Regression Performance of Relational Fusion Networks on Urban Road Networks

Thales E. Cervi[1], Ricardo Lüders[1], Thiago H. Silva[1], Myriam R. Delgado[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro, 3165 – Rebouças – CEP 80230-901 – Curitiba – PR – Brasil

tcervi@alunos.utfpr.edu.br, {luders,thiagoh,myriam}@utfpr.edu.br

Abstract. *Urban transportation planning in densely populated areas is a problem in constant need of efficient solutions. Graphs can represent urban street networks and be used to train algorithms, enriching decisions with information learned from structural and topological data of cities. Relational Fusion Networks (RFNs) are Graph Neural Networks specifically tailored for learning on road networks. This work explores the use of RFNs in estimating free-flow travel times and includes experiments on relevant cities from all continents. Results demonstrate the significance of fusion functions and city characteristics in both the learning process of RFNs on regression tasks and the capacity to extrapolate acquired knowledge to different cities.*

1. Introduction

The relationship between the world population and its inhabited spaces is constantly changing. Since urban centers have become more attractive to large concentrations of people, several social and environmental issues are still waiting for sustainable solutions. The challenge of urban transportation in areas with increasing population density reinforces the ongoing demand for urban design efficient solutions [Albino et al. 2015].

Urban design is the technical process of shaping the physical features of cities to provide services to its inhabitants, generating equitable and sustainable environments [Karimi 2012]. When investigating urban design problems, it is possible to use networks (or graphs) to represent street maps [Boeing 2017] and take advantage of network science to study, for example, urban form [Strano et al. 2013] and urban transportation [Parthasarathi et al. 2013]. Networks' power lies in their generality and focus on relationships between objects rather than the properties of individual objects [Hamilton 2020].

Machine learning techniques have continuously advanced network science and are recently being extended to address problems posed on irregular domains. In these domains, data is typically represented by pairwise interactions along the network's segments. Urban street networks, or road networks graphs [Boeing 2017, Jepsen et al. 2018], structure human interactions and transportation inside cities and can be mathematically represented by a graph holding structural information along with any features that its components might have. Graph learning algorithms, like Graph Neural Networks (GNNs), can evaluate transportation system patterns, using information learned from road network structural and topological data to enrich urban design decisions.

Literature on graph representation learning ranges from node classification tasks, such as in social networks [Ying et al. 2018], to regressions on the urban context, such as local culture prediction [Silva and Silver 2024]. However, learning with urban road network graphs differs substantially from such tasks and might require specialized architectures and techniques [Jepsen et al. 2018].

Most of the literature on road network learning tasks explores traffic prediction in the temporal context [Yu et al. 2017, Zhao et al. 2020, Derrow-Pinion et al. 2021, Abhinav Nippani 2024]. Despite possible contributions to urban design challenges, only a few GNN models have been designed for static representations of road networks.

In the literature on static representations of road networks, the Graph Attention Isomorphism Network (GAIN) [Gharaee et al. 2021] is a model proposed to overcome general-purpose GNNs on the road type classification task. It uses topological neighborhoods instead of the direct one-hop neighborhood usually selected for aggregation on GNNs, sampling neighbors based on an unbiased random walk to extend node representation and improve the learning procedure. Although claimed to overcome other GNNs on the road type classification task, it requires more processing to generate topological neighborhoods. Therefore, an extended comparison of training costs is still needed. Moreover, it was tested for classification tasks, while this work focuses on regression tasks.

The HyperRoad (Hypergraph-Oriented Road network representation) [Zhang and Long 2023] captures at the same time the pairwise, high-order, and long-range relationships among the roads. It constructs *hyperedges* based on regions with similar semantic functions to capture the high-order relationships among road segments and creates a *hypergraph* from it. Then, a dual-channel aggregating mechanism propagates simultaneously, information through the simple and the *hypergraph*. Both channels are then fused via a gating mechanism as the road embedding. The model is tested on both road type classification and travel time regression tasks.

Different from the HyperRoad proposed by [Zhang and Long 2023], which performs a regression task with temporal data (travel time) related to trajectories, our work explores the use of RFNs in a static context. The experiments consider the regression task of estimating free-flow travel times, which depend on streets' static features like type, length, and maximum legal speed. Regression performance is measured for twelve cities representing all continents.

Relational Fusion Networks (RFNs), the main model considered in this work, are GNNs designed specifically for learning on road networks [Jepsen et al. 2020]. Although RFNs have been demonstrated to be adapted for learning tasks on road networks, there is currently no discussion on its capacity of extrapolating acquired knowledge.

The present work proposes to compare four RFNs variants for predicting free-flow travel times in twelve cities selected from all continents: Aalborg, Brisbane, Nara, Manhattan, Miami, Seattle, Damascus, Cape Town, Johannesburg, Curitiba, Niteroi, and Recife. The mean absolute regression error is computed in self-test (RFN is trained and tested with data of the same city) and extrapolation-test (RFN is trained with data of one city and tested in other city). Moreover, the influence of feature selection is also evaluated.

The main contributions of our work are twofold: i) a comparative analysis of RFN performance in estimating free-flow travel times across urban road networks on relevant cities from all continents; ii) an evaluation of the ability of RFNs to extrapolate the learned model to cities whose road networks were not included in the training.

