Graph Neural Networks for Long-Term Traffic Forecasting

Can GNNs effectively handle long-term predictions and how does their accuracy degrade over time?

Vlad Vrânceanu

Supervisor: Elena Congeduti

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Vlad Vrânceanu
Final project course: CSE3000 Research Project
Thesis committee: Elena Congeduti, Lilika Markatou

Abstract

Traffic forecasting is a branch of spatiotemporal forecasting that involves predicting future traffic speed or volume based on real-world data. It has a significant impact on urban mobility and quality of life, as it directly contributes to improving traffic management and trip planning. This study evaluates the performance of Graph Neural Networks (GNNs) in handling long-term forecasting, defined as predictions made up to 10 hours ahead. It addresses the evolution of performance and factors that may impact accuracy, such as fluctuations in traffic speed and road network configurations. The experiments are done using subsets of a benchmark dataset for traffic forecasting and a state-of-the-art GNN model. The findings showcase a logarithmic growth in prediction errors and the presence of two types of traffic jams—sudden and regular—along with their impact on prediction accuracy. Furthermore, the results highlight the complexity of quantifying the influence a factor has on forecasting performance, such as road network configuration or missing values.

1 Introduction

Spatiotemporal forecasting is the process of predicting future states or events across spatial and temporal dimensions. There are multiple applications for this, but the paper will focus on traffic forecasting, which predicts traffic flow or speed depending on how data is collected. The task is challenging due to complex spatial dependencies, non-linear temporal dynamics, and the inherent difficulty of long-term forecasting [1]. The structure of a road network can be easily interpreted as a graph, with nodes as intersections and edges as roads. Due to this situation, Graph Neural Networks(GNNs) seem to be the best tool for capturing the complex spatial dependencies found in road networks. A GNN is a type of neural network that works on graph data [2].

Traffic management agencies can use long-term traffic forecasting [3] to avoid potential traffic congestion, ensuring a smooth traffic flow. Furthermore, infrastructure works can be planned to interfere with traffic as little as possible [4]. Individuals can also greatly benefit from the advancement of such technologies, as apps such as Google Maps[1] allow trip planning that accounts for future traffic conditions. Therefore, the importance of long-term forecasting cannot be understated, as it helps cities to adapt to current and future demands, leading to a better quality of life for its residents.

The definition of long-term forecasting varies in the literature, with some considering any multi-step predictions to be long-term [5], meaning more than one horizon. A horizon is an integer time frame for predictions. Others mention predictions over half an hour to be long-term [6] and some say that one day to several is long-term [7]. To settle this the paper defines long-term forecasting as predictions that are **10 hours ahead** of the current time, which is roughly the commute time [8] plus the average working hours of a person in California [9].

The purpose of this paper is to evaluate the performance of GNNs for long-term traffic forecasting and address the research question: **Can GNNs effectively handle long-term predictions and how does their accuracy degrade over time?** Furthermore, the research question was divided into three sub-questions to accommodate more specific areas of study:

1. *Does the performance of the GNN noticeably degrade at specific points in time during long-term traffic forecasting?*
2. *Do fluctuations in traffic volume/speed contribute to the decline in the GNN's performance for long-term traffic forecasting?*
3. *Are there specific configurations of road networks (e.g. straight roads, multiple intersections) that contribute to the decline in the GNN's performance for long-term traffic forecasting?*

The paper is structured as follows, Section 2 details the related work on this subject and gives a formal description of traffic forecasting. In Section 3 the decision process of choosing a model and datasets. Section 4 describes the hardware used, how long-term forecasting is done on the chosen model, and the subsets used for the experiments. Furthermore, the results of the experiments are presented and discussed. Section 5 describes the ethical implications found during the experimentation part of the research. Lastly, in Section 6 conclusions are drawn out and possible future work is discussed.

2 Background

Most research on traffic forecasting focuses on developing models with better prediction performance. In contrast, the study of behaviour in specific situations is mentioned and sometimes looked into. However, it is not as thoroughly examined as the features added to improve results.

2.1 Related Work

One of the pioneering works in the field of GNNs for traffic forecasting is the paper "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting" (DCRNN) by Li *et al.* [1]. It proposes combining graph convolutional networks with recurrent neural networks in order to capture spatial and temporal dependencies from traffic data. Furthermore, the paper mentions that Deep Neural Network methods work better on long-term forecasting than linear baselines, such as the Auto-Regressive Integrated Moving Average model (ARIMA) [10] and the Vector Auto-regressive model (VAR) [11]. As mentioned in the DCRRN paper, this happens as the temporal dependencies become increasingly non-linear as the number of horizons increases.

Later iterations of GNNs have focused more on short-term forecasting rather than long-term, but there were efforts to overcome the challenge of long-term prediction. The paper entitled "GMAN: A Graph Multi-Attention Network for Traffic Prediction" by Zheng *et al.* [12] tackles two challenges

[1] https://www.google.com/maps

that hinder long-term predictions: complex spatiotemporal correlations and sensitivity to error propagation. To solve the mentioned challenges, they proposed two solutions for each issue: a spatial and temporal attention mechanism and a transform attention mechanism. The results presented in the paper were done with 12 horizons(1 hour), showing that their performance improved at later horizons. The solutions presented were later used by later models to tackle the challenge of long-term forecasting efficiently.

One of the best-performing models for traffic prediction, as reported by Papers with Code[2] and in the literature [13], is the D^{2}STGNN model (Decoupled Dynamic Spatial-Temporal Graph Neural Network) by Shao *et al.* [14]. The paper acknowledges two types of dependencies: short- and long-term. It proposes two separate solutions for each of them, respectively Gated Recurrent Units (GRUs) for short-term dependencies and a multi-head self-attention layer for long-term dependencies, which is a variation of the attention layer that GMAN uses. Similar to other papers, long-term forecasting is highlighted only when it boosts the model's overall performance and is not prioritized.

Other challenges are observed in the literature, such as external factors in "Road Traffic Forecasting: Recent Advances and New Challenges" by Laña *et al.* [15]. The factors that challenge long-term and short-term traffic predictions are road works, incidents, events, weather, proximity to traffic-affecting facilities (parking lots, shopping areas, work/study centres), and calendar matters (bank holidays, weekends).

2.2 Gaps in previous research

As mentioned in the introduction of this section, the work done in the literature focuses more on coming up with newer, more innovative models. It doesn't concentrate on studying the behaviour of more specific situations, such as long-term predictions using GNNs. Therefore, there is a noticeable gap in examining the behaviour of traffic forecasting models for longer horizons. This includes understanding the points in time where performance deteriorates, the effect of traffic jams on long-term forecasting, and the effect of road network configuration on long-term forecasting.

