due to the physical connection of most neighbors. Additionally, we allocated 70% of the data for training, 10% for validation, and 20% for testing. Each accident has 60 shots with a 5-minute interval. The 40th shot is used as the input, as it is the one taken immediately following each accident. Our objective is fundamentally distinct from many other related research in traffic speed prediction, which need the input of a time series just before the target.

Additionally, we experiment on two subgraphs of the original one. Specifically, we split the PeMS08 traffic sensor nodes by the county code information from the CalTrans. The county code is the unique number that identifies the county that contains this census station within PeMS. Our experiment identifies two counties (county number 65 and 71) in the PeMS08 area. Therefore, we split the original PeMS08 dataset into two sub-dataset representing county numbers 65 and 71, which have 695 and 495 nodes. In the next step, it is necessary to separate the traffic accident into two parts correspondingly. This segmentation method depends on the location of the accidents and attributes the data to the nearest sensor node. Through this method, we split the total 8574 traffic accident records into 4634 and 3940. In order to ensure the accuracy of this division, some accident data near the boundary of two counties would drop off.

Following the preparation of the dataset, this section will discuss the models. To begin, we propose the AGWN model, which has a single critical hyperparameter: scale. To determine the scale range, we use the Dijkstra shortest path algorithm, which is included in the NetworkX package [5]. Then, based on these scales, we calculate different abrupt graph wavelets and integrate them into an AGWN model in which the weight is determined by the self-attention mechanism [34]. By contrast, we also implement some classical graph models that are suitable for our task. All the baselines are implemented with PyTorch Geometric [6], except that GWNN is implemented by its independent official code [7]. Baselines include popular graph neural networks such as GCN [3], ChebNet [35], GAT [4], GraphSAGE [5], Graph Isomorphism Network (GIN) [36], Simplifying GCN (SGC) [37]. Graph Wavelet Neural Network (GWNN) [14] is also included for showing the advantage of AGW over non-AGW. In this experiment, the event forecasting task is to predict accident duration. Hence, we choose the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), and the Mean Absolute Percentage Error (MAPE) as the evaluation criterion. Besides, we choose the Adam as the optimizer, and the learning rate is set as 0.001 at the beginning and would decay after 200 epochs. All models ran for 1000 epochs, and they all converged. The following two Ta-

[2] Implementated by Ricker with SciPy [link]

[3] https://catalog.data.gov/dataset/tiger-line-shapefile-2015-state-california-primary-and-secondary-roads-state-based-shapefile

[4] https://ritis.org/access

[5] Implementated by single-source Dijkstra path length with NetworkX

[6] https://github.com/rusty1s/pytorch_geometric

[7] https://github.com/benedekrozemberczki/GraphWaveletNeuralNetwork

bles 1 and 2 record the experiment results by implementing different graph models.

Table 1: Experiment results on real world traffic accident datasets: MAE, RMSE, MAPE in the first subgraph of the county

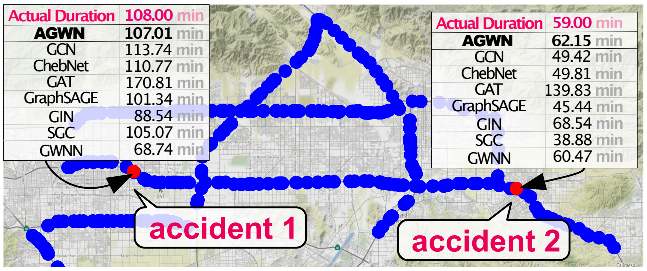
Method	MAE	RMSE	MAPE
GAT	10.60 ± 3.38	8.37 ± 1.91	24.02% ± 12.79%
SGC	7.61 ± 2.87	8.05 ± 1.97	21.77% ± 17.79%
GIN	11.85 ± 2.61	9.9 ± 2.01	18.02% ± 14.26%
GWNN	11.79 ± 3.14	9.25 ± 2.45	19.34% ± 7.30%
GCN	7.54 ± 3.27	7.98 ± 1.94	20.65% ± 16.67%
GraphSAGE	10.6 ± 3.38	8.37 ± 1.91	24.02% ± 12.79%
ChebNet	7.51 ± 3.3	6.94 ± 1.97	19.36% ± 18.09%
AGWN	**6.37 ± 2.41**	**5.70 ± 1.11**	**14.64%± 5.41%**

Table 2: Experiment results on real world traffic accident datasets: MAE, RMSE, MAPE in the second subgraph of the county