2. Brackground

This section introduces GNNs and provides details of RFNs, including the fusion and aggregation functions explored in the present work.

2.1. Graph Neural Networks

A GNN is a learning model that generates node representation with a strong dependency on both, the input network structure and any additional information its components might have. It generates an updated version of the input network, which can be useful on prediction tasks where a prediction model could be trained based on its output [Hamilton 2020].

GNNs can be synthesized as a locally shared learnable function g, applied to every neighborhood, that maps the multi-set of features $\boldsymbol{M}(\mathcal{N}(u))$ to a latent space. Later, a learnable function f is applied over this latent space, producing an updated version of the input graph. This transformation alters the feature representations while preserving the graph's connectivity. As illustrated in Figure 1, in general, the defining feature for most of the GNN models is the use of a neural message-passing mechanism, in which vector messages are exchanged between nodes and updated using neural networks, making the learned representations aware of the graph connectivity [Battaglia et al. 2018].

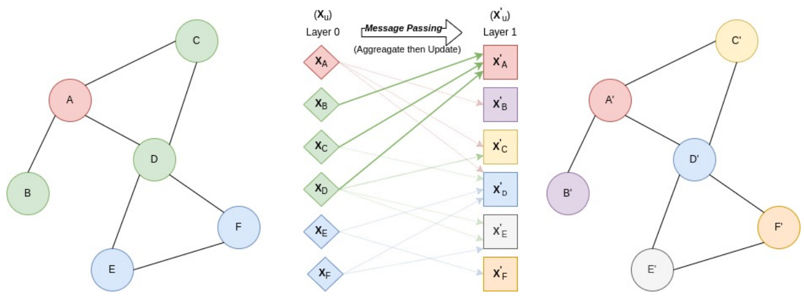


Figure 1. Message Passing: Updating Node A representation

In summary, GNNs need to learn, for every component (e.g., nodes), its local representations by aggregating neighborhood information in a way that it is neither affected by the neighborhood order (i.e., *permutation invariant*), nor by the order that components are presented (i.e., *permutation equivariant*) [Sanchez-Lengeling et al. 2021]. Using neighborhood features to generate information about a node's position and role in the graph is the key to generalizing knowledge learned to unseen nodes and graphs.

A road network graph has few edge and node feature information. It is usually a low-density network (i.e., few adjacent street segments), and the small size of neighborhoods makes aggregation on such graphs highly sensitive to abnormal neighbors. It also has between-edge features denoting the relation between road segments (e.g., turn direction, turn angle), characterizing them as edge-relational graphs [Jepsen et al. 2018].

General purpose GNNs are capable, in theory, of leveraging the structure of an urban graph, but are usually designed for node classification tasks on graphs that differ too much from road networks in terms of information available and structural characteristics (e.g., social networks, proteins). Also, GNNs are commonly based on the implicit assumption that graphs exhibit *homophily*[1], while road network graphs exhibit *volatile ho-*

[1]When neighbor nodes have similar features and changes in the graph occur gradually.

mophily[2] [Jepsen et al. 2018]. Therefore, some implicit assumptions of general-purpose GNNs do not hold for road network related tasks. Literature have shown that general-purpose GNNs fail to leverage road network graph structure and do not even outperform regular MLPs on such tasks [Jepsen et al. 2020].

When learning from road networks, edge features are crucial information to generate good descriptive representations since street segments are more interesting than simple street segment junctions (nodes). Therefore, one option for learning with urban graphs is to apply a Line Graph Transformation to the input data [Beineke and Bagga 2021]. A Primal Graph $\mathcal{G}^{\mathcal{P}} = (\mathcal{V}, \mathcal{E})$ is transformed by a Line Graph Transformation into a Dual Graph $\mathcal{G}^{\mathcal{D}} = (\mathcal{E}', \mathcal{B})$, in which every primal graph edge $(u, v) \in \mathcal{E}$ is transformed into a dual graph node $e \in \mathcal{E}'$. The between-edge connections in the primal graph (i.e., nodes) are transformed into dual graph edges $b \in \mathcal{B}$.

2.2. Relational Fusion Networks

A Relational Fusion Network (RFN) is a spatial, inductive, and general-purpose model proposed to overcome all the aforementioned specificities associated with road networks. It simultaneously learns representations based on primal (i.e., node relational) and dual (i.e., edge relational) representations. It aggregates over representations of relations, instead of representations of neighbors, and is designed to disregard relations with outlier neighbors using an attention mechanism.

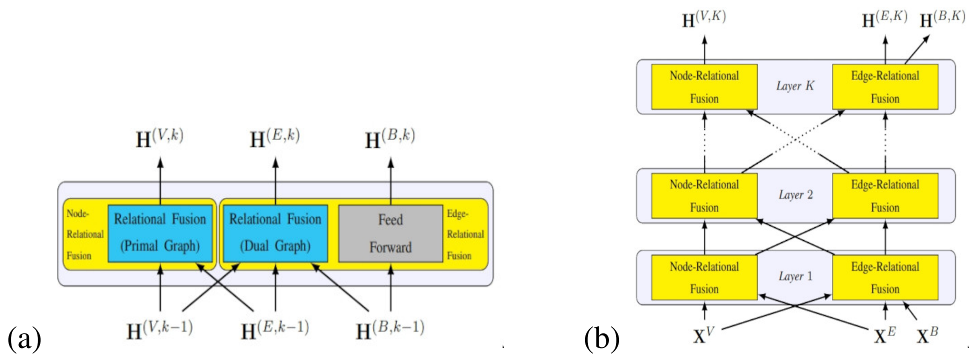


Figure 2. RFNs: (a) A RF layer and (b) a K-layered RFN [Jepsen et al. 2020]