Furthermore, GNN models in literature often use only up to 12 horizons and there are rarely experiments with more horizons. This showcases the secondary nature of long-term predictions to short-term predictions in this field of study.

2.3 Formal Problem Description

This subsection aims to give a formal definition of the traffic forecasting task and its complementary elements, also the notations used are present in Table 1. The definitions are taken from the D^{2}STGNN paper [14].

Definition 1. ***Traffic Sensor.*** *A traffic sensor is a sensor deployed in a traffic system, such as a road network, and it records traffic information such as the number of passing vehicles or vehicle speeds.*

Definition 2. ***Traffic Network.*** *A traffic network is a directed graph G = (V,E), where V is the set of $|V|$ = N nodes and each node corresponds to a deployed sensor, and E is the set of $|E|$ = M edges. The reachability between nodes, expressed as an adjacent matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, could be obtained based on the pairwise road network distances between nodes.*

Table 1: Notations Used

Notation	Definitions
G	Graph representing the traffic network, defined as $G = (V, E)$ with node set V and edge set E.
N	Number of sensors (nodes) of the traffic network, i.e., $\|V\|$ = N.
$\mathbf{A}$	The adjacency matrix of traffic network G.
C	Number of feature channels in a traffic signal.
T_h	The number of past traffic signals considered.
T_f	Number of future time steps to predict in traffic forecasting.
$\mathbf{X}_t$	Traffic signal at time step t.
$\mathcal{X}$	Traffic signals of the $\mathbf{T}_h$ most recent past time steps.
$\mathcal{Y}$	Traffic signals of the $\mathbf{T}_f$ nearest-future time steps.

Definition 3. ***Traffic Signal.*** *The traffic signal $\mathbf{X}_t \in \mathbb{R}^{N \times C}$ denotes the observation of all sensors on the traffic network G at time step t, where C is the number of features collected by sensors.*

Definition 4. ***Traffic Forecasting.*** *Given historical traffic signals $\mathcal{X} = [\mathbf{X}_{t-\mathbf{T}_h+1}, \ldots, \mathbf{X}_{t-1}, \mathbf{X}_t] \in \mathbb{R}^{T_h \times N \times C}$ from the passed $\mathbf{T}_h$ time steps, traffic forecasting aims to predict the future traffic signals $\mathcal{Y} = [\mathbf{X}_{t+1}, \mathbf{X}_{t+2}, \ldots, \mathbf{X}_{t+T_f}] \in \mathbb{R}^{T_h \times N \times C}$ of the $\mathbf{T}_f$ nearest future time steps.*

3 Methodology

This section describes the selection procedure of elements needed for the experiments done during this research, such as the dataset, evaluation methods, and GNN model.

3.1 Dataset selection and description

Most of the datasets used in the papers about traffic forecasting on Papers with Code[3] are located in the United States, more specifically around the two major metropolitan areas in the state of California, the San Francisco Bay Area and Greater Los Angeles. The 4 most popular—METR-LA, PEMS-BAY, PEMS08, and PEMS04—were further analyzed to decide which would be used. Furthermore, the chosen datasets are split into 2 types of recorded data: traffic speed and traffic flow. The characteristics of the datasets are summarized in Table 2, showing the type, name, number of nodes, and time steps.

[2] https://paperswithcode.com/task/traffic-prediction

[3] https://paperswithcode.com/task/traffic-prediction

The coordinates of the sensors for two of the datasets, METR-LA and PEMS-BAY, are publicly available. This makes the selection of subsets with a specific road configuration possible. Furthermore, from the two, METR-LA is smaller, as it has fewer nodes and time steps. This makes it the obvious choice for a task such as long-term traffic forecasting, which has high computational costs. Therefore, the selected dataset is METR-LA.

Type	Dataset	Number of Nodes	Time Steps
Speed	**METR-LA**	207	34272
	PEMS-BAY	325	52116
Flow	PEMS04	307	16992
	PEMS08	170	17856

Table 2: Dataset types, number of nodes, and time steps that the 4 most popular datasets have.

METR-LA is a public traffic speed dataset with 207 loop detector sensors over 4 months from March 1st 2012 to June 27th 2012. The data was collected on the highways of Los Angeles County [16], and a map of the sensors can be seen in Figure 1. The interval at which the traffic data is recorded is 5 minutes, with a total of 34272 time steps.

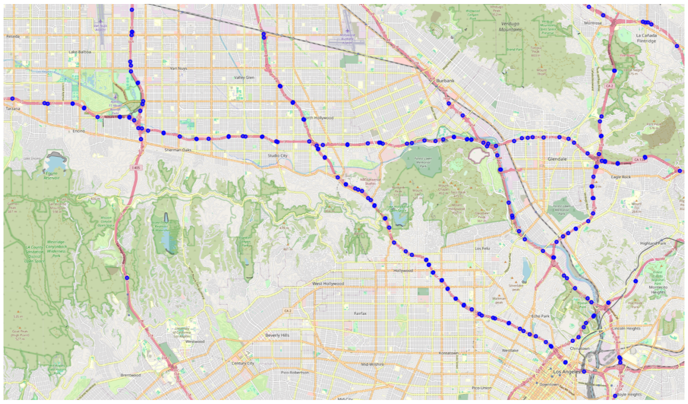

Figure 1: METR-LA sensors map.

3.2 Evaluation methods

To evaluate the performance of future experiments three commonly used metrics in traffic forecasting are utilized: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). The formulas for each are given below:

$$\mathrm{MAE}(x, \hat{x}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} |x_i - \hat{x}_i|, \tag{1}$$

$$\mathrm{RMSE}(x, \hat{x}) = \sqrt{\frac{1}{|\Omega|} \sum_{i \in \Omega} (x_i - \hat{x}_i)^2}, \tag{2}$$

$$\mathrm{MAPE}(x, \hat{x}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \frac{|x_i - \hat{x}_i|}{x_i}. \tag{3}$$

In the formulas x represents the ground truth and $\hat{x}$ represents the predicted values with Ω being the indices of the observed samples and $|\Omega|$ being equal to the number of future time steps predicted. Furthermore, the time to train and infer per epoch will be used to compare efficiency, this will be primarily used when choosing which model to use.

3.3 Selection of the GNN model and its description

The models presented in the literature were selected by looking at the section about traffic prediction on Papers with Code. Then, the papers presented were read through and the most relevant were selected for further analysis. DCRNN [1] was chosen as it was one of the pioneering models for traffic forecasting using GNNs. GMAN [12] was selected as it was one of the first GNN models to address the issue of long-term traffic forecasting and made it one of its main contributions. Lastly, D^{2}STGNN [14] was chosen as it is one of the best-performing models across multiple datasets and used methods to address long-term dependencies specifically.

