Regional Transferability of Graph Neural Networks for Traffic Forecasting

Ivans Kravcevs

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 students: Ivans Kravcevs
Final project course: CSE3000 Research Project
Thesis committee: Elena Congeduti, Lilika Markatou

Abstract

Efficient traffic forecasting is an important component of modern traffic management systems, enabling real-time route guidance and traffic control. Graph Neural Networks (GNN) have demonstrated state-of-the-art performance in this domain due to their ability to capture spatial and temporal dependencies in complex traffic data. However, GNNs typically require extensive historical data and are highly dependent on the specific road structure of the training region, posing challenges for their application in areas lacking such data. This study explores the transferability of GNN models in traffic forecasting, specifically how a GNN, trained in the region with long-horizon historical data, performs when applied to structurally different regional scenarios without historical data. The research investigates the impact of spatial differences between regions on the model's performance. The paper examines multiple metrics for regional similarity between training and transfer regions and shows their correlation with the transferred model's performance.

1 Introduction

Modern traffic management systems require traffic forecasting tools for effective work. Short-term traffic forecasting can be used for real-time route guidance and traffic control [1]. The main function of traffic forecasting models is to predict the future traffic situation from a few seconds to several hours based on historical traffic data [2].

Innovative deep neural network technologies can achieve good performance in traffic forecasting. Graph Neural Networks (GNN) show state-of-the-art performance in traffic forecasting according to Jiang & Luo's survey on traffic forecasting [3]. The study noted that GNN can be applied for various traffic forecasting tasks and maintain the best short-term prediction performance due to GNN's ability to capture both spatial and temporal data dependencies. However, GNNs are strongly dependent on the road structure of the training region and require long-horizon historical data to train.

Collecting consecutive historical traffic data in the region is a complicated and costly process. It includes creating a network of sensors and their maintenance for a long period. The collected data should also be clean, and contain a minimum amount of incomplete or corrupted data, to make traffic forecasting effective. The transferable traffic forecasting model can resolve the issue of long historical data collection for model training. Transferability is the ability to gain knowledge in one domain and reuse it in another domain [4].

Transferability in traffic forecasting involves model training on one traffic region and deploying it for traffic prediction in another area. Effective transferability can help one pre-trained traffic forecasting model be used on multiple traffic regions without extensive direct training over each of the regions.

This project investigates how a pre-trained graph neural network model, originally developed for traffic forecasting of a specific region, performs when applied to different regional scenarios. The research explores the impact of GNN's strong dependency on regional spatial data on the model's transferability.

The research includes the comparison of multiple common traffic forecasting models. It uses Diffusion Convolutional Recurrent Neural Network (DCRNN) [5] as the experiment model. DCRNN is a well-performing GNN trained directly on the spatial dependencies of the graph. It is also used in multiple studies related to transferability in traffic forecasting [6, 7].

This research paper explores the relationship between the differences in graph representations of traffic networks and the performance of the DCRNN model in these regions. It uses multiple distance metrics between the adjacency matrices of the graph as the measure for road network spatial differences.

Insights from this research are helpful for the development of more adaptable and transferable GNN models for traffic forecasting in data-scarce regions. They are also useful for GNN traffic forecasting model applications in data-scarce regions.

The research question guiding this study is:

How does the GNN traffic forecasting model, trained with long-horizon historical data from one traffic scenario, perform in regions lacking historical traffic data, and how are these performance variations correlated with spatial differences among the regions?

The main research question is further divided into the following sub-questions:

- *What is the performance of the GNN model in the traffic forecasting of the training region?*
- *What is the performance of the same model transferred to different unexplored regions?*
- *How does the structural difference between training and transfer regions correlate with the model's performance in the transfer region?*

The research finds that the model performance varies based on the selected region. It also shows that the models with a good performance over the training region often perform badly in transferability tasks. It concludes that the structural difference between training and transfer regions is weakly correlated with the model's performance.

The structure of the paper is as follows: Section 2 describes the background of the research. It shows the related literature and formally defines the problem. Section 3 describes the methods used to complete the experimental work. It gives an overview of used data and metrics. It also compares models and describes the graph selection procedure. Section 4 explains the model training and transfer process and obtained performance results. Section 5 describes the experiments exploring the correlation between regional structure and model transferability. Section 6 reflects on the ethical aspects of the research paper. Section 7 evaluates the obtained results and discusses the limitations of the current work. Section 8 summarizes the findings obtained during the research and gives the potential future improvements.

2 Background

This section gives an overview of the literature about traffic forecasting, graph neural networks for traffic forecasting, and transferability possibilities for GNNs. It also formally defines the problem that is researched in the experimental work.

Related Work

Traffic forecasting is a longly-studied research area. Poor traffic flow prediction is still one of the biggest issues in implementing advanced traffic management systems [8].

Jiang and Luo [3] provide a comprehensive literature survey on traffic forecasting using GNNs. The authors describe the similarities between traffic road structure and graph structure. They state that GNNs show state-of-the-art performance on traffic flow prediction problems and many other traffic forecasting problems. The paper also specifically declares Diffusion Convolutional Recurrent Neural Network to be one of the best-performing traffic forecasting models that successfully capture traffic data complex spatial and temporal dependencies.

Li et al. [5] introduce DCRNN model architecture. They also show the model's performance on two traffic forecasting datasets popular for benchmarking (METR-LA and PEMS-BAY). The authors compare the model performance with multiple previously known models (such as Long Short-Term Memory (LSTM) [9], feedforward Neural Network (FNN) [10]). According to the paper, the DCRNN model performs better than the other models. However, several recent papers have introduced better-performing models.

Shao et al. [11] describe the Decoupled Dynamic Spatial-Temporal Graph Neural Network (D2STGN) model, which shows better performance than DCRNN on the benchmarking datasets. The authors introduce the problem of the previous model's dependency on the static graph adjacency matrix, which restricts the ability to represent complex road structures. The paper presents a dynamic graph learning block that learns the transition matrices based on static, dynamic, and time information in historical traffic data.

A recent study by Lablack and Shen [12] presents a current state-of-the-art model for the METR-LA dataset - Spatio-Temporal Graph Mixformer (STGM). However, the D2STGN model slightly outperforms it on PEMS-BAY dataset. The authors use similarity learning and transformer architecture in the model to achieve the performance.

Jiang et al. [4] introduce the idea of transferability in neural networks. The paper also gives a formal definition of transferability as a task. It also describes the main stages of transferability tasks in deep learning of pre-training, adaptation, and evaluation. The current study follows the transferability steps introduced in this paper: the model is firstly pre-trained on the chosen dataset, the adaptation step involves domain adaptation to fit the pre-trained model, and the evaluation step is based on model performance evaluation on the adapted domain.