An RFN layer receives an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of node features $X^{\mathcal{V}} \in \mathbb{R}^{|V| \times d^{\mathcal{V}}}$, a set of edge features $X^{\mathcal{E}} \in \mathbb{R}^{|E| \times d^{\mathcal{E}}}$ and a set of between-edge features $X^{\mathcal{B}} \in \mathbb{R}^{|B| \times d^{\mathcal{B}}}$, where $d^{\mathcal{V}}, d^{\mathcal{E}}$ and $d^{\mathcal{B}}$ represent, respectively, the node, the edge and the between-edge feature vectors dimensions. The input graph can be broken on its node-relational (i.e., primal), $\mathcal{G}^{\mathcal{P}} = (\mathcal{V}, \mathcal{E}), X^{\mathcal{V}}, X^{\mathcal{E}}$, and its edge-relational (i.e., dual), $\mathcal{G}^{\mathcal{D}} = (\mathcal{E}, \mathcal{B}), X^{\mathcal{E}}, X^{\mathcal{B}}$, representations.

Each RFN layer has three outputs: node, edge, and between-edge latent representations, that can be used for learning on any of the three components or even for multi-task learning settings. Each fusion layer of Figure 2 (a) performs both node-relational fusion and edge-relational fusion to learn representations from both views simultaneously. It captures their interdependencies from outputs of the previous layers, according to Figure 2 (b). The between-edge representations are transformed by a single feed-forward operator at each layer to increasingly learn representations.

[2] Regions can be highly *homophilic* and have sharp boundaries with abrupt changes

A relational fusion is a graph convolutional operator in which aggregation is performed over representations of relations (u, v) that a given node $u \in \mathcal{V}$ participates, as opposed to aggregation over-representation of u and its neighbors. A relation representation includes representations of nodes $u \in \mathcal{V}$, $v \in \mathcal{V}$ and the representation of the edge $(u, v) \in \mathcal{E}$, all fused by a Fusion Function ($FUSE^k$). A fusion function is responsible for extracting the right information from each relation, designed to capture Road Network Graphs intrinsic volatile *homophily*, allowing the creation of sharp boundaries at the edge of *homophilic* regions [Jepsen et al. 2020].

Fusion functions can be *ADDITIVE* or *INTERACTIONAL*. The first summarizes the relationship between u and v; however, it does not explicitly model interactions between representations, while the second better addresses the challenge of volatile *homophily* at the cost of an increase in parameters. Aggregation is then performed over the fused representations set, generating a single latent representation of u, called u'. RFNs can use an attention mechanism to filter out irrelevant and abnormal neighbors by weighting their contribution. Such filtering capacity is highly desirable for road network graphs since its low average node degree can amplify the aggregation noise.

3. Methodology

The present paper explores four RFN variants (see Table 1) on an edge regression task, using the same code made publicly available in [Jepsen et al. 2020].

Table 1. Relation Fusion Network Variants

Acronym	Fusion Function	Aggregation Function
RFN-AI	Interactional	Attentional
RFN-NAI	Interactional	Non-Attentional
RFN-AA	Additive	Attentional
RFN-NAA	Additive	Non-Attentional

This work addresses road networks of different cities, whose drivable maps are available in the collaborative mapping project OpenStreetMap® (OSM) [OpenStreetMap 2017]. OSM not only provides topological network structure, with direction and feature information included but provides it openly for anyone interested in verifying or reproducing experimental results. OSM data quality may vary between countries, but in general, high-quality street information is available [Boeing 2019].

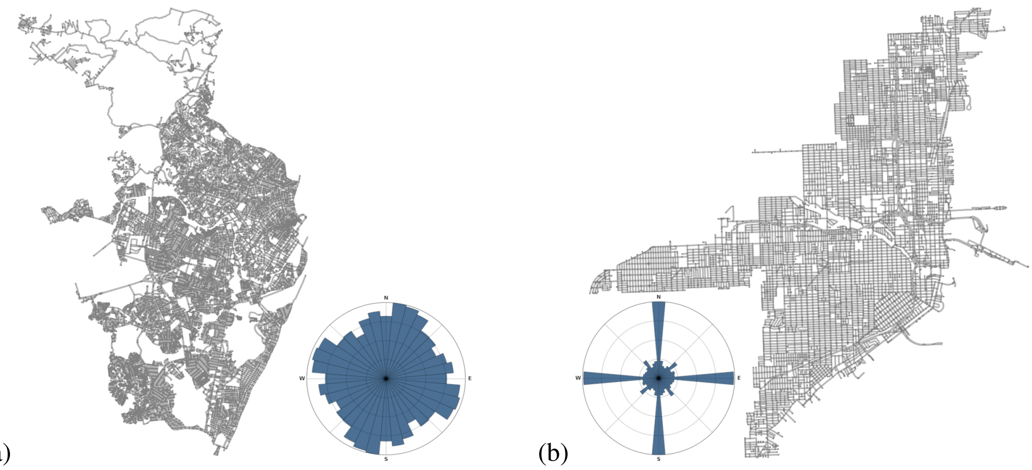
OSMnx Python package [Boeing 2017] has been chosen to retrieve data from OSM mainly because it addresses two issues when working with street networks: the data over-simplification and the repeatability of results. The retrieved information includes several useful graph global statistics on the selected road network graph, including the circuity average[3] (ς) and the orientation-order[4] (ϕ) [Boeing 2019]. OSMnx retrieves directed graphs with multi-edges containing the road network representation of a given city. All OSMnx graphs contain both node and edge features. The graph topology can be