The difficulty of setting up models varied. DCRNN has a comprehensive paper reproduction[4] that helped with setting up the model and also configuring it to run on more horizons. Even with this material, issues still prevailed, such as broken dependencies. GMAN was the most tedious to get to run, as the PyTorch [17] implementation that was given by the paper[5] has no instruction on how to run the model. The best option was the D^{2}STGNN model, which had a ReadMe file with every necessary detail for running it and a configuration file that was easily modifiable for every type of experiment that needed to be done.

From a performance point of view, the D^{2}STGNN model is the most efficient and the best performing out of the 2 models. The results are extracted from "Unified Data Management and Comprehensive Performance Evaluation for Urban Spatial-Temporal Prediction [Experiment, Analysis & Benchmark]" by Jiang *et al.* and they are summarized in Tables 3 and 4.D^{2}STGNN performs the best and is also the most adaptable. Therefore, making it the best choice for the experiments in this paper.

Performance Comparison				
Datasets	Methods	MAE	RMSE	MAPE
METR-LA	DCRNN	3.16	6.44	8.66%
	GMAN	3.16	6.43	8.65%
	D^{2}STGNN	**2.91**	**5.84**	**7.93%**

Table 3: Performance results for METR-LA.

3.4 Description of the GNN model

The architecture of the D^{2}STGNN model can be found in Figure 2, which begins with taking as input the historical data of

[4] https://dcrnn-reproduced-paper.notion.site/Diffusion-Convolutional-Recurrent-Neural-Network-Data-Driven-Traffic-Forecasting-A-Paper-Reproduc-85653c40503d4075a5be4c433c6f2f23#4045804f686548d3899b39ceb0d696c1

[5] https://github.com/VincLee8188/GMAN-PyTorch

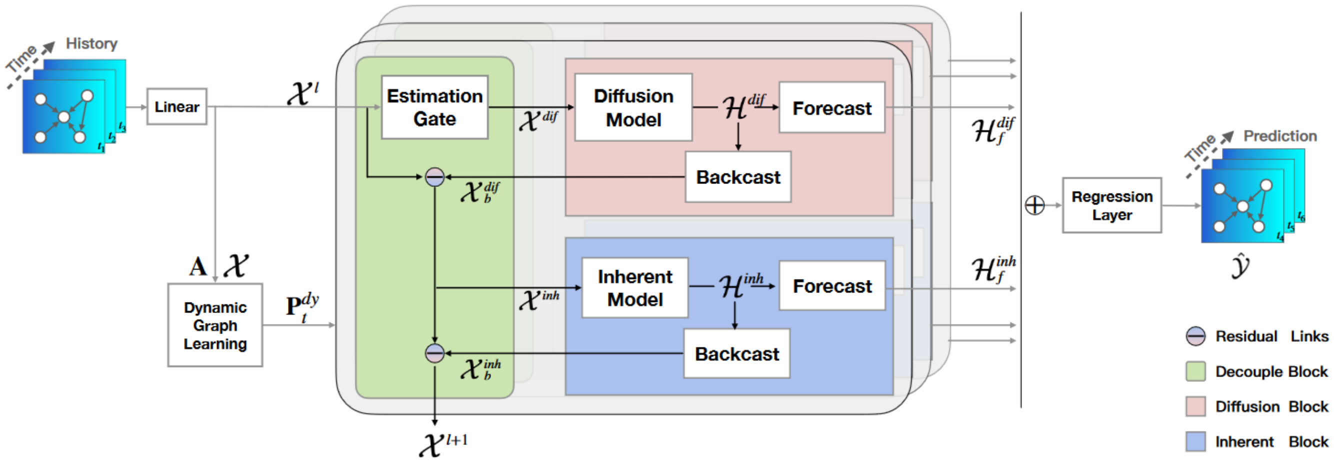

Figure 2: The overall architecture of D^{2}STGNN [14].

Efficiency Comparison		
Datasets	Methods	Time Per Epoch
METR-LA	DCRNN	605.88s
	GMAN	738.24s
	D^{2}STGNN	**259.78s**

Table 4: Efficiency results for METR-LA.

$\mathcal{X}$ as input. This data is passed to the *dynamic graph learning module* along with the adjacency matrix $\mathbf{A}$, which computes the dynamic transition matrices $\mathbf{P}_t^{dy}$. The *estimation gate* splits $\mathcal{X}_i$ into its diffusion components $\mathcal{X}^{dif}$ and inherent components $\mathcal{X}^{inh}$. The *diffusion module* processes $\mathcal{X}^{dif}$ using a graph neural network to capture diffusion dependencies for both forecasting and backcasting. Backcasting is used to trace back to derived historical patterns which help in forecasting. While the *inherent module* does the same for $\mathcal{X}^{inh}$, it also incorporates a self-attention mechanism.

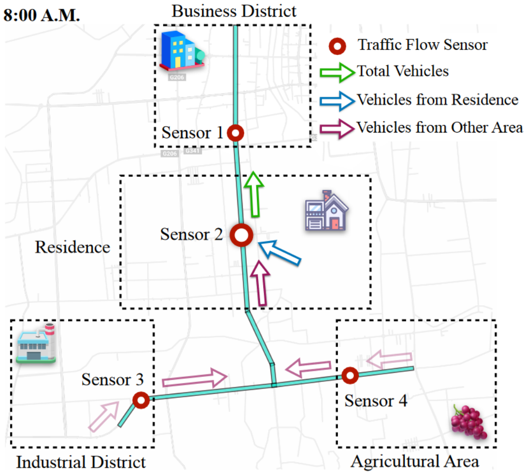

Figure 3: Traffic network system [14].

Figure 3 highlights the difference between inherent and diffusion data. In sensor 2 there are two types of traffic detected, the one from the residential area marked with the blue arrow and the one from other areas marked with the red arrow. The traffic from the residential area is independent and represents an inherent signal. The traffic from the other areas is influenced by regions, such as the Industrial District or the Agricultural Area, representing a diffusion signal.

Finally, the regression layer transforms the combined features from the diffusion and inherent model into the final prediction of the model.

4 Experimental Setup and Results

This section gives a detailed description of the experimental environment, while also presenting some challenges faced along the way. After this, the results of the experiments are presented.

4.1 Setup

Hardware

The experiments were conducted on DelftBlue [18], TU Delft's supercomputer. The node used for this task had 8 Intel XEON E5-6448Y 32C 2.1GHz CPU cores running with 16GB of memory per node and an NVIDIA Tesla A100 80GB graphics card.