Mallick et al. [6] explain the possibilities of transfer learning regarding traffic forecasting. They introduce a way to make DCRNN transferable to other regions. They also compare the performance of different transferred models in traffic forecasting. They introduce the assumption that GNN prediction performance is strongly related to the road sensor graph structure regarding the transferability problem. The current study aims to investigate this assumption and show the correlation between the sensor graph structures and the performance of the transferred model.

The current study is not performing the transfer learning as it is described in the Mallick et al. paper [6], the train region is not divided into multiple subgraphs and trained on all of them, to keep the training graph structure static and usable for comparison. It uses smaller training and evaluation region subsets, due to data availability and to reduce model training and testing times. However, the current study follows the other steps of data preparation, model training, and evaluation, introduced in the paper. The obtained results are also compared with the given by Mallick et al. [6] results in Section 4.

Formal Problem Description

Traffic forecasting aims to predict traffic conditions for a given time series. According to Li et al. [5], the traffic forecasting problem includes predicting the future traffic speed in the region based on the previously observed traffic flow. The traffic forecasting models take a sequence of historical traffic signals from the region and map it to future signal predictions for a specific amount of timestamps, called forecast horizon.

The traffic region is represented by a set of n speed-detecting sensors placed on the region's main road. The region can be described as a weighted directed graph $G = (V, E)$. V represents the set of sensors, and weighted edges E represent the road distances between sensors in graph G. This research will describe the traffic region graph as $G = (V, W)$, where $W \in R^{n \times n}$ is a weighted adjacency matrix representing road distances between the sensors. Each row of the matrix represents the outgoing distances from a specific sensor, and each column vector represents the incoming distances to that sensor.

The GNN models take the graph with observed sensor data for T' historical timestamps as the input and produce the predictions for the T future timestamps. The model's performance is measured by the comparison of $y = \{y_1, \ldots, y_T\}$ representing the ground truth values and $\hat{y} = \{\hat{y}_1, \ldots, \hat{y}_T\}$ representing the predicted values for each sensor. The measurement metrics are described in Section 3. The formal description of the traffic forecasting model is based on the text of the paper by Li et al. [5].

The GNN models can potentially be used in transferred problem scenarios. The idea of transferability in traffic forecasting is the possibility of training the model on one set of sensors G_{train} and the model application on one of the other geographically different regions $G_{test} \in \{G_{\text{test}_1}, G_{\text{test}_2}, \ldots, G_{\text{test}_n}\}$. Since GNN models operate directly on the graph values, it is assumed that the model's transferability performance is strongly dependent on the local structure of the training graph G_{train}, according to Mallick et al. [6]. This assumption is researched in the current study by the exploration of the G_{train} and G_{test} similarity impact on the performance of the model. The results for the experiments are shown in Section 5.

The correlation between the similarity of two graphs and

the performance of transferred model can majorly improve future model transferring in traffic forecasting. The similarity between two graphs is measured using graph distance metrics $D(G_{train}, G_{test})$, where D is one of the distance measures between weighted directed graphs. The performance is measured in prediction error $E(G_{train}, G_{test})$ of the model transferred from the region G_{train} to the region G_{test}. Since GNN models operate directly on the G_{train} and are designed to predict the traffic on $G_{test} = G_{train}$, the hypothesis for the general correlation, explored in this study, is defined in Equation 1.

$$E(G_{\text{train}}, G_{\text{test}_1}) > E(G_{\text{train}}, G_{\text{test}_2}) \iff D(G_{\text{train}}, G_{\text{test}_1}) > D(G_{\text{train}}, G_{\text{test}_2}) \tag{1}$$

The current study's formal hypothesis can be described as follows: the prediction error increases as the graph distance between the training and test regions increases, and conversely, the prediction error decreases as the graph distance between these regions decreases.

3 Methodology

This section gives an overview of methods defined to do the experimental work. It describes the used datasets, defines metrics used to evaluate the results, explains the choice of the GNN model, and shows the graph selection techniques.

3.1 Data preparation and dataset description

Two real-world traffic speed datasets were used in the experiments.

Dataset for model training

The model was trained on part of the METR-LA dataset. The METR-LA dataset consists of 207 average speed sensors on the highways in Los-Angeles County, USA. It contains data for the period from March to June 2012. There are 34272 timesteps with a 5-minute difference between each timestep.

Two subsets of the initial dataset were used for training. The first subset consists of 50 sensors on the three highly used highways in the region. (see Figure 1) This sensor set represents small-scale traffic patterns, such as road intersections, and bigger-scale regional patterns, such as a circle formed by multiple roads. The second subset comprises 10 sensors on two highway intersections (see Figure 2). This sensor set only represents small-scale traffic patterns. This sensor selection helped significantly decrease model training times compared to the full METR-LA dataset training. It also made it easier to find similar regions suitable for model transferring, by increasing the number of potential testing regions in the dataset for transferring.

Dataset for transferring

The PEMS-BAY dataset was used to find the regions for transferring. The PEMS-BAY dataset consists of 325 average speed sensors in the San Francisco Bay Area, USA. The PEMS-BAY contains traffic data from January to June 2017. The dataset consists of 52116 timesteps with a 5-minute rate between them. The overall structure of the dataset is similar to the METR-LA dataset, making it suitable for the model's transferability.

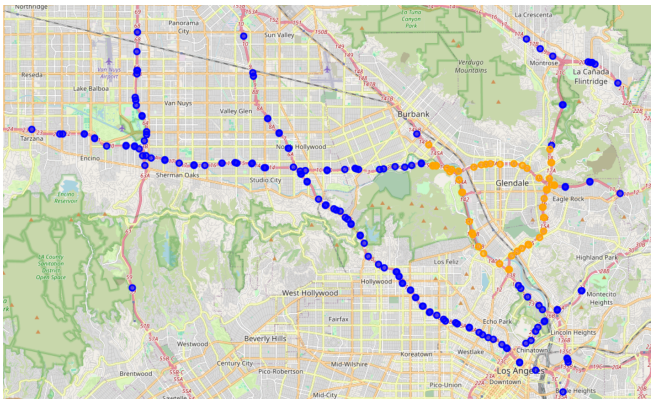


Figure 1: Map of Los Angeles: Orange markers are 50 training sensors; blue markers are the full METR-LA dataset

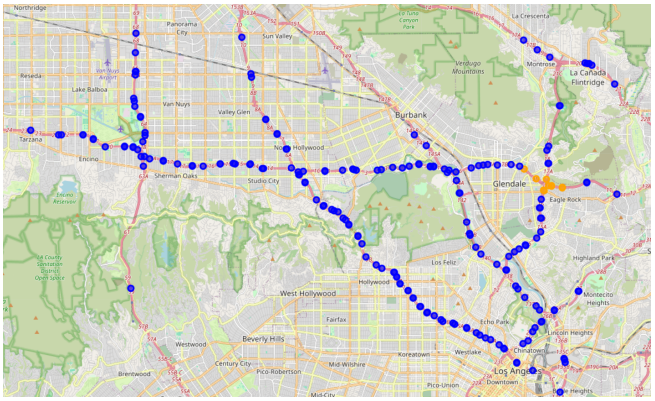


Figure 2: Map of Los Angeles: Orange markers are 10 training sensors; blue markers are the full METR-LA dataset

Data preprocessing

Both datasets were divided into 3 parts: the first 70% of the dataset was used for the model's training, the next 10% was used for validation during the training process, and the last 20% was used to test model performance. The data division can be observed in Table 1. The data division is based on the paper of Mallick et al. [6] and Li et al. [5] and makes it possible to compare the model's performance with the literature.

