Enhancing Urban Mobility with Graph Traffic Attenuation (GTA)

Syed Ali
syedali@my.unt.edu

Mica Haney
micahaney@my.unt.edu

Twumasi Mensah-Boateng
twumasimensah-boateng@my.unt.edu

***Abstract*—In this study, we explore the enhancement of traffic forecasting models by applying advanced graph neural network (GNN) architectures to a specific segment of Seattle's road network. By incorporating a comprehensive dataset that details vehicular velocities through inductive loop detectors across the city's main arteries, this work aims to innovate upon existing GNN frameworks to address the pressing needs of urban mobility. Our approach utilizes the intrinsic graph structure of road networks to more accurately predict and manage traffic flows, thereby contributing to smarter, more sustainable urban transport systems. The proposed model, underpinned by the latest GNN research, employs advanced spectral graph convolution techniques, leveraging both spatial and temporal traffic data to refine prediction accuracy and generalizability, and is tested against traditional methods to highlight its performance gains.**

Capitalizing on the spectral properties of the graph Laplacian, the model introduced in this paper, GTA Model, enhances traffic forecasting through improved graph convolution operations and LSTM integration. By integrating dropout and batch normalization into spectral graph convolutions, and incorporating residual connections, the model aims to address overfitting and improve training stability—key factors for deep learning in complex network scenarios. The advancements in spectral filtering and the strategic use of regularization techniques point to this model's ability to learn more effectively from the intricate patterns in traffic data. This research underscores the transformative potential of GNNs in urban traffic management, presenting a forward-looking solution that can adapt to the dynamic and interconnected nature of city road networks.

I. Introduction

We will apply Graph Neural Networks to a transportation network for this project. Specifically, we will be performing the task of vertex classification using the Seattle Loop dataset. We will be looking at vertices on the dataset and creating a model that can look at vehicle speeds and classify if traffic at that vertex location is flowing correctly or if there is a traffic slowdown.

Transportation has been a cornerstone of human civilization throughout history, evolving from simple footpaths to complex networks of roads, railways, and air routes. This evolution mirrors the progression of societies, facilitating migration, commerce, and the interconnection of distant regions. Yet, as urban areas burgeon with dense populations, the once sufficient infrastructure and road networks now falter under the strain of serving an ever-growing populace. In contemporary urban environments, efficient transportation systems are not just conveniences but necessities, underpinning economies, enabling access to employment, and ensuring the delivery of goods and services. Amidst this backdrop, our project delves into the challenge of forecasting traffic—a task of increasing complexity and critical importance as cities expand and evolve.

Traditional traffic management systems, reliant on historical data and static models, are proving inadequate for the dynamic and interconnected nature of modern urban traffic flows. Herein lies the potential of neural networks, which have demonstrated remarkable efficiency in handling various data types, including images, text, and sound. However, the unique spatiotemporal characteristics of traffic data—with its patterns influenced by myriad factors such as time of day, weather, and unforeseen events—demand a more sophisticated approach than conventional neural network models can offer.

Enter graph neural networks (GNNs), a cutting-edge adaptation of neural network technology, designed to excel with data that naturally forms a graph structure—such as road networks. Unlike their predecessors, GNNs are adept at managing the complexity of graph data, enabling the extraction of rich insights from the interconnected elements of traffic systems. By leveraging GNNs, we can surmount the limitations of traditional models, harnessing the voluminous data generated by the Internet of Things (IoT) and other sources to forecast traffic with unprecedented accuracy and efficiency.

In this project, we propose the use of graph neural networks to revolutionise traffic control and management. Our objective is to predict traffic flow across road networks with a precision that allows for proactive traffic management, potentially mitigating congestion, reducing emission levels, and enhancing urban mobility. By innovatively applying GNNs to this realm, we not only address a pressing urban challenge but also contribute to the broader field of transportation science.