Long Term Forecasting

D^{2}STGNN preprocesses data by splitting it into 2-hour windows, from which the first hour is used as input for forecasting and the second hour is used as a ground-truth for testing purposes. The window is composed of 24 time-segments of 5 minutes each, in this case, the model forecasts 12 horizons into the future or 1 hour into the future. The straightforward way of predicting long-term was to increase the number of segments taken in by the input and the output. This method can be seen in Figure 4 on the first row, where the boxes represent the input of 10 hours in the past used to predict 10 hours into the future, with the output as the ground truth. Unfortunately, this drastically increased the computational time.

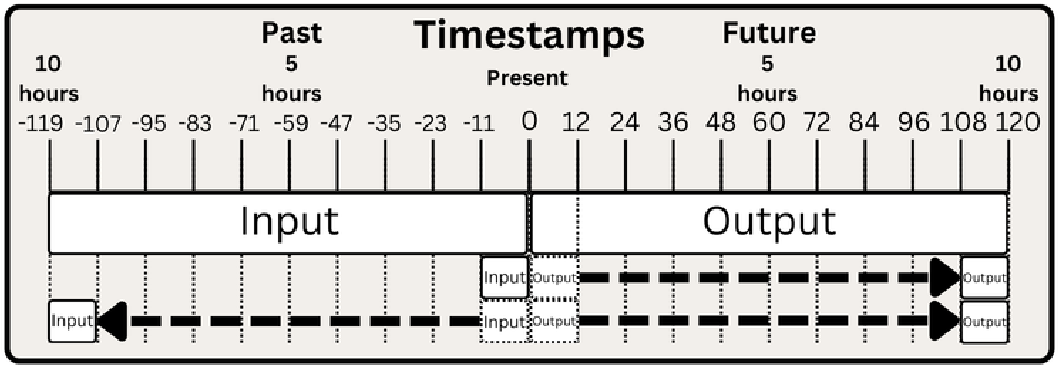

Figure 4: D^{2}STGNN inputs for long-term forecasting on the first row and parallelization trials on the second and third row

Exploratory experiments show the run-time of training the model for a 10-hour prediction on a subset of 20 nodes of METR-LA results in a run time of 13 hours for 100 epochs while for a 1-hour prediction, it had a run time of 30 minutes for 100 epochs.

To mitigate the issue of computation time two parallelization methods were attempted. In the first method, the input and output had the size of 12 time-segments, meaning one hour. The input remained in place but the output was moved further into the future, as seen in Figure 4 on the second row. This method would work for predicting 2 hours into the future but would give poor results at anything above that.

In the second attempt, the input and the output were moved progressively. Therefore, if the prediction needed to be made 10 hours into the future, then the input was moved 10 hours into the past and the output was moved 10 hours into the future, as seen in Figure 4 on the third row. Unfortunately, this method would give the same results as the previous one, rendering it unusable. Therefore, the original method was used going forward, even with its high computational cost.

Subsets

As the computational costs couldn't be brought down by parallelizing the forecasting task, the next option was to divide the original dataset into subsets. The sensor maps of each subset can be found in Appendix A. Furthermore, the subsets were created to show different types of road configurations inside the original dataset, helping in answering the third sub-question:

1. One intersection with 20 nodes
2. Two intersection with 40 nodes
3. Three intersections with 55 nodes
4. Straight road with 35 nodes

Run configuration

The configuration of running the model was almost the same as the one presented by its paper [14]. The main changes were made to the embedding size of nodes and time slots. They were set to 120 from 12, to accommodate long-term forecasting of 10 hours from 1 hour. Lastly, the number of epochs to be trained on increased from 80 epochs to between 350 and 450, depending on the size of the subset.

4.2 Results

Error evolution over horizons.

The MAE, RMSE, and MAPE are normalized using min-max normalization to make the errors comparable. Then they are plotted over the horizons to highlight the model performance as it predicts further into the future. A first example is Figure 5, which shows a logarithmic growth in the errors up to horizon 80 in the one intersection subset. Afterwards, there is quite a sharp increase until the last horizon. This could be due to the need for more training epochs for the model to achieve logarithmic growth over all horizons. To check this, a logarithmic function $a \times \log_{10}(x) + b$ is defined to represent the potential logarithmic growth pattern of the errors. Then curve fitting techniques are used to find the best fitting parameters for this function. To quantify how good the fit is the R^2 value is calculated, which as it gets closer to 1 the better the fit. It uses the following formula:

$$R^2 = 1 - \frac{\sum_i (y_{\text{actual},i} - y_{\text{predicted},i})^2}{\sum_i (y_{\text{actual},i} - \bar{y}_{\text{actual}})^2} \quad (4)$$

Where $y_{\text{actual},i}$ is the i^{th} actual normalized error, $y_{\text{predicted},i}$ is the i^{th} predicted normalized error generated by the logarithmic function defined earlier, and $\bar{y}_{\text{actual}}$ is the mean of the actual normalized error.

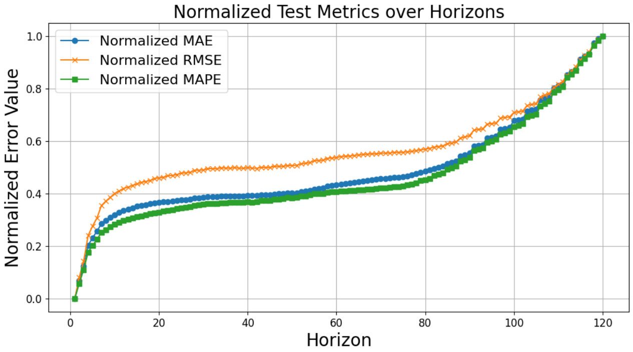

Figure 5: The one intersection subset with 20 nodes at epoch 450 over 120 horizons.

Starting from the 100th epoch and using an interval of 50 epochs the R^2 value is calculated. This shows the conver-

Epoch	MAE	RMSE	MAPE
100	0.24	0.20	0.23
150	0.37	0.47	0.40
200	0.59	0.41	0.62
250	0.57	0.43	0.60
300	0.53	0.49	0.56
350	0.58	0.68	0.57
400	0.55	0.62	0.56
450	0.63	0.74	0.62

Table 5: R^2 over the epochs on the one intersection subset with 20 nodes and 120 horizons.

Epoch	MAE	RMSE	MAPE
100	0.35	0.30	0.43
150	0.60	0.66	0.68
200	0.66	0.76	0.63
250	0.66	0.77	0.58
300	0.75	0.87	0.66
350	0.88	0.94	0.95
400	0.91	0.92	0.95
450	0.93	0.92	0.96

Table 6: R^2 over the epochs on the one intersection subset with 20 nodes and 80 horizons.

gence of the R^2 error towards 1, indicating that the errors over the horizons settle to a logarithmic shape. The results can be seen in Table 5. Due to the sharp rise in the values of the errors at later horizons, the improvements over the epochs are not noticeable. If the number of horizons is reduced from 120 to 80 in order to eliminate the outlying values at later horizons, the improvements over the epochs are more noticeable, as the curves of the errors converge faster to the logarithmic one. The results for 80 horizons are summarized in Table 6. The R^2 values improve noticeably until epoch 350 after which they plateau.