Only the testing part of the PEMS-BAY dataset was used in the later experiments for model transfer. Using the full dataset for transferability testing could potentially increase the accuracy, however, it would significantly increase the testing times. This paper decides to stay with the initial data setup described in the paper of Mallick et al. [6] and use only the subset of the PEMS-BAY dataset to decrease the computational complexity of the experiments.

Datasets contain missing values for multiple sensors (e.g. zero or NaN values). However, removing this data from the datasets is impossible due to the disruption of data continuity. This issue is fixed by using the performance metrics with missing value masking. The performance metrics are described more in Section 3.2

Distance dataset

Two datasets with road distances between METR-LA and PEMS-BAY sensors are used for adjacency matrix creation.

Dataset	Timestep	Period
METR-LA	34272	2012.03.01 - 2012.06.27
METR-LA train	23990	2012.03.01 - 2012.05.23
METR-LA validate	3428	2012.05.23 - 2012.06.04
METR-LA test	6854	2012.06.04 - 2012.06.27
PEMS-BAY	52116	2017.01.01 - 2017.06.30
PEMS-BAY test	10423	2017.05.25 - 2017.06.30

Table 1: Datasets used for model training, testing, and tranfering

The datasets represent a road distance from one sensor to another. The distance datasets contain the driving distances from each sensor to the sensors with a driving distance smaller than approximately 20000 feet (6096 meters), creating a non-complete graph with the region structure. Farther distances are represented with infinity value in the adjacency matrix.

3.2 Evaluation metrics

Performance metrics

Mean Average Error (MAE) is used as the main performance metric in the research. Additionally, Root Mean Squared Error (RMSE) is added to compare the performance of the models. MAE and RMSE are calculated as shown in Equations 2 and 3. Both metrics were adjusted to mask the missing values in the data. The missing values are not considered in the performance measurements, by excluding indexes of zero, negative, or not-a-number values from the Ω set.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|\Omega|}\sum_{i\in\Omega}|y_i - \hat{y}_i| \tag{2}$$

$$\text{RMSE} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{|\Omega|}\sum_{i\in\Omega}(y_i - \hat{y}_i)^2} \tag{3}$$

where:
n : *amount of sensors*
Ω : *indices of observed samples*
y : *ground truth values*
$\hat{y}$: *predicted values*

Masked MAE and RMSE metrics are used in many traffic forecasting papers [3, 5] and can be used to evaluate performance compared to related research. Lower values of MAE or RMSE indicate a better model prediction performance.

Graph distance metrics

Graph distance metrics, introduced here, are used to measure the similarity between two road traffic regions in this research. Graph distance metrics represent the difference in the structure of two graphs. Lower distance values correspond to the bigger similarity between the two graphs.

The distance between the weighted directed graphs can be expressed as the matrix norm on the difference between this graph adjacency matrices [13].

Multiple matrix metrics are used in this research to represent the distance between graphs. The primary metric used is the Frobenius norm (see Equation 4). As described by Golub and Van Loan [14] the Frobenius norm is one of the most popular matrix norms. This norm represents the Euclidean distance between two matrices. The research also uses the sum of absolute values (see Equation 5) in the matrix. This metric represents the Manhattan distance for matrices and is less impacted by the outlier values in the matrix than the Frobenius norm.

$$\text{FroD}(A,B) = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij} - b_{ij}|^2} \tag{4}$$

$$\text{AbsSum}(A,B) = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij} - b_{ij}| \tag{5}$$

where:
A, B : *compared matrices*
m, n : *amount of rows and columns in the matrices*
a_{ij}, b_{ij} : *values of a and b in the row i and column j*

The last metric used in the research is the cosine distance between column and row vectors in two matrices. Cosine distance is the distance representation of cosine similarity. It is often used to find the degree of similarity between objects, represented as vectors [15]. Cosine distance measures an angle between two vectors, where 0 is a similar vector, 1 is an orthogonal vector and 2 is an opposite vector. Cosine distance can take values only from 0 to 1 when operated on non-negative vectors.

The research uses the average of column-wise and row-wise cosine distances between two matrices (see Equation 6), comparing the similarity between incoming and outgoing distance vectors for each pair of sensors in two matrices.

$$\begin{aligned}\text{CosD}(A,B) = &\frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n}1 - \frac{\mathbf{A}_{i\cdot}\cdot\mathbf{B}_{i\cdot}}{\|\mathbf{A}_{i\cdot}\|\|\mathbf{B}_{i\cdot}\|}\right.\\ &\left.+\frac{1}{m}\sum_{j=1}^{m}1 - \frac{\mathbf{A}_{\cdot j}\cdot\mathbf{B}_{\cdot j}}{\|\mathbf{A}_{\cdot j}\|\|\mathbf{B}_{\cdot j}\|}\right)\end{aligned} \tag{6}$$

where:
$\mathbf{A}_{i\cdot}, \mathbf{B}_{i\cdot}$: *the row vectors from matrices A and B*
$\mathbf{A}_{\cdot j}, \mathbf{B}_{\cdot j}$: *the column vectors from matrices A and B*

Correlation metric

Pearson correlation coefficient is used to measure the observed correlation between graph distance and the model's performance in the research. Accordingly to Segwick [16], the Pearson coefficient is a good representation of linear correlations between two variables. The Pearson coefficient can take values from -1 to 1, where -1 shows a strong negative correlation, 0 means a lack of correlation, and 1 shows a strong positive correlation.

3.3 Model selection

A wide variety of models can be used for traffic forecasting problems. The survey by Jiang and Luo [3] describes Diffusion Convolutional Recurrent Neural Network (DCRNN) [5]. However, there are multiple innovative traffic forecasting models, that were not included in a survey. Here is the comparison of DCRNN model with Decoupled Dynamic Spatial-Temporal Graph Neural Network (D2STGN) [11] and Spatio-temporal graph mixformer (STGM) [12] models introduced

in Section 2 in terms of the current research questions.

According to Li et al. [5] DCRNN model is introduced specifically to solve a traffic forecasting problem. The model consists of diffusion convolutional layers that learn the patterns in data based on spatial dependencies. Additionally, recurrent neural network architecture is used to capture the temporal dependencies of data. The combination of both architectures makes it possible to learn complex dependencies of traffic data and show reasonable performance.