To ground our work in the latest research, we draw upon seminal papers such as "A Comprehensive Survey on Graph Neural Networks" by Zonghan Wu et al. [1], which elucidates the foundations and applications of GNN technology. Our project is poised at the intersection of transportation needs and technological innovation, offering a forward-looking solution to a perennial challenge. By integrating graph neural networks into traffic forecasting, we not only aim to enhance urban transportation systems but also contribute to the sustainable development of cities, ensuring they remain vibrant and livable for future generations.

II. Background

In the domain of computational modeling, traditional neural networks have made significant strides in fields such as image recognition, natural language processing, and audio analysis.

Despite these achievements, such architectures reveal inherent limitations when applied to the modeling of transportation networks. These networks are fundamentally graph-structured, comprising an intricate web of nodes (like intersections and waypoints) and edges (such as roads and pathways) that illustrate the spatial and functional relationships within the system. The standard neural network's matrix-based approach does not naturally extend to the complex topologies of transportation data.

Graph Neural Networks (GNNs) emerge as a formidable solution, crafted specifically to navigate the complexities of graph-structured data. Unlike traditional neural networks, GNNs are intrinsically designed to grasp the rich relational context of graphs, enabling them to adeptly perform tasks ranging from node classification to complete graph prediction. By assimilating the transportation network's structure into their models, GNNs are uniquely equipped to discern the intricate dependencies and emergent patterns within vast traffic datasets.

The initiative detailed in this paper employs GNNs to tackle the formidable challenge of traffic forecasting within urban transportation networks. Utilizing a dataset cited in [2] and [3], the project aspires to devise a predictive model capable of recognizing and anticipating traffic conditions with a high degree of accuracy. The successful implementation of this model promises to transform transportation management, providing a foundation for smarter, more responsive urban traffic systems.

This research is part of a larger quest to apply advanced artificial intelligence methodologies to improve the robustness and adaptability of urban infrastructure. By introducing GNNs into the analysis of transportation networks, the study positions itself at the forefront of a technological revolution in traffic management. It aims to contribute to a future where traffic flow is not only monitored but also prognosticated with precision, leading to the alleviation of congestion, the reduction of emissions, and the enhancement of overall urban mobility.

In doing so, the project contributes to the broader domain of transportation science, standing at the intersection of technological innovation and practical application. The potential benefits extend beyond smoother traffic flow; they encompass the sustainable development of urban environments, ensuring cities remain habitable and efficient in the face of escalating population density and mobility demands. Through the application of GNNs to the Seattle Loop dataset, the project not only addresses a critical urban challenge but also paves the way for future advancements in the integration of AI in city planning and management.

III. Related Works

A. TGC-LSTM

The paper introduces a novel deep learning framework known as the Traffic Graph Convolutional Long Short-Term Memory Neural Network (TGC-LSTM). This model is designed to address the challenges of traffic forecasting by leveraging the structural data of traffic networks modeled as graphs. The TGC-LSTM framework innovatively combines graph convolution with LSTM units to capture both the spatial and temporal dependencies in traffic data, thus enhancing the accuracy of traffic state forecasts across network-wide locations.

A key innovation of the TGC-LSTM model is the introduction of a traffic graph convolution operator, tailored to respect the physical layout of traffic networks. This operator utilizes the physical network topology to define graph convolution in a way that aligns with actual roadway interconnections. By incorporating regularization techniques, such as L1-norm on graph convolution weights and L2-norm on graph convolution features, the model achieves improved interpretability and robustness. These regularizations help maintain the sparsity of the model and stabilize the learned features, making them more meaningful and easier to understand in real-world contexts.

Experimental results demonstrate that the TGC-LSTM model surpasses various baseline methods on real-world datasets, proving its effectiveness in traffic forecasting. The model not only shows superior performance in predicting traffic states but also provides insights into the most influential road segments within the traffic network. This capability of identifying critical areas can be instrumental for urban planning and traffic management, underscoring the practical implications of integrating deep learning with graph-based modeling in traffic systems.