Method	MAE	RMSE	MAPE
GAT	11.26 ± 3.42	7.45 ± 2.5	25.77% ± 13.54%
SGC	7.04 ± 3.6	7.79 ± 2.59	11.52% ± 14.15%
GIN	15.10 ± 3.84	10.76 ± 2.81	21.22% ± 10.56%
GWNN	13.81 ± 4.05	10.38 ± 2.85	22.00% ± 6.70%
GCN	7.54 ± 3.56	7.93 ± 2.43	16.99% ± 13.11%
GraphSAGE	7.19 ± 3.58	5.66 ± 2.46	13.37% ± 13.79%
ChebNet	7.37 ± 3.61	6.94 ± 2.55	15.93% ± 13.60%
AGWN	**6.97 ± 2.98**	**4.64 ± 2.12**	**6.64%± 2.78%**

Experimental Results Analysis. Table 1 and the table 2 show 5-fold experiment results. AGWN significantly outperforms the other baselines in predicting accident duration. In the first subgraph, AGWN improves the best baseline (i.e., ChebNet) by 17.89% of MAE, 21.75% of RMSE, and 23.1% of MAPE. Besides, we could find the criterion on the AGWN model does not vary much on both subgraphs which could prove the robustness of the AGWN . Furthermore, the AGWN model has the smallest variance comparing to other models, and this could illustrate the stability of itself. GWNN has average performance as the other baselines, implying that non-AGW does not have any advantage beyond GF-based methods in abrupt signal modeling. Each training epoch of AGWN only took 0.09 min on average, dramatically faster than all the baselines. *Efficiency:* For each epoch, models' average runtime (in minute) are: 1.77 for GAT , 2.56 for SGC, 0.29 for GIN, 1.88 for GWNN, 1.02 for GCN, 0.4 for GraphSAGE, 1.43 for ChebNet. Our AGWN is 0.09, indicating a high efficiency.

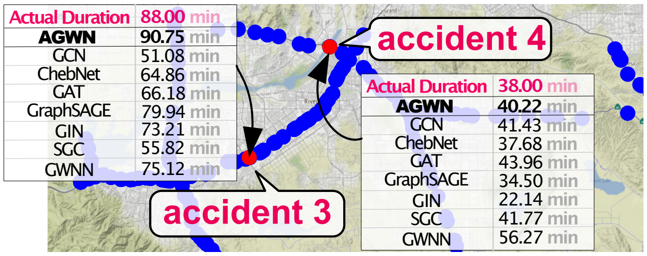
Case Study. Figure 6 and Figure 7 depict four accident cases from the dataset: accident 1 (latitude/longitude is 34.078035/-117.62377) and accident 2 (34.061758/-117.179004) happened in the first subgraph, whereas accident 3 (33.905268/-117.458919) and accident 4 (33.878030/-117.658212) happened in the second subgraph. Comparing the actual accident durations to the predicted results by all of the models, we could summarize the following items: firstly, although occasionally other models could give out more accurate results. Specifically, AGWN could predict more accurate results in all the cases and outperforms the other models overall except in the accident 2 (still is the second best model); secondly, the AGWN is more robust as the prediction results are closed to the ground truth and the error value would not exceed 4 minutes, whereas other models have violent fluctuations or have unstable performance.



Actual Duration	108.00 min
AGWN	**107.01** min
GCN	113.74 min
ChebNet	110.77 min
GAT	170.81 min
GraphSAGE	101.34 min
GIN	88.54 min
SGC	105.07 min
GWNN	68.74 min

Actual Duration	59.00 min
AGWN	**62.15** min
GCN	49.42 min
ChebNet	49.81 min
GAT	139.83 min
GraphSAGE	45.44 min
GIN	68.54 min
SGC	38.88 min
GWNN	60.47 min

Figure 6: Case study: **(Accident 1)** and **(Accident 2)**



Actual Duration	88.00 min
AGWN	**90.75** min
GCN	51.08 min
ChebNet	64.86 min
GAT	66.18 min
GraphSAGE	79.94 min
GIN	73.21 min
SGC	55.82 min
GWNN	75.12 min

Actual Duration	38.00 min
AGWN	**40.22** min
GCN	41.43 min
ChebNet	37.68 min
GAT	43.96 min
GraphSAGE	34.50 min
GIN	22.14 min
SGC	41.77 min
GWNN	56.27 min