D2STGN [11] architecture is based on the DCRNN model. D2STGN uses a dynamic graph learning algorithm, that models dynamic relationships in traffic spatial data, based on historical information. This feature helps to improve the model's performance compared to DCRNN but requires historical data of the specific region for it.

STGM model [12] uses the convolutional layers to capture spatial dependencies, together with a gated mechanism and mixer layer to integrate information effectively. The model uses a similarity estimator trained on historical data to approximate node contributions. This architecture leads to increased performance and lower memory usage.

This paper opts to use the DCRNN model for experiments on model transferability. Firstly, the DCRNN model applies weights directly to a static sensor distance matrix, that can be obtained from regions without the historical traffic data. The other models use a dynamic approach to create the adjacency matrix, which can be difficult in transferability research. Secondly, the DCRNN model was successfully used in the other transfer learning research [6]. Due to this reason, the performance results of this research can be compared with those obtained in the literature. Lastly, DCRNN is a widely recognized and well-documented model, ensuring that the methodology and implementation are well-understood and can be reliably replicated.

The biggest limitation of DCRNN usage is the impossibility of testing on the different size regions. The testing region of DCRNN should consist of the same amount of sensors as the training region of this model. The data zero padding was introduced to address this issue in the paper by Mallick et al. [6]. However, it will not be used in this research paper due to the need to explore the similarity between training and transfer regions.

3.4 Graph selection

Random graph selection

Random sensor set generation is one of the easiest approaches for selecting traffic network regions. This research uses a pseudo-random generator function to choose the fixed-size random subgraph from the original graph.

Random graph selection is a performance-efficient and easily implementable approach. It helps to create a large amount of test regions, so the average performance of the model can be tested effectively. However, this approach mostly generates sparse datasets with large distances within the sensors. It can not create graphs with some specific required structure.

Simulated annealing selection

Another approach for graph selection is based on the search for graphs with specific parameters. The current research scenario involves searching the graphs with various graph distances to the model training graph for deeper research of the correlation between transfer graph distance to the training graph and the transferred model performance. However, the large search space makes it impossible to use classical search algorithms to select graphs with a maximally wide distance variation.

The current study uses a simulated annealing algorithm to search for structurally different graph generation. Simulated annealing (SA) [17] is the optimization algorithm introduced to find the global optima of the cost function. It iteratively accepts or rejects neighbor solutions with a certain probability. The probability is based on the difference in previous and next solution costs.

The distance between the target and searched graphs is used as the explored cost function in the current study scenario. The graphs with the difference in one sensor are assumed to be the neighbor graphs. This makes searching for the graphs with the possibly best and worst distances possible using the SA algorithm.

This research introduces the Bucketed Simulated Annealing (BSA) approach to select the graphs with a wide variation of distances to a target graph. The SA algorithm is limited to searching only for minimum values of the function, because of that the modified version of SA is introduced in this study. The BSA algorithm divides the range of cost function values into smaller, similarly sized ranges (buckets) and uses the SA approach to search for the graph in each bucket. For this scenario, the cost function for simulated annealing is adjusted to equal the difference between the distance and the closest value in the required range (see Equation 7).

$$C(G_{test}) = \begin{cases} |D(G_{train}, G_{test}) - r_{min}|, & \text{if } D < r_{min} \\ |D(G_{train}, G_{test}) - r_{max}|, & \text{if } D > r_{max} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where:

$D(G_{train}, G_{test})$: *graph distance between target graph and test graph*
r_{min} : *minimum value in the bucket*
r_{max} : *maximum value in the bucket*
C : *cost of the solution*

4 Model training and transfer

This section describes the setup of experiments to train and transfer the GNN model. It also shows the performance results of the model in the training region and transferred into other areas.

4.1 Model training

This research uses PyTorch 2.3.0[1] implementation of DCRNN model[2]. Models were trained on two datasets discussed in Section 3. Delft University of Technology supercomputer was used to train the models. It uses Intel XEON E5-6448Y 32C 2.1GHz CPU and NVIDIA Tesla A100 80GB GPU for computations. The models were trained with the parameter specification described in the paper by Li et al. [5],

[1] Available at: https://pytorch.org/

[2] Available at: https://github.com/chnsh/DCRNN_PyTorch

Dataset	MAE	RMSE
Full dataset	3.60	7.59
50-sensor subset	**2.92**	**6.43**
10-sensor subset	**4.75**	**14.56**

Table 2: Model performance for 1 hour predictions on sensors with historical data in METR-LA area

Model	MAE	RMSE
STGCN	6.53	10.07
FC-LSTM	4.69	8.48
GMAN	4.05	7.57
DCRNN	3.3	6.91
50-sensor subset DCRNN	**4.74±0.02**	**9.96 ±0.03**
10-sensor subset DCRNN	**3.75±0.04**	**7.78±0.07**

Table 3: Transfered model performance on sensor sets in PEMS-BAY. The first 4 rows represent models trained on full dataset. The last 2 rows represent the average performance of the model tested on subsets of the dataset with the standard error

which proved to be optimal for this model for traffic forecasting.

The model performance was tested on the METR-LA testing dataset and performance was compared to the results shown in the paper by Li et al. [5]. Table 2 shows the model performance on the full dataset obtained from the paper and the model performance on subset graphs obtained during the experimental work. The model performs best on the 50-sensor subset. This can be due to the strong dependencies between the sensors and good spatial representation of the road network. The model's performance on the smaller region is much lower, due to the lack of regional information from neighbor nodes.

4.2 Model transfer

The model's performance was tested on random sensor sets from the PEMS-BAY testing dataset to check the model's general transferability. The sensor sets were chosen to have the same amount of sensors as the training region of the model. due to the limitation of DCRNN described in Section 3.3. 200 random graphs were taken from the San Francisco Bay Area for each scenario and model performance was tested on the graphs.

The observed results were compared with the information provided by Mallick et al. [6]. The study by Mallick et al. shows the results for the models trained on the full LA dataset and transferred to the full PEMS-BAY dataset. The comparison with current research models can be observed in Table 3. The error of this research model is given as the average error for 200 random graphs. Both subset models underperform compared to the DCRNN model trained on the full METR-LA dataset and tested on the full PEMS-BAY dataset, yet they still outperform several other models. The difference in performance can happen because DCRNN learned more global traffic patterns from the bigger dataset.

The model trained on a smaller subset outperforms the model trained on a larger sensor set in the transfer regions. This can occur because the 50-sensor subset is strongly correlated, causing the model to learn specific regional data patterns that do not generalize well to the other regions and harm the predictions. Conversely, the smaller subset model learned primarily individual sensor behavior without relying on a particular graph structure, which helps the model to perform better in transfer problem scenarios.