[3]This metric indicates how much more circuitous a city is than it would be if all of its edges were straight-line paths between nodes. That is the relation between all edge lengths in the graph and the sum of all great-circle distances between all pairs for connected nodes [Boeing 2019].

[4]Orientation-order is defined by calculating the bearings of all edges and dividing it into 36 bins of 10^o each, measuring then the entropy of the binned bearings.

simplified by removing all non-intersection and non-dead-end nodes. Additionally, nodes outside the city boundary polygon are truncated, even if they have neighbors inside it.

Twelve cities have been selected for the experiments: the European city of Aalborg, the same city used in [Jepsen et al. 2020] experiments; Brisbane, an Australian city; Nara, in Japan; Manhattan island, Miami, and Seattle, all in the USA; Damascus, in Syria; Cape Town and Johannesburg, South African cities; and three Brazilian cities: Curitiba, Niteroi, and Recife. Figure 3 presents two road network examples and their respective polar plots[5] of edge orientation distribution. Large city road network graphs, including primal and dual graphs, and their associated feature sets, might not entirely fit in GPU memory. For those cities, networks were cropped to focus on their more central areas.



(a) (b)

Figure 3. Cities RNGs: (a) Recife, BR; (b) Miami, USA

Although OSM can have rich information on both the road network nodes and edges, the availability of features might vary from city to city, and even within the same city. Hence, this work establishes a standard setup for all the studied road networks. The proposed modeling only considers in **node features** *x* and *y* geographical coordinates (latitude and longitude) of an intersection. In the set of **edge features**, *lanes* and *length* represent, respectively how many lanes a street segment has and its length in meters; *one_way* indicates whether the street segment has one or two traffic ways; a vector of size five represents the one-hot-encoded *highway* classes of an edge. Finally, the **between-edges feature** set contains the *x* and *y* geographical coordinates of the edges intersection node; a one-hot-encoded vector for the *turn_direction* between the edges: {*"right", "left", "forward", "backward"*}; and *bearing*, representing the turn angle in degrees.

This work considers the following edges highway classes: {*residential, living_street, primary, primary_link, secondary, secondary_link, tertiary, tertiary_link, trunk, trunk_link, motorway, motorway_link, unclassified*}[6]. Nevertheless, it is naturally expected that an urban scenario would have few, or even none, *motorway* and *trunk* road segments. As they are classes representing high-speed roads connecting two or more different cities, they usually only outline or cross an urban road network. Moreover, within

[5]A polar plot divided into 36 sections of 10^o each, in which each section has a bar with height representing the count of edges that have that bearing orientation

[6]OpenStreetMap® documentation [OpenStreetMap 2017]

cities, major arterial roads usually connect different city's regions, and, therefore, edges classified as *primary* and *secondary* are expected but on a lower count when compared to *residential* or *living_street* edges. The same applies to *tertiary* and *unclassified* edges, although those represent minor public roads more often expected than arterial roads.

For instance, Figure 4 shows Aalborg *highway class* distribution. It is not a densely concentrated city within its territory. However, as depicted in Figure 4(a), it presents a considerable concentration of *residential* edges when compared to the other classes. Although unbalanced distributions are expected for urban networks, encoding 13 classes as one hot encoding might jeopardize the learning process. To reduce the number of edge features, this work reclassifies the original 13 classes into five class groups as Highest Performance Roads (HPR): motorway and trunk; Arterial Main Roads (AMR): primary and secondary; Minor Public Roads (MPR): unclassified and tertiary; Link Only Roads (LOR): all link classes; Local Traffic Roads (LTR): living_street and residential.

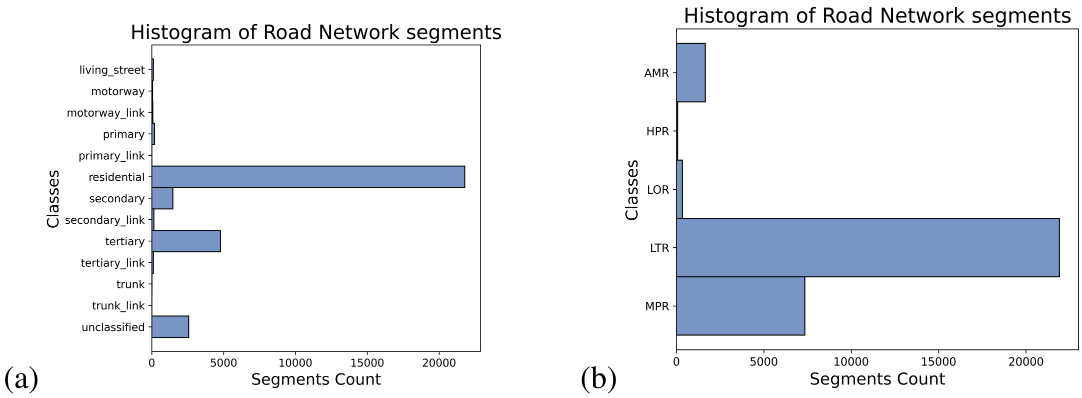


Figure 4. Edge classes in Aalborg road network: (a) original; (b) 5 classes.