IV. Methods

A. Data

The foundational dataset for our project is the Seattle road network data, sourced from references [2] and [3]. This dataset was compiled using inductive loop detectors—a type of sensor that records the velocity and classifies vehicles as they pass over. With a network of 323 such sensors embedded across critical junctures, the dataset forms a comprehensive graph where each sensor is conceptualized as a graph vertex, and each road segment connecting these sensors is represented as an edge in an adjacency matrix.

It aggregates time-series velocity data captured by these loop detectors distributed strategically along four major freeways in Seattle—namely I-5, I-405, I-90, and SR-520. The granularity of the data is such that it provides average vehicular speeds over 5-minute intervals, encapsulating an entire year's traffic dynamics across 323 sensor stations.

An inherent strength of the LOOP dataset lies in the uniformity of sensor technology used, which translates to homogenous data quality with closely matched noise profiles. This uniformity is instrumental in achieving a consistent analytical baseline. Moreover, the dataset's coverage of a limited geographic region ensures that extrinsic factors like weather impact the sensor readings uniformly, which simplifies the process of accounting for such variables in traffic analysis.

However, the LOOP dataset is not without its constraints. Notably, it lacks diversity in features—it predominantly tracks

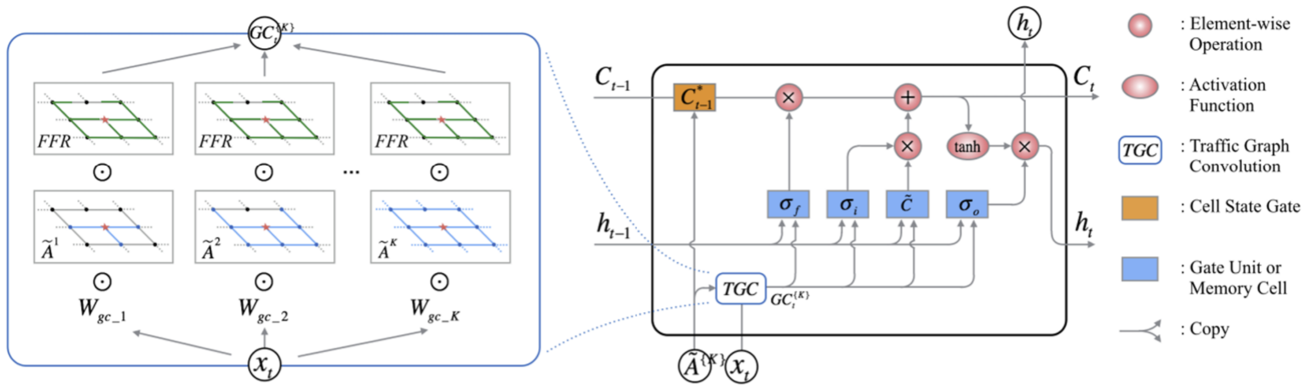

Fig. 1.

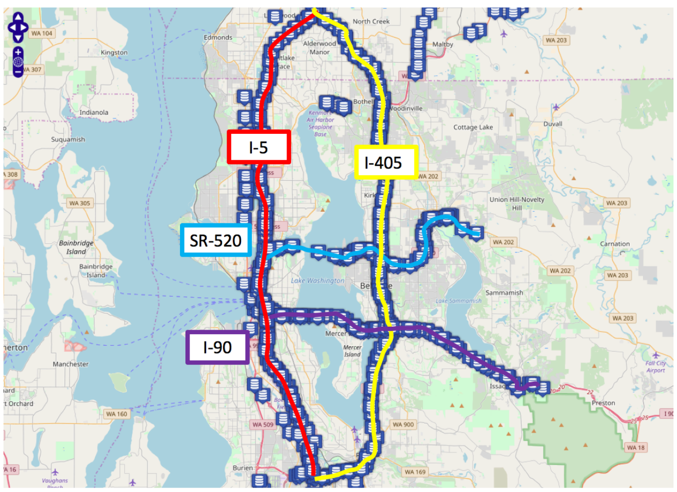

Fig. 2. Residual Plot

average speeds and does not include other potentially influential variables. Additionally, the dataset does not provide precise geolocation data for each sensor, which can be critical for detailed spatial analysis. While the dataset offers free-flow matrices that serve as proxies for node reachability within certain time frames, the absence of accurate distance measurements between sensors presents a challenge.