As an example, Figure 3 compares two model predictions for one specific sensor. The 10-sensor model can recognize some major traffic changes. However, the 50-sensor model shows large prediction fluctuations and recognizes incorrect traffic patterns. This pattern repeats for most other timeframes and sensors, providing insights into overall model performance values.

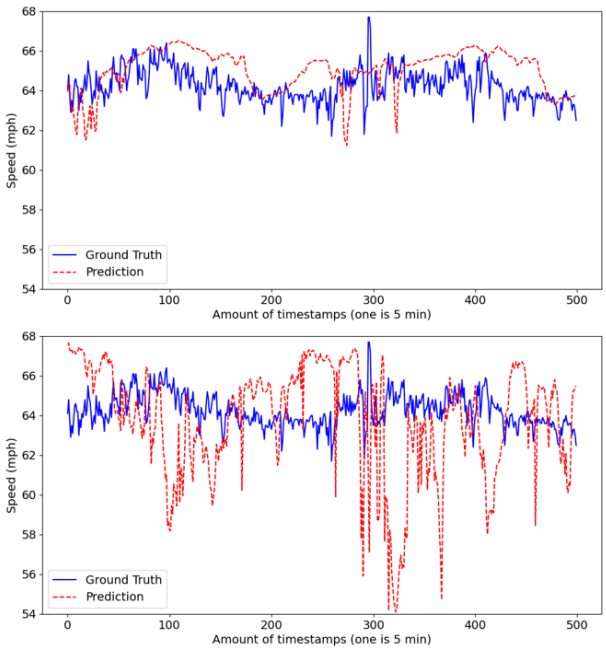


Figure 3: Comparison of DCRNN predictions using 10 sensors (top) and 50 sensors (bottom) for a specific sensor, showing ground truth (blue) versus predictions (red)
(Dataset: PEMS-BAY; Sensor: 404586; Timeframe: 4500-5000)

5 Correlation of Sensor Structure and Model Performance

The correlation between the transfer region structure and the model's performance is explored here. Multiple metrics described in Section 3 are used here as the metrics for the distance between the model's training region and transfer region and will be later mentioned as graph distance.

The DCRNN model is trained on the normalized version of the graph adjacency matrix, where far-away distances are set to 0 and closer distances result in higher values up to 1. Such matrix is created using Gaussian kernel and is further described in the paper by Li et al. [5]. However, this matrix normalization does not fully represent the graph's adjacency matrix. In this research, the non-normalized adjacency matrices of the graphs are used to compute the graph distance as well. The adjacency matrices of non-complete graphs have infinite values representing non-connected vertices. The masks of 0 - as the minimum value in distance datasets, 20000 - as the approximate maximum value of dis-

Metric	Mask	Correlation ± SE
CosD	40000	0.129 ± 0.061
CosD	20000	0.102 ± 0.055
AbsSum	0	-0.028 ± 0.030
FroD	0	-0.037 ± 0.026
AbsSum	normalized	-0.051 ± 0.057
CosD	normalized	-0.078 ± 0.030
FroD	normalized	-0.089 ± 0.046
AbsSum	40000	-0.121 ± 0.100
FroD	40000	-0.128 ± 0.104
AbsSum	20000	-0.128 ± 0.106
FroD	20000	-0.142 ± 0.112
CosD	0	-0.192 ± 0.100

Table 4: Correlation between graph distances and model performance for randomly selected transfer graphs (average between two models)

tance datasets, and 40000 - as the twice bigger value than the maximum value, are used for infinite distance values. This makes it possible to use graph distance metrics, defined in Section 3.2, in the later graph comparison.

The choice of mask value influences the calculated distance metrics. Smaller mask values (like 0) might lead to an underestimation of distances, while larger mask values (like 40000) increase the impact of missing values on the overall distance values. This research tests 3 possibilities of mask values. However, the effect of the chosen mask on the graph distance and model performance should be further examined in future work.

5.1 Correlation on random graphs

The correlation values between the model's performance and graph distance metrics were checked on the 200 randomly selected graphs for each model. Table 4 show the average correlation for each of the metrics between 2 models. Full results can be observed in Appendix A.

Most correlation values are close to 0, meaning no strong correlation is observed. The possible reason is that most testing graphs are not similar to the training graph and show a large distance metric. The model should be tested on the graphs with different ranges of graph distance to investigate the correlation.

It can be observed, that most of the correlations are negative, meaning that this metric recognizes incorrect graph similarities. The model performs better on the graphs that are less similar to the training region by such metrics.

5.2 Correlation using simulated annealing

This research takes 4 metrics with the most promising correlation, based on Table 4, to further investigate their correlation with performance.

The BSA approach is used for each metric to find 50 graphs with diverse distances. Firstly, the approximate maximum and minimum values of the distance between all the graphs are found using the standard simulated annealing approach. After, the distance range is divided into 50 buckets and BSA is used to find a graph within each of the buckets. This results in 50 graphs with a wide difference in graph distance to the training graph.

The simulated annealing parameters were adjusted for each metric using a manual trial-and-error approach. Parameter optimization has small importance in our scenario because bucketed simulated annealing will perform a local search in the worst case. This gives a reasonable algorithm performance for our scenario even with a local search approach. However, all the parameters are given in Appendix B for the result reproducibility.

Metric	Mask	50-sensor model	10-sensor model
CosD	40000	0.4	0.35
CosD	20000	0.23	0.11
AbsSum	0	-0.37	-0.24
FroD	0	-0.55	-0.32

Table 5: Correlation between graph distances and performance of the 50-sensor and 10-sensor models for the transfer graphs selected using BSA

The results for 4 different metrics are shown in this research. The cosine distance with 20000 and 40000 masks were researched, as the only metrics with a positive correlation on random graphs. Frobenius distance and Absolute Sum with 0 mask were explored, as the metrics with close to 0 correlation.

Table 5 shows the correlation values for the researched metrics. All the obtained results can be explored visually in Appendix C.

Cosine distance

The results, observed in Table 5, show that the cosine distance metric performs the best compared to other metrics, especially with a 40000 mask. The cosine distance is the only metric showing a positive correlation. This suggests, that the model's performance improves as the graph distance between the training and transfer regions, measured by cosine distance, decreases.

However, the obtained correlation values are weak to moderate based on Turney [18]. The strong dependency also can not be observed based on data plots shown in Figure 4 and Figure 5. There is also a big variation in the performance of the graphs with a small graph distance to the training region.