Notice in Figure 4(b) that the classes' unbalanced nature is maintained. Balancing the class distributions within urban road networks is out of the scope of this paper. Instead, our experiments aim to evaluate RFN models' performance on unbalanced datasets, considering the potential significance of edge classification in modeling strategies.

In this paper, an urban road network is characterized by 12 global features: number of nodes, number of edges, average node degree, edge length mean and median (in meters), average streets per node, circuity average, entropy, free-flow travel-time mean and median (in seconds), maximum legal speed mean and median (in kilometers per hour). Moreover, a metric called Major Highway Class Rate is used to represent the edge class distribution of a road network. It measures how concentrated an edge class distribution is towards a specific class by computing the count ratio between the number of edges in the most common class and the total number of edges.

We consider two regression scenarios in the evaluation: i- *self-test* , which trains the model on a city A and tests it on unseen data (edges) for the same city, i.e., this scenario evaluates the generalization of regression on travel time with a fixed topology; and ii- *extrapolation test*, which trains the model on a city A and tests it on entirely new data from other cities, i.e., this scenario evaluates generalization of regression on travel time for a varying topology – this is useful to explore how well an RFN can extrapolate what was learned from an urban road network to a different urban road network.

4. Results

This work applies the same experimental setup to train and test all RFNs listed in Table 1, with experimental parameters detailed in Table 2. Notably, no normalization was applied in the output layers, and the final edge representation was intended to characterize the edge's free-flow travel time in seconds. This consistency holds for both regression scenarios under consideration.

Table 2. Experimental Parameters Setup

Parameter	Value	Parameter	Value
Layers	2	Hidden Size	16
Input Size	Feature vector size[7]	Output Size	1[8]
Mini-batch Size	512	Hidden Size	16
Optimizer	Adam [Kingma and Ba 2014]	Learning Rate	0.001

To avoid biasing the training towards any particular edge feature, training batches were generated without ensuring that: i) each batch maintains the same distribution for any edge feature, and ii) each batch exhibits a distribution similar to the entire training set. Apart from biasing the training process, ensuring that batches maintain the distribution of edge features from the entire training set would require preprocessing steps, which extrapolate the scope of this work.

Before training, all road network graphs were split into five disjoint subsets (i.e., 5-fold). Each model was then trained once for each fold, in a training session that considered the fold as the test data (i.e., 20% of the road network edges) and the remaining folds (i.e., 80% of the road network edges) as the training data. Training data was additionally randomly split into training (60%) and validation (20%) sets.

For urban road networks with varying edge lengths and driving speeds, edge-free-flow travel time might vary on wider scales. Therefore, the training cost function is based on a loss value defined by the Mean Absolute Error (MAE), such that errors with different magnitudes will be penalized equally.

All experiments have been run with the following hardware specs: 12th Gen Intel(R) Core(TM) i7-12700F Processor with 12 cores (2 Threads per core), CPU max frequency of 4900.0000 MHz, and minimum frequency of 800.000 MHz; 32GB RAM; NVIDIA GeForce RTX 3060 Ti Graphic Card with 8 GB GDDR6 memory available and 4864 CUDA cores.

Figure 5 shows the results, for each, city of the average *self-test* regression error of interactional fusion RFNs (RFN-AI and RFN-NAI) and additive fusion RFNs (RFN-AA and RFN-NAA). Light blue bars show results for attentional RFNs, dark blue bars for non-attentional RFNs, and black lines represent 95% confidence intervals.

In Figure 5 the same RFN presents different regression errors when applied to different cities. An example is RFN-AI which presents lower self-test MAE values for Aalborg and Manhattan than it does for Niteroi and Seattle. In general, interactional RFNs present better regression results than additive RFNs. Moreover, there are no relevant

[7] Node, edge, or between-edge.

[8] Only the edge representation output was considered since the experiment explores edge regression.

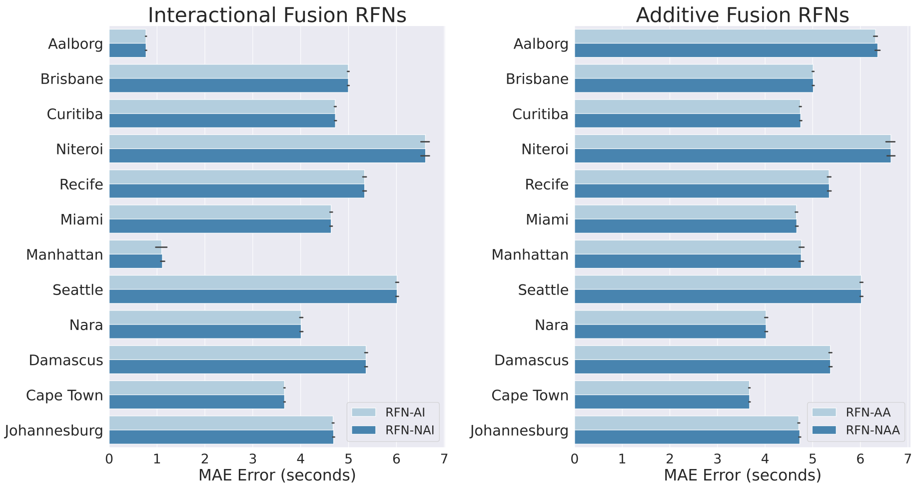


Figure 5. Average (5-fold) self-test regression error (MAE) for interactional and additive fusion RFNs.