To mitigate these limitations, we have reconstructed several free-flow rate (FFR) matrices, incorporating estimated distances at road intersections. This manual intervention was necessitated when a close examination of the dataset revealed discrepancies in the provided FFR matrices. Specifically, sensors positioned at proximate mile markers on the same highway and direction often appeared unconnected, a gap that, when bridged, significantly impacted the network's topology.

Through the iterative process of refining these FFR matrices, which we treat as a hyperparameter, we have developed an enhanced methodological framework for our project. These matrices have been calibrated to align more closely with the physical road network, allowing for a more accurate representation of traffic flow and connectivity within the analyzed network segment. By addressing the dataset's shortcomings, we aim to bolster the reliability of our subsequent traffic forecasting models, thereby contributing to the advancement of intelligent transportation systems.

B. Model

The model will be a GNN built with PyTorch. We will use both PyTorch tutorials linked in [5] and code examples linked from and found in the GitHub repository [6] for the survey paper [7]. The model will have at least two hidden layers, and experiments will be run to determine the best model architecture.

1) Original Model: The GraphConvolutionalLSTM model represents an advanced integration of spectral graph convolutions with the recurrent neural network architecture of LSTMs, tailored for dynamic graph-structured data analysis. At its core, the model operates by harnessing the adjacency matrix of the graph, raising it to successive powers to encapsulate varying hop distances. These calculated matrices are then combined with the free-flow reachability matrix (FFR), effectively weighting the graph connections based on reachable nodes within specific time frames. The graph convolution aspect of the model employs a method called FilterLinear, a specialized linear transformation that is adapted according to the adjacency matrix to maintain the integrity of the graph's structural information. Integrating LSTM units into this setup allows the model to process temporal sequences effectively, with gate mechanisms being informed by a fusion of the outputs from graph convolutions and the recurrent network's hidden states. This sophisticated combination is governed by sigmoid and tanh activation functions, which regulate the gates and state updates essential for the LSTM's temporal learning capability.

The SpectralGraphConvolution layer underpins the GraphConvolutionalLSTM, providing a means to implement spectral analysis on the graph structure. It calculates the eigenvalues and eigenvectors of the graph Laplacian matrix, a representation that reflects the essential features of the graph's topology in the spectral domain. These spectral components are then selectively scaled using a parameterized approach, allowing for refined control over the filtering process. The model leverages these parameters to accentuate or diminish specific spectral features before the node features are reconstructed, thereby implementing a form of feature engineering that is both nuanced and sensitive to the underlying graph structure. The use of SpectralGraphConvolution enables the model to exploit the graph's inherent symmetries and clusters, providing a powerful mechanism to enhance feature representation and facilitate more accurate predictions in traffic forecasting and beyond.

2) Enhanced Model: The GTA Model model introduces significant enhancements over its predecessor, primarily by incorporating the SpectralGraphConvolution_2 in place of the simpler FilterLinear graph convolutions. This updated convolutional approach integrates dropout and batch normalization to address issues of overfitting and stabilize the learning process. Residual connections also mark a key advancement, allowing the network to potentially maintain effective gradient flow during training, a crucial factor in enabling deeper network architectures.

SpectralGraphConvolution_2 builds upon the original spectral analysis by adding regularization elements such as dropout to combat overfitting and batch normalization to maintain stability across the graph's nodes. These improvements are designed to bolster the model's ability to generalize by refining its spectral filtering operations. The enhancements in the convolutional layers, combined with robust regularization techniques, are anticipated to result in a model that is both more stable during training and more adept at generalizing from the training data. Residual learning introduced in the GTA Model can facilitate deeper learning by supporting the network in learning identity functions where beneficial and improving the flow of gradients.

V. Experiment

Two models were crafted for this project. The first model is a direct recreation of the model in [8] and was created using code provided by the paper's authors. The second model was created on a base of the original model with some modifications as discussed in IV-B2.