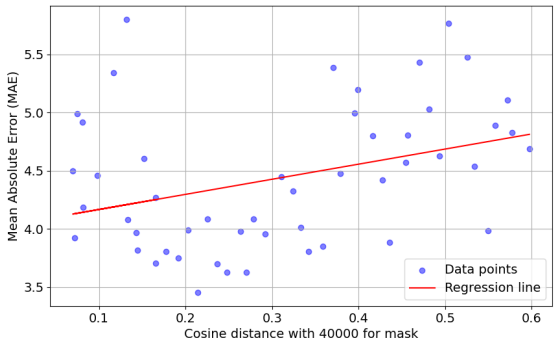


Figure 4: Correlation between graph distance(measured in CosD with mask 40000) and performance of the 10-sensor model

Other metrics

The analysis of other metrics, including Frobenius distance and the Absolute Sum distance, showed negative correlations with model performance. The full results can be observed in

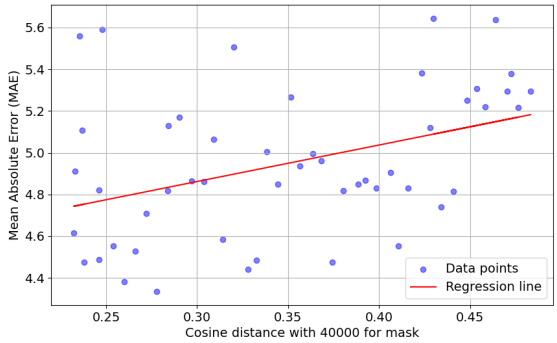


Figure 5: Correlation between graph distance(measured in CosD with mask 40000) and performance of the 50-sensor model

Appendix C. This indicates that the metrics are not suitable for the transferability measure in traffic forecasting. The negative correlations imply that as the graph distance increases, indicating a greater structural difference between the training and transfer regions, the model's performance improves, which disapproves the proposed hypothesis.

The analysis of the plots reveals an absence of a strong correlation between the distance to the training region and model performance. The plots show high variability in model performance across the graphs with similar distances, suggesting that regional differences do not consistently impact the model's effectiveness.

6 Responsible Research

In conducting this research, several ethical considerations and measures were taken to ensure research trustworthiness. This section describes the concerns ensuring that this study's contributions to model transferability in traffic forecasting are reliable, valuable, and ethical.

6.1 Data privacy

Data privacy is an important aspect of the research. The research adheres to data privacy standards by using anonymized and aggregated traffic data, thus respecting individuals' privacy and adhering to ethical guidelines for data usage. The primary datasets, METR-LA and PEMS-BAY, consist of publicly available traffic data collected from sensors on public roads, ensuring no personally identifiable information is involved.

6.2 Research reproducibility

Reproducibility is a key aspect of responsible research, and this study has taken several steps to ensure that the methods and results can be reliably reproduced by others. Detailed descriptions of the datasets, data preprocessing steps, and experimental setup are provided in Section 3 and Section 4, allowing other researchers to replicate the study. The use of the publicly accessible and well-documented model, such as the DCRNN model implementation in PyTorch [3], further supports reproducibility. All code used for data processing, model training, result collection, and evaluation is made available through a public GitHub repository[4], promoting transparency and enabling result verification.

6.3 Research integrity

Integrity in the research is essential for ensuring the trust and reliability of findings. The study adheres to honesty, transparency, independence, and responsibility, as outlined in the Netherlands Code of Conduct for Research Integrity [19]. The findings are reported truthfully, throughout the study, using rigorous and justified methods, and disclosing potential conflicts of interest. Upholding these standards helps to prevent harm and promotes a culture of mutual trust among researchers and the public.

7 Discussion

This section discusses the results of the presented experiments and points out the limitations of the research.

7.1 Results of the study

The performed experiments explore the transferability of GNN for traffic forecasting. The study shows the performance of DCRNN model in transferability tasks and is tested on multiple structurally different regions. Two DCRNN models were trained on size and structurally different sensor sets to make the research less specific on the training set.

The models were trained on the different subsets of the METR-LA dataset and the performance was tested to answer the first subquestion of the research. Accordingly to Section 4.1 it can be concluded, that the model performance depends on the chosen sensor set and performs the best on the dataset with the strong spatial dependencies when tested in the same region.

The two trained models were transferred to the random graphs from another dataset in the Section 4.2. It can be observed that the smallest model (10-sensor) performs the best in transferability, despite showing the worst performance in the direct learning task. It can be observed, that the model trained on the less spatially correlated region performs the best in transferability tasks because it avoids overfitting to specific spatial patterns of the training region.

The next experiment explores the correlation between the transfer region distance to the training region and the model's performance in those regions. It shows the regional spatial structure metrics that can be correlated with the model performance on that region data. It can be observed that most of the metrics are not useful in terms of transferability and show a negative correlation with the model performance.

The cosine distance is the only metric that showed a positive correlation. However, the observed weak to moderate correlation and the diversity in model performance for the regions with similar metric values show a small dependency of the model's performance on that metric. It is suggested that the usage of this metric for GNN transferability in traffic forecasting should be explored more in later deeper research.

As it can be observed, the correlation varies based on the selected metric. Other graph distance metrics can be researched in the future, for the possible findings of the better-performing metrics. The variety of different matrix distance

[3] Available at: https://github.com/chnsh/DCRNN_PyTorch

[4] Available at: https://github.com/ikrvc/dcrnn_transferability

metrics [20] can be explored in further research.

7.2 Limitations

The current study was limited in exploring the transferability due to the short time frame of the research and limited access to computing power. The research was performed in 10 weeks, limiting the possibility for extensive experimental work and literature research. The long access queues to the computation power on DelftBlue supercomputer and the high computational complexity of DCRNN model limited the amount of practically performed experiments and the size of the chosen datasets.

Due to the limitations discussed here, the experiments were performed only on two trained models and smaller datasets. A larger number of models could give a greater confidence level and generalizability of the results and derived conclusions. Not all the metrics, shown in the Appendix C, were explored using the simulated annealing approach due to the shortage of time and complexity of computations. However, the four most promising metric configurations were analyzed and described during the research. A deeper exploration of metrics and mask values could help to give more confidence in the conclusions.

8 Conclusions and Future Work

This paper explores the transferability of GNN in traffic prediction tasks. DCRNN model is used for the paper as one of the most suitable and researched GNN models for transferability. The study examines the model's performance within the training region and evaluates its effectiveness when applied to a different regional context. Finally, the paper presents the correlation of the model performance and the distance between the train and transferred traffic regions.

During the model training step, it can be noticed, that the model performance varies based on the selected training region. The model performs best in the region with a strong dependency between sensors, which supports the general idea of GNN's ability to capture regional spatial patterns.

However, in transfer scenarios, the model with a smaller correlation between sensors shows better performance. It can be concluded that the model is learning spatial patterns in the region, which can potentially be harmful for transferred model predictions. This supports finding the graph metrics that will reveal useful spatial patterns and be valuable in transfer domain adaptation.