differences between RFN-AI and RFN-NAI, or between RFN-AA and RFN-NAA, for any city. Therefore, results suggest that the attentional aggregator does not enhance the learning capacity of RFNs, for this regression task.

Figure 6 shows the average *extrapolation-test* regression error of interactional and additive RFNs trained with a single city data and tested with data from all other cities.

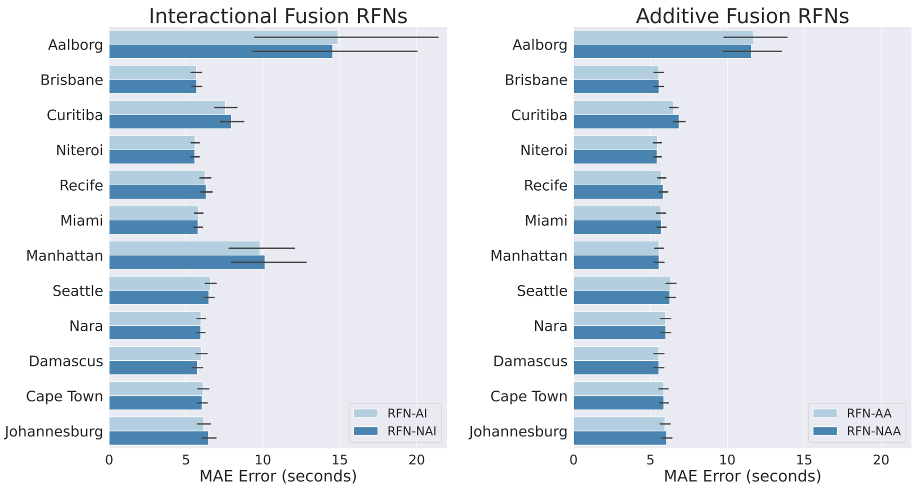


Figure 6. Average extrapolation-test regression error (MAE) for different RFNs when trained for one city (vertical labels) but tested with data from all other cities.

As expected, Figure 6 shows higher regression errors than Figure 5, i.e., the task of training an RFN for one city and testing it to the other cities is even more challenging. As occurred in the *self-test* the attentional aggregator does not enhance the learning capacity of the models. However, the results are slightly better for additive variants than for interactional fusion variants. It is important to point out that RFNs trained with road networks that performed best in the *self-test* (i.e., Aalborg and Manhattan) were the two worst performers on the *extrapolation* test.

Figure 7 presents the Pearson correlation between *self-test* and *extrapolation-test* regression errors of cities for interactional and additive fusion RFNs. The Pearson correlations are 0.43, 0.42, -0.81, and -0.82 for RFN-AA, RFN-NAA, RFN-AI, and RFN-NAI, respectively. The positive correlations between their *self* and *extrapolation* tests enforces that additive RFNs extrapolate better than interactional RFNs which exhibit negative correlations.

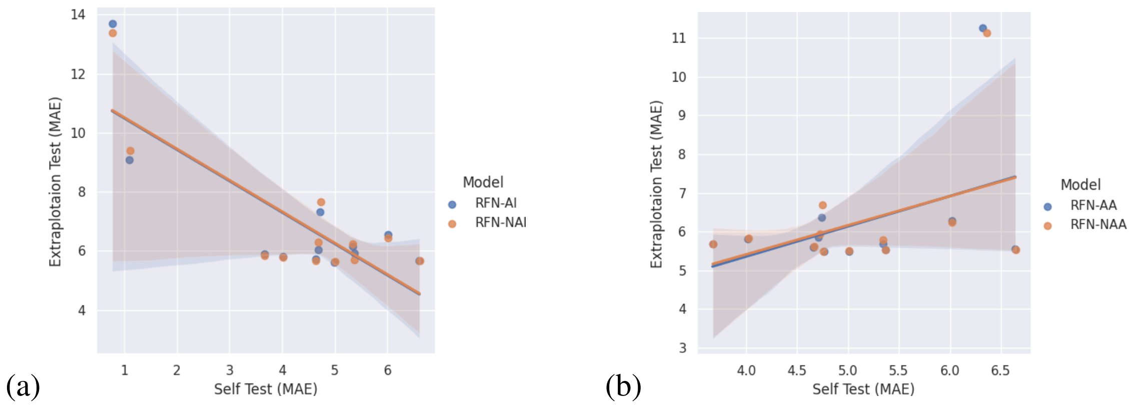


Figure 7. Correlation between test performances (MAE) for RFNs with (a) interactional and (b) additive fusion.