¡MODIFICATIONS¿

70

VI. Results

A. Training

Figure 3 presents a comparison of training and validation loss between the original and the GTA model across epochs. For both models, the training loss shows a declining trend, indicating learning progress. Notably, the GTA model exhibits a consistently lower loss than the original, indicating improved learning efficiency. This is true for both the training and validation phases, hinting at the GTA model's superior ability to generalize. By the 10th epoch, the validation loss of the GTA model converges closely with its training loss, demonstrating stability and indicating that additional training epochs might not yield significant gains. The original model, while also showing a convergence of training and validation loss, does so at higher loss values, indicating that the GTA model will achieve better performance on new data.

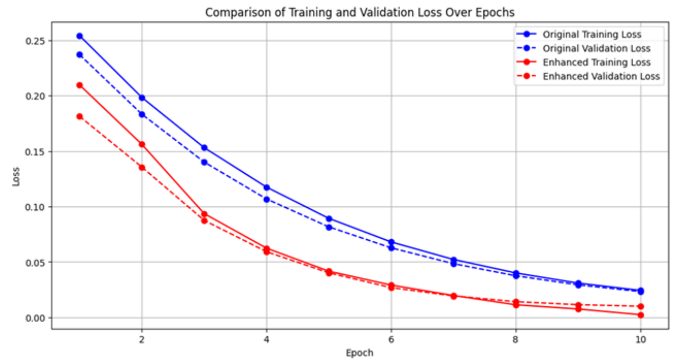

Fig. 3. Training loss vs validation loss comparison for the both models.

TABLE I
EXPERIMENT RESULTS

		Original Model	GTA Model
MAE	Min.	0.0226	0.0334
	Avg.	0.0268	0.0380
	Max	0.0323	0.0437
	Std. Dev.	0.0018	0.0021
MSE	Min.	0.0011	0.0022
	Avg.	0.0017	0.0030
	Max	0.0023	0.0038
	Std. Dev.	0.0002	0.0003
R^2	Min.	0.3478	-0.4390
	Avg.	0.5746	0.1327
	Max	0.6947	0.4013
	Std. Dev.	0.0661	0.1305

B. Metrics

Results are being measured using Mean Squared Error Loss (MSE Loss), Mean Absolute Error Loss (MAE Loss), and R^2 Score. Originally we intended to use accuracy, precision, recall, and F1 scores as the evaluation metrics. However, due to time and data constraints the problem had to be reformatted from a classification prediction to a regression prediction. Since the previously mentioned metrics are measures of classification, we could no longer use them and switched to regressive metric measures.

The reported results scores are being calculated from the held-out test set. Training and validation scores are not reported as they are used for model tuning.

As can be seen in Table 1, the loss scores for the GTA model are higher than the original model's, though not significantly, with both models having fairly good loss scores. The R^2 scores, on the other hand, are significantly worse for the GTA model.