Diverse graph distance metrics were explored to identify the potential of their usage for the model transferability. This study concludes that most metrics, such as Frobenius distance and Absolute Sum distance, can not identify the required patterns and show a negative correlation with the model performance. Onky the cosine distance metric showed a positive correlation with the model performance, making it potentially useful for future transferability tasks in traffic forecasting. However, the observed correlation was weak to moderate, so it can not be presented as a reliable metric.

The observed correlation between graph distance and the model's performance disproves the proposed hypothesis of positive correlation. The metrics mostly capture incorrect regional spatial patterns or are unrelated to performance. It shows that the approach of using the more generally trained model, as described in the paper by Mallick et al. [6], will be more effective in transferability tasks than the domain adaptation based on graph distance metrics. However, the proposed hypothesis can be researched more deeply by examining other possible metric configurations in future work.

The future work for the full exploration of model transferability will include (1) training and testing the model for different datasets and regional scenarios to research the generality of this research findings (2) exploration of all the proposed metrics configurations using the simulated annealing approach and (3) exploration of other possibilities for the mask values and graph distance metrics to make extensive research of a metric with a strong positive correlation possibly, (4) deeper exploration of cosine distance between graphs as the transferability measure.

References

[1] M. Papageorgiou, M. Ben-Akiva, J. Bottom, P. H. L. Bovy, S. P. Hoogendoorn, N. B. Hounsell, A. Kotsialos, and M. McDonald, "Chapter 11 ITS and Traffic Management," in *Handbooks in Operations Research and Management Science*, ser. Transportation, C. Barnhart and G. Laporte, Eds. Elsevier, Jan. 2007, vol. 14, pp. 715–774. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927050706140116

[2] K. L. Soon, R. K. C. Chan, J. M.-Y. Lim, and R. Parthiban, "Short-term traffic forecasting model: prevailing trends and guidelines," *Transportation Safety and Environment*, vol. 5, no. 3, p. tdac058, Jun. 2023. [Online]. Available: https://doi.org/10.1093/tse/tdac058

[3] W. Jiang and J. Luo, "Graph Neural Network for Traffic Forecasting: A Survey," *Expert Systems with Applications*, vol. 207, p. 117921, Nov. 2022, arXiv:2101.11174 [cs]. [Online]. Available: http://arxiv.org/abs/2101.11174

[4] J. Jiang, Y. Shu, J. Wang, and M. Long, "Transferability in Deep Learning: A Survey," Jan. 2022, arXiv:2201.05867 [cs]. [Online]. Available: http://arxiv.org/abs/2201.05867

[5] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting," Feb. 2018, arXiv:1707.01926 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1707.01926

[6] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane, "Transfer Learning with Graph Neural Networks for Short-Term Highway Traffic Forecasting," in *2020 25th International Conference on Pattern Recognition (ICPR)*. Milan, Italy: IEEE, Jan. 2021, pp. 10 367–10 374. [Online]. Available: https://ieeexplore.ieee.org/document/9413270/

[7] Y. Huang, X. Song, Y. Zhu, S. Zhang, and J. J. Q. Yu, "Traffic Prediction with Transfer Learning: A Mutual Information-based Approach," Mar. 2023, arXiv:2303.07184 [cs]. [Online]. Available: http://arxiv.org/abs/2303.07184

[8] H. Almukhalfi, A. Noor, and T. H. Noor, "Traffic management approaches using machine learning and deep learning techniques: A survey," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108147, Jul. 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197624003051

[9] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997.

[10] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, Oct. 1994. [Online]. Available: http://ieeexplore.ieee.org/document/329294/

[11] 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," Sep. 2022, arXiv:2206.09112 [cs]. [Online]. Available: http://arxiv.org/abs/2206.09112

[12] M. Lablack and Y. Shen, "Spatio-temporal graph mixformer for traffic forecasting," *Expert Systems with Applications*, vol. 228, p. 120281, Oct. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423007832

[13] T. Gervens and M. Grohe, "Graph Similarity Based on Matrix Norms," Jun. 2022, arXiv:2207.00090 [cs, math]. [Online]. Available: http://arxiv.org/abs/2207.00090

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins University Press, 1985.

[15] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine similarity to determine similarity measure: Study case in online essay assessment," in *2016 4th International Conference on Cyber and IT Service Management*, Apr. 2016, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/7577578

[16] P. Sedgwick, "Pearson's correlation coefficient," *BMJ*, vol. 345, pp. e4483–e4483, Jul. 2012.

[17] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, p. 671–680, May 1983.

[18] S. Turney, "Pearson Correlation Coefficient (r) | Guide & Examples," May 2022, (Accessed: 2024-06-10). [Online]. Available: https://www.scribbr.com/statistics/pearson-correlation-coefficient/

[19] "Netherlands Code of Conduct for Research Integrity | NWO," (Accessed: 2024-06-05). [Online]. Available: https://www.nwo.nl/en/netherlands-code-conduct-research-integrity

[20] "Matrix Distance," (Accessed: 2024-06-14). [Online]. Available: https://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/matrdist.htm

Appendices

A Random graph correlation

Metric	50-sensor model	10-sensor model	Average ± SE
Frobenius with 0 mask	-0.011	-0.063	-0.037 ± 0.026
Frobenius with 20000 mask	-0.030	-0.254	-0.142 ± 0.112
Frobenius with 40000 mask	-0.024	-0.232	-0.128 ± 0.104
Frobenius with normalized matrices	-0.134	-0.043	-0.089 ± 0.046
CosD with 0 mask	-0.092	-0.292	-0.192 ± 0.100
CosD with 20000 mask	0.047	0.156	0.102 ± 0.055
CosD with 40000 mask	0.068	0.190	0.129 ± 0.061
CosD with normalized matrices	-0.108	-0.048	-0.078 ± 0.030
AbsSum with 0 mask	0.002	-0.058	-0.028 ± 0.030
AbsSum with 20000 mask	-0.022	-0.234	-0.128 ± 0.106
AbsSum with 40000 mask	-0.021	-0.221	-0.121 ± 0.100
AbsSum with normalized matrices	-0.108	0.006	-0.051 ± 0.057

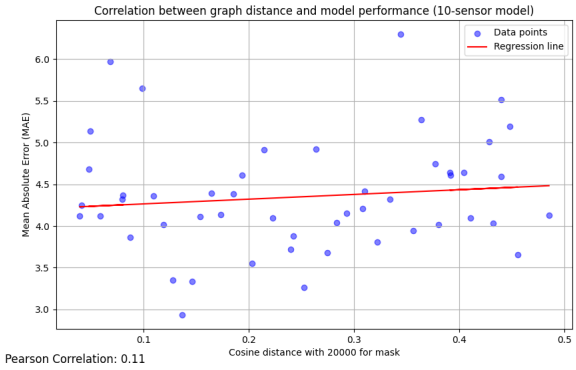
Table 6: Correlation between graph distances and performance of the 50-sensor and 10-sensor models for randomly selected transfer graphs