Figure 8 details the *extrapolation-test* of each RFN variant for each city. Each row contains the average *extrapolation-test* regression error of a model trained with data of a city (row label) and tested individually on all remaining cities (column labels). The lowest error in a row often occurs in the diagonal entry as expected, i.e., for the *self-test* scenario. In addition, note that entries in the heatmap are not symmetric. Therefore, the results of training an RFN for city A and testing in city B are not the same as training for B and testing in A.

The idea of extrapolating learned knowledge to other cities should be better investigated by relating road network global features to the corresponding regression errors. Figure 9 presents the heatmap of Pearson correlations between road network features and the corresponding regression error of each RFN variant, using data of all cities for both tests (*self-test* and *extrapolation-test*).

According to Figure 9, the numbers of nodes and edges do not show any relevant effect in the RFN regression performance, i.e., no relevant correlation between those features and test performance was found. The average degree, usually low for urban networks, circuity average (ς), and orientation order (ϕ) are also not relevantly correlated with the test performance for the set of explored cities.

Additionally, the road network global features that present relevant correlations with the regression performance are semantically connected to what is expected to affect the static free-flow time: the average degree, the average streets per node (i.e., street intersections), the edge length, legal speed distribution, and travel-time distribution. This result adds up to [Jepsen et al. 2020] findings, since RFNs present the behavior of weighting semantically connected features for the explored regression task.

The average edge length (in meters) presents a relevant positive correlation with

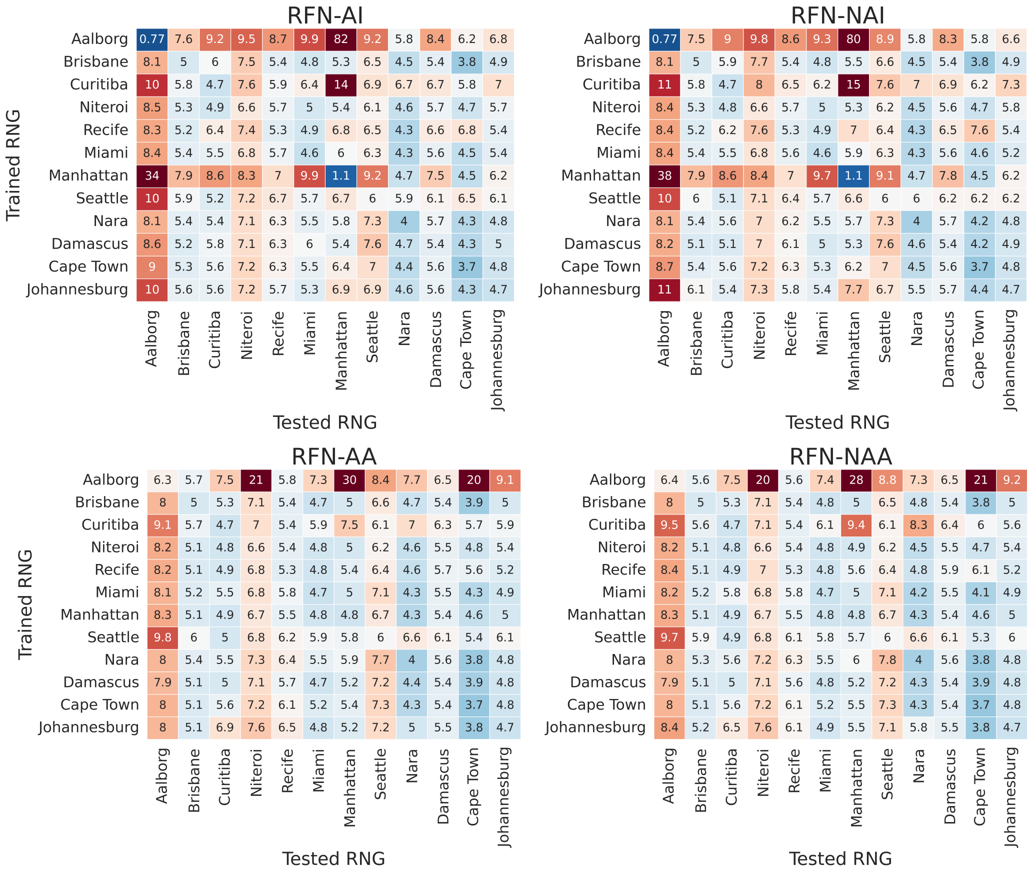


Figure 8. Heatmap of average *extrapolation-test* regression error (MAE) of each RFN variant when trained for one city (row label) and tested with data of another city (column label)

additive RFN performance for both tests. A positive correlation means that as the average edge length increases also does the test MAE value. Additive RFNs then, might perform better on regressions when trained with road networks with narrow edge length distribution around smaller values. On the contrary, interactional RFNs will extrapolate better when trained with lower average edge length networks, yet its *self-test* presents a relevant negative correlation with such feature. That is, the higher the average edge length, the lower the MAE value for *self-test* .

The average degree and the average streets per node correlate negatively with the travel-time regression performance for both additive RFN tests and for the interactional RFNs *extrapolation test*. There is though a slight positive correlation between such features and the interactional RFNs *self-test* . Results show that interactional RFNs can benefit from road networks' low-density characteristics for the *self-test* , but will be negatively affected by it when trying to extrapolate learned knowledge. Additionally, results indicate that additive RFNs may not benefit from road networks' low-density characteristics.