The logarithmic growth can also be seen in short-term predictions of 1 hour on the one intersection subset and it is not specific for long-term forecasting. The R^2 values after 100 epochs on the for the errors are almost perfect: MAE - 1.00, RMSE - 1.00, and MAPE - 0.99. The lower number of epochs is due to fewer horizons needing to be trained. The plots showing the errors over 12 horizons for the subsets can be found in Appendix B.

This behaviour was the same for most subsets except for the straight road subsets. Which, had two peaks in R^2 values, at epoch 250 and epoch 380. These results can be seen in Table 7. The first decline could be attributed to how the model trains over time, but the second one is due to overfitting. Furthermore, even in short-term forecasting, this subset was prone to overfitting compared to the others. The plots showing the errors for this subset can be found in Appendix C.

In the literature, the same logarithmic growth trend was observed for the PEMS datasets [12]. However, the trend is not observed for a dataset located in the city of Xiamen in China, which records traffic volume instead of traffic speed. The nature of the datasets could have a bigger impact on how errors evolve than the underlying characteristic of GNNs. Therefore, the logarithmic growth trend observed in the experiments could be due to the nature of the METR-LA dataset.

Epoch	MAE	RMSE	MAPE
250	0.62	0.77	0.72
300	0.43	0.52	0.55
380	0.59	0.73	0.64
450	0.40	0.52	0.59

Table 7: R^2 over the epochs on the straight road subset with 35 nodes and 120 horizons.

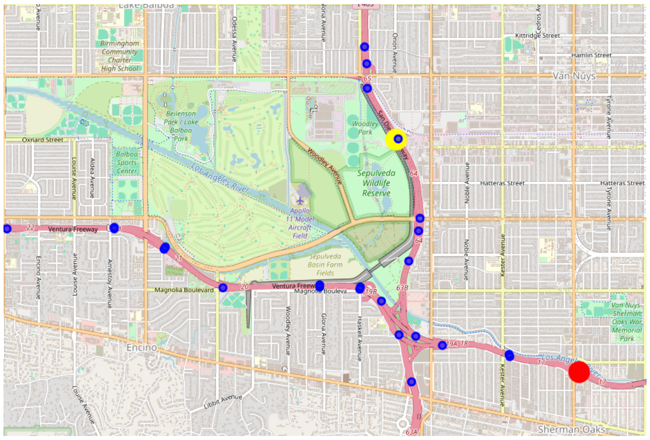

Figure 6: The location of nodes 717499 and 767350 on the map, with colours red and yellow respectively.

Predictions around traffic jams and anomalies.

Traffic jams and anomalies can cause disturbances in traffic forecasting, but some types cause more disruptions than others. To showcase this, two plots have been created depicting the real values and the predicted values from two separate nodes in the road network, highlighted in red and yellow in Figure 6. For nodes that have traffic jams at regular intervals the impact on the performance of the prediction is not significant until later horizons. This can be clearly seen in Figure 7, which shows that a recurrent traffic jam can be predicted quite accurately until horizon 96, showing issues at horizon 120. The traffic jam still causes issues at later horizons, such as 96 and 120, where predicted values deviate from the real values. This is in contrast with periods when traffic values don’t fluctuate, as even at the last horizon the predictions are close to the real values.

Another type of traffic jam, which lacks the well-defined pattern discussed in Figure 7, can have disruptive effects on predictions. Therefore, various results can be seen when forecasting such an event. The model can completely ignore it and predict a continuation of traffic values or it can broadly forecast a slowdown in speed. This can be seen in Figure 8, where, from horizon 24 to horizon 96, the model predicts an overall slowdown in traffic speed. While in horizon 120,

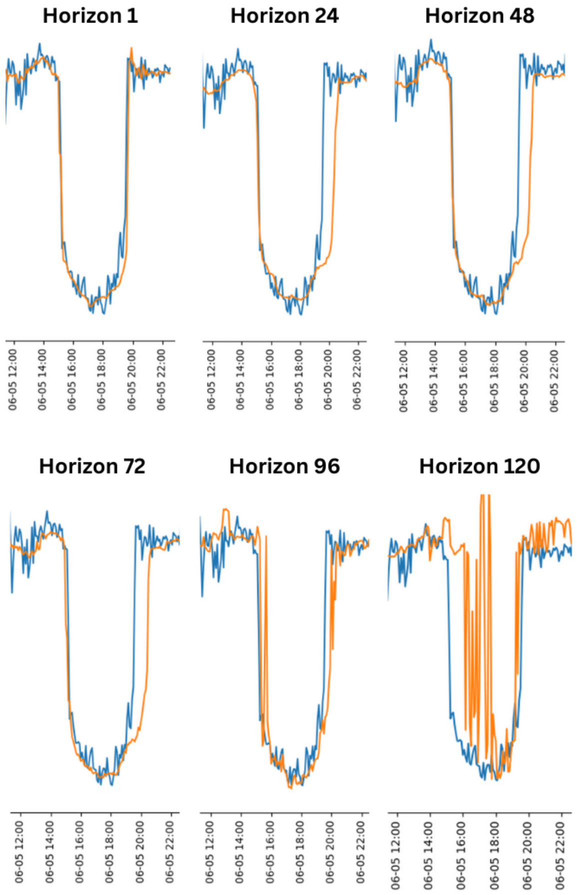

Figure 7: Real speed values (blue) and predicted speed values (orange) on the 6th of June 2012 for the one intersection subset on sensor 717499.