Figure 7: Case study: **(Accident 3)** and **(Accident 4)**

6 Conclusions

This paper studied the early forecast of traffic accident impact with graph learning. After formulating traffic accidents as abrupt graph signals, we analyzed major graph learning techniques such as graph Fourier and graph wavelet by quantifying their linear separability, showing an apparent difference between them. Then sensitivity analysis and the admissibility condition are used to distinguish abrupt and non-abrupt graph wavelets, highlighting the prominent advantage of abrupt graph wavelets. Accordingly, we designed an end-to-end graph neural network, AGWN, to characterize traffic accidents. Finally, the synthetic data experiment verified our understanding between graph Fourier and graph wavelet, justifying our kernel design. Promising results and cast study on real-world data demonstrated AGWN 's effectiveness and efficiency in the early forecast of traffic accident impact.

Acknowledge

A student abstract with the same title has been published in AAAI 2021 Student Abstract and Poster Program.

References

[1] M. W. Adler, J. van Ommeren, and P. Rietveld, "Road congestion and incident duration," *Economics of transportation*, vol. 2, no. 4, pp. 109–118, 2013.

[2] R. Li, F. C. Pereira, and M. E. Ben-Akiva, "Overview of traffic incident duration analysis and prediction," *European Transport Research Review*, vol. 10, no. 2, p. 22, 2018.

[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[5] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurlPS*, 2017.

[6] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv:1709.04875*, 2017.

[7] X. Geng *et al.*, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *AAAI*, 2019.

[8] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *TITS*, 2019.

[9] H. NT and T. Maehara, "Revisiting graph neural networks: All we have is low-pass filters," *arXiv:1905.09550*, 2019.

[10] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *AAAI*, 2018.

[11] Q. Li, X.-M. Wu, H. Liu, X. Zhang, and Z. Guan, "Label efficient semi-supervised learning via graph filtering," in *CVPR*, 2019.

[12] Z. Chen, F. Chen, L. Zhang, T. Ji, K. Fu, L. Zhao, F. Chen, and C.-T. Lu, "Bridging the gap between spatial and spectral domains: A survey on graph neural networks," *arXiv:2002.11867*, 2020.

[13] D. M. Mohan, M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Wavelets on graphs with application to transportation networks," in *ITSC*. IEEE, 2014.

[14] B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng, "Graph wavelet neural network," *arXiv:1904.07785*, 2019.

[15] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *KDD*, 2018.

[16] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, 2011.

[17] N. Owens, A. Armstrong, P. Sullivan, C. Mitchell, D. Newton, R. Brewster, and T. Trego, "Traffic incident management handbook," Tech. Rep., 2010.

[18] L. Bai, L. Yao, S. Kanhere, X. Wang, Q. Sheng *et al.*, "Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," *arXiv preprint arXiv:1905.10069*, 2019.

[19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv:1901.00596*, 2019.

[20] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE SPM*, 2013.

[21] C. Yang, R. Wang, S. Yao, S. Liu, and T. Abdelzaher, "Revisiting" over-smoothing" in deep gcns," *arXiv preprint arXiv:2003.13663*, 2020.

[22] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *International Conference on Learning Representations*, 2019.

[23] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *International Conference on Learning Representations*, 2019.

[24] W. Huang, Y. Rong, T. Xu, F. Sun, and J. Huang, "Tackling over-smoothing for general graph convolutional networks," 2020.

[25] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1725–1735.

[26] M. Liu, H. Gao, and S. Ji, "Towards deeper graph neural networks," in *KDD*, 2020, pp. 338–348.

[27] K. Zhou, X. Huang, Y. Li, D. Zha, R. Chen, and X. Hu, "Towards deeper graph neural networks with differentiable group normalization," 2020.

[28] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *AAAI*, vol. 34, no. 04, 2020, pp. 3438–3445.

[29] Y. Min, F. Wenkel, and G. Wolf, "Scattering gcn: Overcoming oversmoothness in graph convolutional networks," *arXiv preprint arXiv:2003.08414*, 2020.

[30] A. J. Izenman, "Linear discriminant analysis," in *Modern multivariate statistical techniques*. Springer, 2013.

[31] J. Chapa and R. Rao, "Algorithms for designing wavelets to match a specified signal," *IEEE Trans. Signal Process.*, 2000.

[32] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," in *IEEE INFOCOM*, 2003.

[33] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[35] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *NeurlPS*, 2016.

[36] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," *arXiv:1905.12265*, 2019.

[37] F. Wu, A. H. Souza Jr, T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *ICML*, 2019.