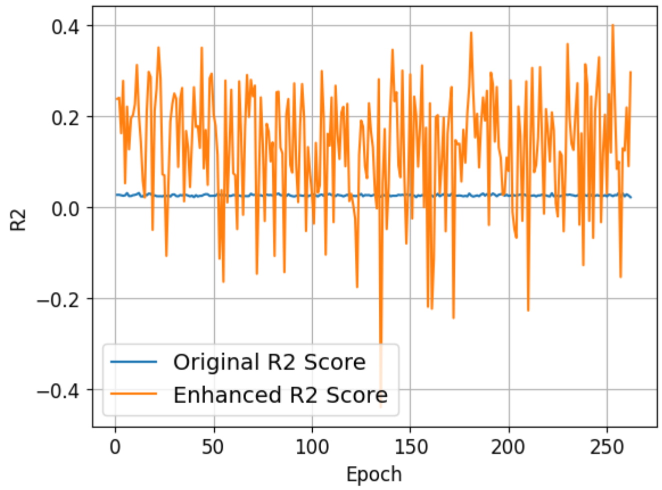

Fig. 4. Residual Plot

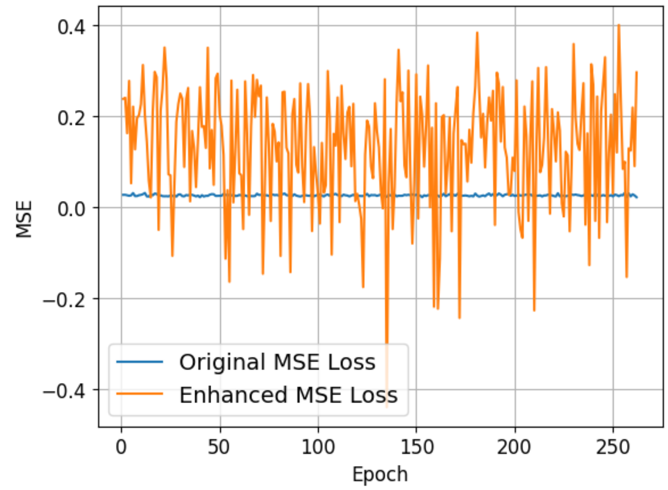

Fig. 6. Residual Plot

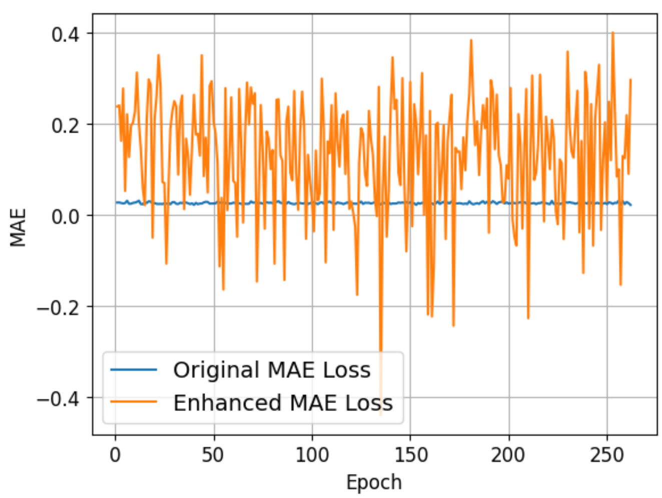

Fig. 5. Residual Plot

VII. Conclusion

The enhanced models introduce significant improvements intended to stabilize training, enhance feature learning through advanced spectral techniques, and potentially improve performance through architectural tweaks like residual connections and regularization. These enhancements are designed to make the models more robust and adaptable to complex graph-structured data, offering better performance especially in scenarios where the original models may struggle with depth and complexity. The worse testing scores of the GTA model indicate that the faster training has likely resulted in overfitting, meaning that future experiments with this architecture will need to either end training sooner or slow the learning rate. Overall the GTA model offers performance enhancements over the original model without unduly sacrificing model performance.

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: 10.1109/tnnls.2020.2978386.

[2] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic Graph Convolutional Recurrent Neural Network: A deep learning framework for Network-Scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020, doi: 10.1109/tits.2019.2950416.

[3] Z. Cui, R. Ke, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," *arXiv (Cornell University)*, Jan. 2018, doi: 10.48550/arxiv.1801.02143.

[4] GceLab, "What is Inductive Loop Detector? 8 Important Points." https://www.gcelab.com/blog/what-is-inductive-loop-detector-and-advantages

[5] "PyG Documentation — pytorch_geometric documentation." https://pytorch-geometric.readthedocs.io/en/latest/

[6] Jwwthu, "GitHub - jwwthu/GNN4Traffic: This is the repository for the collection of Graph Neural Network for Traffic Forecasting.," *GitHub*. https://github.com/jwwthu/GNN4Traffic/tree/master

[7] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Systems With Applications*, vol. 207, p. 117921, Nov. 2022, doi: 10.1016/j.eswa.2022.117921.

[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," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2019, doi: https://doi.org/10.1109/tits.2019.2950416.