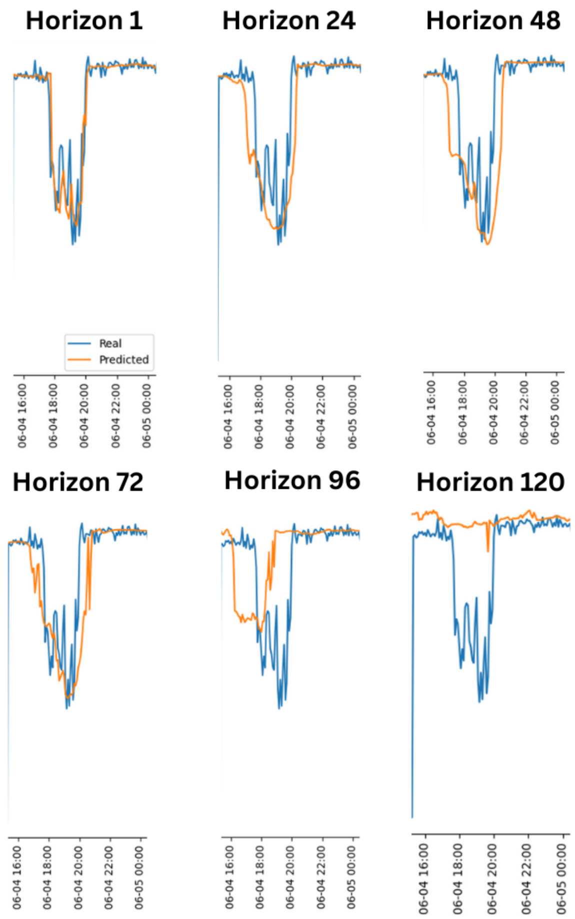

Figure 8: Real speed values (blue) and predicted speed values (orange) on the 6th of June 2012 for the one intersection subset on sensor 767350.

it predicts no change in the traffic speed even when the real data shows a reduction in it. This behaviour could be explained by the inability of the model to adapt to unforeseen traffic conditions that were not present in the training data. Furthermore, traffic jams propagate non-linearly through traffic systems [19], creating complex spatial and temporal patterns that are challenging to tackle. The D^{2}STGNN paper mentions that the methods used still fail to exploit these complex patterns fully [14]. Therefore, the presence of sudden traffic jams poses a great challenge to traffic forecasting and the inherent difficulty of long-term forecasting increases the complexity of this challenge.

Effects of different road network configurations.

The test MAE of the subsets was recorded at epoch 320 over 120 horizons. The summary can be found in Table 8. The results show that the best-performing subset was the two intersections subset, while the worst-performing was the three intersections subset. This matches with results from the literature [20], as more nodes don't necessarily translate to better performance.

Other factors might have an impact on forecasting performance. Therefore, the mean, variance, standard deviation, and the percentage of missing values of the subsets were calculated to see if they have a role. The procedure was as follows: each metric was calculated for each node in the dataset and then averaged for each node to arrive at a result for the whole subset. The results can be seen in Table 9, where the three intersections subset has the lowest variance and standard deviation, while it has the highest percentage of missing values.

The results from Tables 8 and 9 seem to be correlated, as the percentage of missing values goes up the performance starts to decline. The literature confirms this as it states that graph learning [21] and dynamic graph learning [22] are affected by missing values. As D^{2}STGNN uses a dynamic

Subset	MAE
Straight Road - 35 nodes	7.53
One Intersection - 20 nodes	7.78
Two Intersections - 40 nodes	6.29
Three Intersections - 55 nodes	10.15

Table 8: Subsets and their MAE values at epoch 320 averaged over 120 horizons.

Subset	Mean	Var.	Std. Dev.	Missing Values %
Straight Road	56.23	226.08	14.16	7.54
One Intersection	53.84	213.17	14.22	7.8
Two Intersections	55.55	190.75	13.21	7.44
Three Intersections	61.39	65.47	6.96	8.92

Table 9: Subsets and their mean, variance, standard deviation, and # Zeros.

graph learning module [14], it is clear that the missing values impact the performance. Therefore, the road network configuration might influence long-term traffic forecasting. However, due to other factors, such as data variation and missing values, it is uncertain how much of an impact it has.

5 Responsible Research

In traffic forecasting, ethical concerns often centre on the datasets used by the models. This paper uses METR-LA, a public dataset that contains traffic data collected from loop detectors on the highways of Greater Los Angeles. The technology used to collect this information does not infringe on privacy, as it only records the speed of the vehicle passing through it. Therefore, no identifiable information is recorded, making it impossible to trace back to an individual.

Regarding the reproducibility of the experiments the model used in the paper has a degree of randomness, as a random seed is initialized each time. Therefore, every run would produce slightly different results, but not different enough to disprove the findings in this paper. For example, the Mean Absolute Error could differ from one run to another by a maximum of 0.01, which is negligible. Therefore, following the paper, the results can be reproduced within negligible margins.

6 Conclusions and Future Work

6.1 Conclusions

This paper discussed the performance of GNNs in long-term traffic forecasting, focusing on how long-term predictions are handled and how their accuracy degrades over time. Furthermore, several factors, such as traffic jams and road network configurations, are looked into. Their influence on long-term forecasting accuracy is measured and discussed.

First, as time increases, the errors of the predictions follow a logarithmic growth and converge to it over the training epochs. This was also partly observed in the literature for short-term traffic forecasting [12], therefore the paper confirms the same evolution for long-term predictions.

Second, traffic jams have a big impact on the performance of long-term forecasting, but not all types. Regular daily traffic jams do not cause issues only until the last horizons. Sudden and unforeseen traffic jams heavily impact the results, as they do not correspond with the training data and they also introduce complex spatial and temporal patterns increasing the complexity of the task. Compared to short-term forecasting, long-term includes its complexity and therefore the issues caused by traffic jams are amplified.

Third, road network configurations play a role in the performance of long-term predictions, but other factors also influence this task. Missing values seem to be directly correlated with how well a subset performs, also factors like variance and standard deviation impact the outcome. Therefore, it is uncertain what impact different road network configurations have, as they are tied to other factors.

6.2 Future Work

Looking ahead, the first area in which future work could be done is to build subsets isolated from other factors to have a proper comparison. This would provide a concrete answer to the third sub-question.

Furthermore, the experiments could be done on datasets that record traffic volume instead of traffic speed, to see if the results are independent of the data recorded. This is important because the results seen in the GMAN paper [12] suggest there might be a difference between the two types of data. However, other results in the literature for the PEMS traffic volume datasets show a similar behaviour [22] to what was found in this paper. Additionally, external factors, such as those mentioned in Section 2.1 could be accounted for through data fusion, as recommended in the literature [15].

Finally, datasets that are recorded on roads different from highways could be used to see if there is a difference between road types. Also, datasets from different regions would be useful to see if geolocation impacts long-term traffic forecasting.