B Bucketed simulated annealing (BSA) parameters

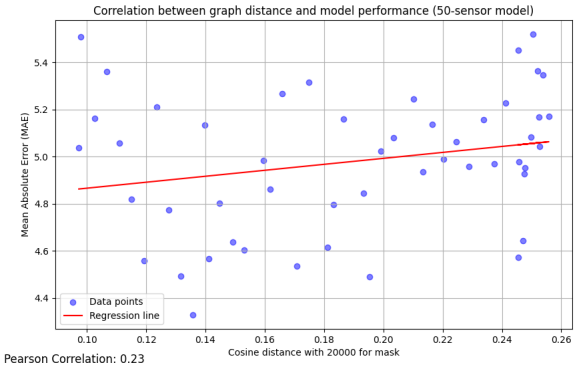
Metric	Graph size (number of sensors)	BSA Iterations	BSA Tempera-ture	BSA Cooling Rate	Minimum value	Maximum value
Frobenius with 0 mask	50	10000	10000	0.95	233343.36	346821.4
CosD with 20000 mask	50	10000	100	0.95	0.097311556	0.249699056
CosD with 40000 mask	50	10000	100	0.95	0.284320593	0.472720206
AbsSum with 0 mask	50	10000	10000	0.95	8778837	15218598
Frobenius with 0 mask	10	10000	10000	0.95	31184.63	77306.36
CosD with 20000 mask	10	10000	100	0.95	0.039547563	0.485200286
CosD with 40000 mask	10	10000	100	0.95	0.069766998	0.456905723
AbsSum with 0 mask	10	10000	10000	0.95	176105.9	655629.5

Table 7: Parameters for BSA graph search for each metric

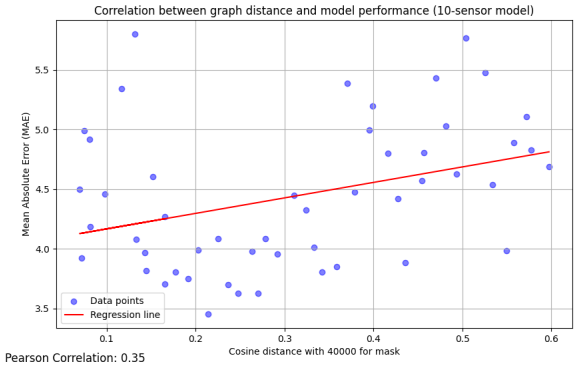
C Correlation graphs for explored graph distance metrics



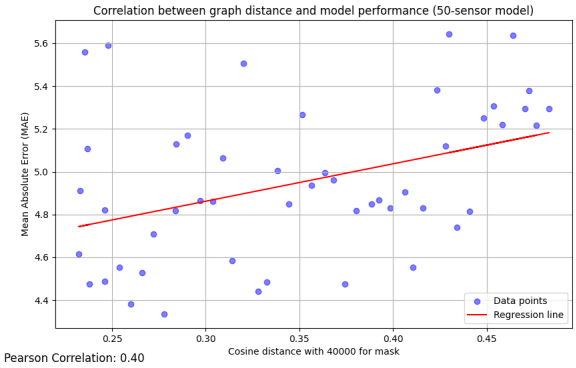
(a) CosD (20000 mask) with 10-sensor model



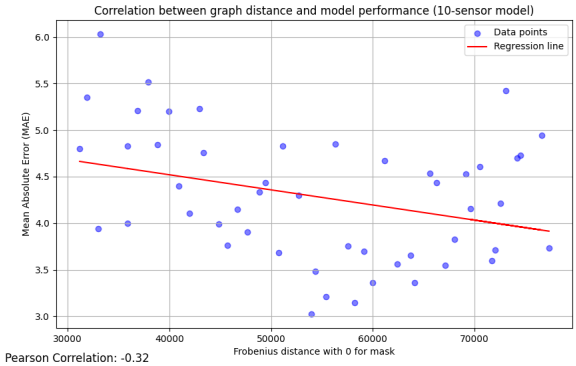
(b) CosD (20000 mask) with 50-sensor model



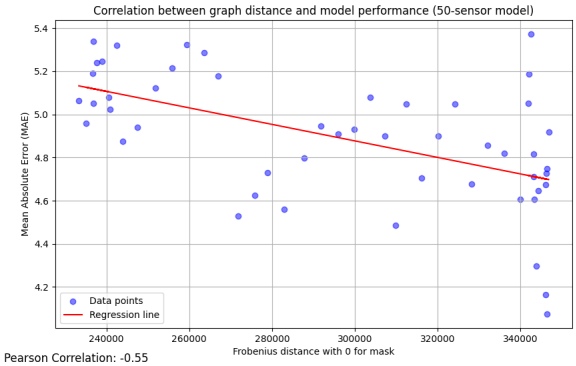
(c) CosD (40000 mask) with 10-sensor model



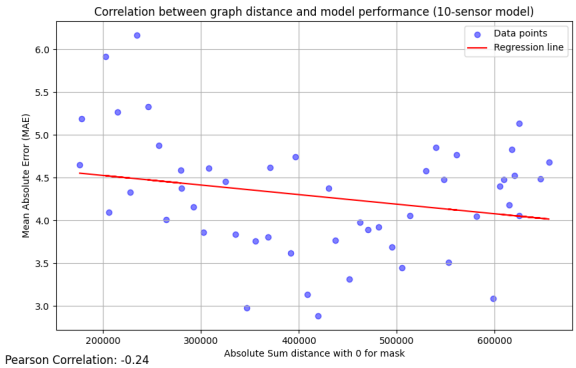
(d) CosD (40000 mask) with 50-sensor model



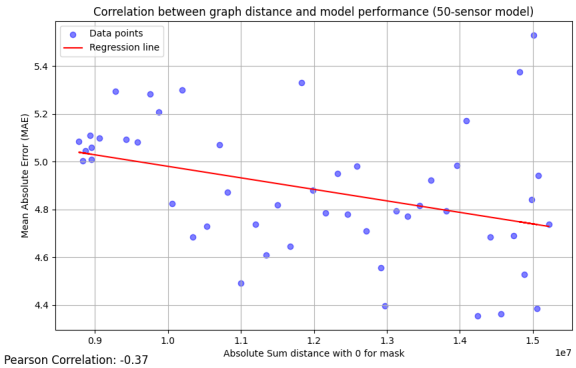
(e) FroD (0 mask) with 10-sensor model



(f) FroD (0 mask) with 50-sensor model



(g) AbsSum (0 mask) with 10-sensor model



(h) AbsSum (0 mask) with 50-sensor model

Figure 6: Correlation between graph distances and performance of the 10-sensor (left column) and 50-sensor (right column) models for the graphs selected using BSA for transferring