References

[1] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Graph convolutional recurrent neural network: Data-driven traffic forecasting," *CoRR*, vol. abs/1707.01926, 2017. [Online]. Available: http://arxiv.org/abs/1707.01926

[2] L. Wu, P. Cui, J. Pei, and L. Zhao, Eds., *Graph neural networks: Foundations, frontiers, and applications*, 1st ed. Singapore, Singapore: Springer, Jan. 2022. [Online]. Available: https://doi.org/10.1007/978-981-16-6054-2

[3] P. Næss and A. Strand, "Traffic forecasting at 'strategic', 'tactical' and 'operational' level," *disP - The Planning Review*, vol. 51, no. 2, pp. 41–48, 2015. [Online]. Available: https://doi.org/10.1080/02513625.2015.1064646

[4] K. Jha, N. Sinha, S. Arkatkar, and A. Sarkar, "A comparative study on application of time series analysis for traffic forecasting in india: Prospects and limitations," *Current Science*, vol. 110, p. 373, 02 2016. [Online]. Available: https://www.currentscience.ac.in/Volumes/110/03/0373.pdf

[5] N. Cini and Z. Aydin, "A deep ensemble approach for long-term traffic flow prediction," *Arabian Journal for Science and Engineering*, Jan 2024. [Online]. Available: https://doi.org/10.1007/s13369-023-08672-1

[6] Y. Wang, Q. Ren, and J. Li, "Spatial–temporal multi-feature fusion network for long short-term traffic prediction," *Expert Systems with Applications*, vol. 224, p. 119959, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742300461X

[7] Z. Hou and X. Li, "Repeatability and similarity of freeway traffic flow and long-term prediction under big data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1786–1796, 2016. [Online]. Available: https://doi.org/10.1109/TITS.2015.2511156

[8] "Average commute time in california." [Online]. Available: https://fred.stlouisfed.org/series/B080ACS006037

[9] "Average daily workin hours in california." [Online]. Available: https://www.bls.gov/sae/tables/annual-average/table-3-average-hours-and-earnings-of-production-employees-on-manufacturing-payrolls-by-state.htm

[10] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*, ser. Holden-Day series in time series analysis and digital processing. Holden-Day, 1970. [Online]. Available: https://books.google.nl/books?id=5BVfnXaq03oC

[11] J. Hamilton, *Time Series Analysis*, ser. Book collections on Project MUSE. Princeton University Press, 1994, no. v. 10. [Online]. Available: https://books.google.nl/books?id=B8_1UBmqVUoC

[12] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, p. 1234–1241, Apr. 2020. [Online]. Available: http://dx.doi.org/10.1609/aaai.v34i01.5477

[13] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Unified data management and comprehensive performance evaluation for urban spatial-temporal prediction [experiment, analysis benchmark]," 2024. [Online]. Available: https://arxiv.org/abs/2308.12899v3

[14] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, and C. S. Jensen, "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," 2022. [Online]. Available: https://arxiv.org/abs/2206.09112

[15] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: Recent advances and new challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018. [Online]. Available: https://doi.org/10.1109/MITS.2018.2806634

[16] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Communications of the ACM*, vol. 57, no. 7, p. 86–94, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1145/2611567

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[18] Delft High Performance Computing Centre (DHPC), *DelftBlue Supercomputer (Phase 2)*, 2024, https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.

[19] D. Helbing, "Traffic and related self-driven many-particle systems," *Reviews of Modern Physics*, vol. 73, no. 4, p. 1067–1141, Dec. 2001. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.73.1067

[20] X. Liu, Y. Xia, Y. Liang, J. Hu, Y. Wang, L. Bai, C. Huang, Z. Liu, B. Hooi, and R. Zimmermann, "Largest: A benchmark dataset for large-scale traffic forecasting," 2023. [Online]. Available: https://arxiv.org/abs/2306.08259

[21] F. Li, J. Feng, H. Yan, G. Jin, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," 2021. [Online]. Available: https://arxiv.org/abs/2104.14917

[22] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9346058

A Subsets METR-LA

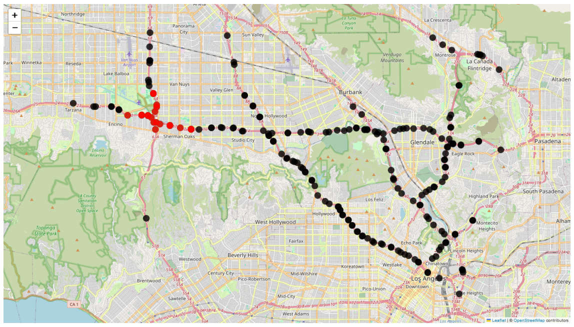

Figure 9: Nodes selected for the one intersection subset.

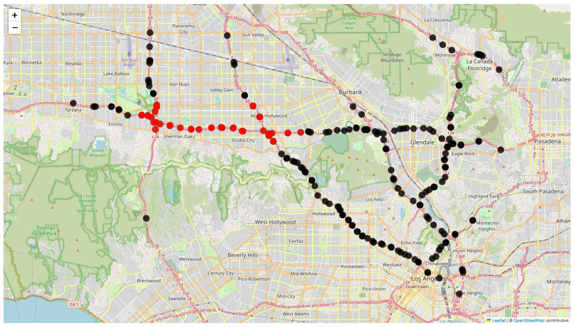

Figure 10: Nodes selected for the two intersections subset.

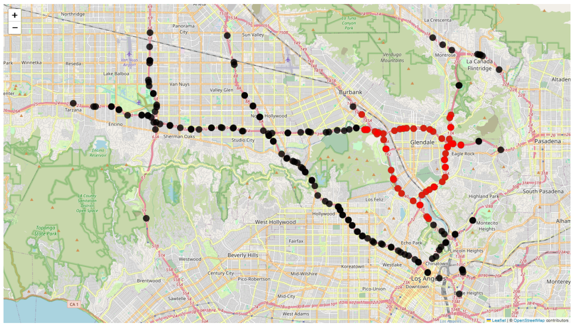

Figure 11: Nodes selected for the three intersections subset.

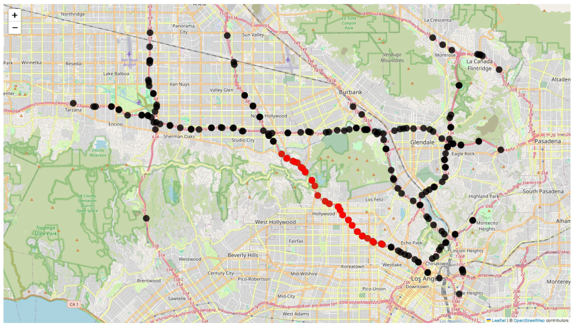

Figure 12: Nodes selected for the straight road subset.

B Error plots over 12 horizons for all subsets

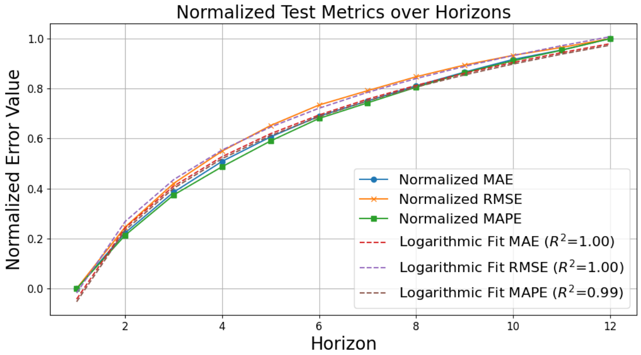

Figure 13: The one intersection subset with 20 nodes at epoch 100 over 12 horizons.

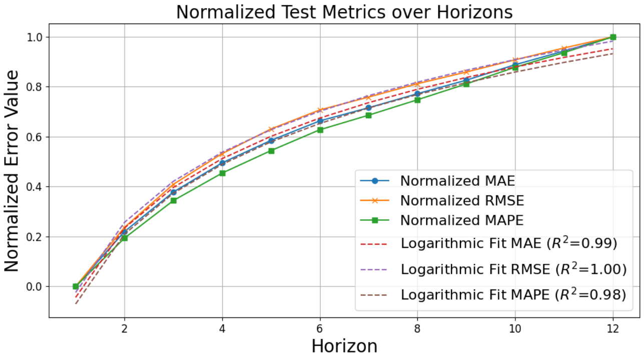

Figure 14: The two intersections subset with 40 nodes at epoch 100 over 12 horizons.

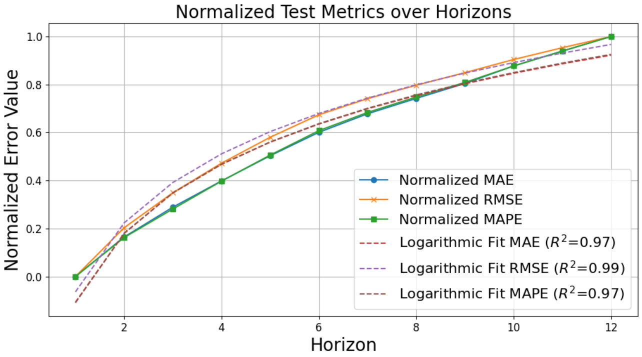

Figure 15: The three intersections subset with 55 nodes at epoch 100 over 12 horizons.

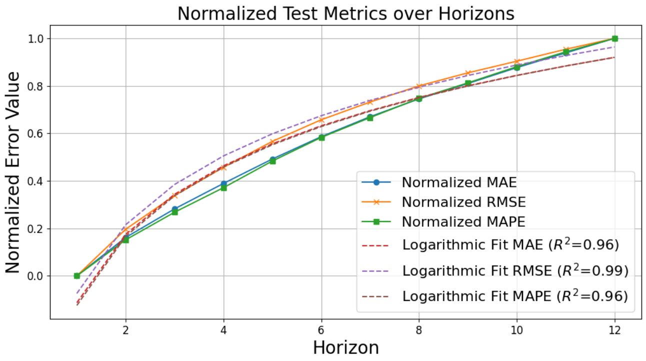

Figure 16: The straight road subset with 35 nodes at epoch 100 over 12 horizons.

C Error plots over 120 horizons for the Straight Road subset

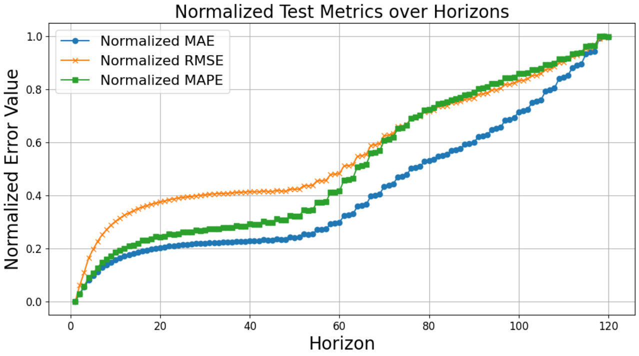

Figure 17: The straight road subset with 35 nodes at epoch 250 over 120 horizons.

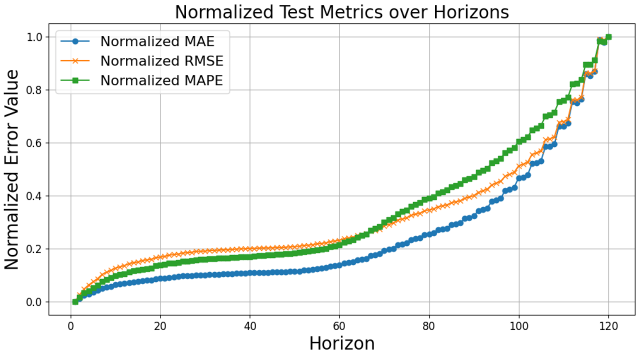

Figure 18: The straight road subset with 35 nodes at epoch 300 over 120 horizons.

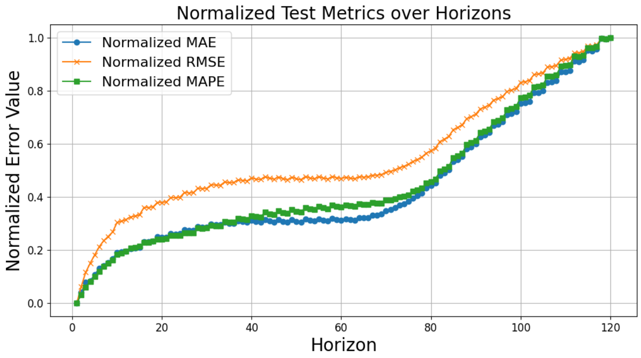

Figure 19: The straight road subset with 35 nodes at epoch 380 over 120 horizons.

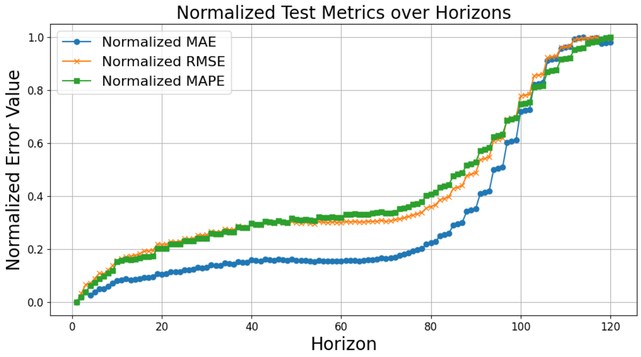

Figure 20: The straight road subset with 35 nodes at epoch 450 over 120 horizons.

D Use of LLMs in the research paper

While writing the paper, LLMs were used to rephrase sentences and give feedback on paragraph structure. The prompts used were: ”Please give feedback on the following paragraph: (...)” and ”Please rephrase the following sentence: (...)”.