The legal speed, indistinctly for median or average (both in kilometers per hour), does not present a relevant correlation with any RFNs *extrapolation test* performance but correlates negatively with all RFNs *self-test average* performances. The magnitude of the correlation is smaller for interactional than for additive RFNs, but both seem to benefit

Figure 9. Heatmap of Pearson correlations between a road network feature (column label) and the corresponding regression error of cities for self-test and extrapolation-test in each RFN variant (row label).

from legal speed distributions centered on higher values. It is important to highlight that for urban road networks, the scale of legal speed distribution will usually be concentrated between 10 and 80 kilometers per hour, hardly exceeding 100 or 120 kph.

The *Major Highway Class Rate* does not present a relevant correlation for additive RFNs. This indicates that additive RFNs performance are not affected by the distribution of edge classes. However, it presents a medium positive correlation of 0.41 for RFN-AI and RFN-NAI, particularly for *self-test* . Therefore, although interactional RFNs do specialize the learned knowledge to the training set (i.e., good *self-test* performance), they are not overfitting towards the edge class information, since the higher the concentration towards a class group, the higher the *self-test* MAE value.

While the travel time (in seconds) is not a road network feature considered for training (i.e., it is the feature being predicted), the correlation between cities average travel time and the RFNs performance demonstrates that the distribution of travel time values does not affect the interactional RFNs *self-test* performance. It negatively affects (i.e., positive correlation) interactional RFNs *extrapolation test* performance though, similarly to how it affects additive RFNs *self-test* performance. That is, the higher the average travel time (i.e., the travel time values distribution wideness), the higher the MAE value for *self-test* (i.e., the performance decreases).

5. Conclusion

The present work explored four Relational Fusion Networks variants in a regression task (predicting the travel time of roads for 12 different cities). It tested two RFNs using additive and two using interactional fusions. The comparison between RFNs using interactional fusion and those employing additive fusion revealed that both variations present much higher errors when trying to extrapolate knowledge than when applying it to the same city used for training. However, additive fusion exhibited lower mean absolute errors when extrapolating the learned knowledge to other cities. The experiments also show that, regardless of the fusion operator employed, the attentional aggregator does not enhance RFN's learning capacity compared to the non-attentional approach in the regression task.

Also, RFN-AI and RFN-NAI presented relevant and negative correlations between self-test and the extrapolation test, indicating that RFNs with interactional fusion operators tend to specialize better to the training dataset and will not extrapolate well for different cities. The same does not apply to RFN-AA or RFN-NAA (additive fusion), where there is a medium and positive correlation between self and generalization test performances. When trying to find correlations between road network global features and their learning and generalization powers, results show that RFNs do weight road network global features that are semantically connected to the predicted edge feature (free-flow travel time), adding up to the results on RFNs for road network learning.

Future work can explore different training approaches that can help diversify road network information available on training, possibly creating datasets of road networks and considering each city as a dataset entry. This might be beneficial for models' extrapolation performance. Further exploration of road network features, including a wider range of cities, is important to better understand which characteristics affect the regression and extrapolation capacity of RFNs. Advancements towards this direction can serve various goals, such as city-specific specialization or extrapolation across different urban environments of distinct cities. Once the same experimental setup is a limitation, different setups for different RFN variants should also be tested. The enhanced models can then be fine-tuned for static edge regression tasks. Moreover, they can serve as structural components in spatiotemporal graph neural networks, enabling road network latent representations to enhance the performance of temporal regression, such as traffic prediction.

Acknowledgments

The authors thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), Project Smart City Concepts in Curitiba, IPPUC, and Curitiba City Hall. This research was partially supported by the SocialNet project (process 2023/00148-0 of Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP) and by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (processes 313122/2023-7, 314603/2023-9 and 441444/2023-7).

References

Abhinav Nippani, Dongyue Li, H. J. H. N. K. H. R. Z. (2024). Graph neural networks for road safety modeling: Datasets and evaluations for accident analysis.

Albino, V., Berardi, U., and Dangelico, R. M. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, 22:3–21.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V. F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Çaglar Gülçehre, Song, H. F., Ballard, A. J., Gilmer, J., Dahl, G. E., Vaswani, A., Allen, K. R., Nash, C., Langston, V., Dyer, C., Heess, N. M. O., Wierstra, D., Kohli, P., Botvinick, M. M., Vinyals, O., Li, Y., and Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. *ArXiv*, abs/1806.01261.

Beineke, L. W. and Bagga, J. S. (2021). Line graphs and line digraphs. *Developments in Mathematics*.

Boeing, G. (2017). Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139.

Boeing, G. (2019). Urban spatial order: street network orientation, configuration, and entropy. *Applied Network Science*, 4(1).

Derrow-Pinion, A., She, J., Wong, D., Lange, O., Hester, T., Perez, L., Nunkesser, M., Lee, S., Guo, X., Wiltshire, B., Battaglia, P. W., Gupta, V., Li, A., Xu, Z., Sanchez-Gonzalez, A., Li, Y., and Velivckovi'c, P. (2021). Eta prediction with graph neural networks in google maps. *Proc. of CIKM*.

Gharaee, Z., Kowshik, S., Stromann, O., and Felsberg, M. (2021). Graph representation learning for road type classification. *Pattern Recognit.*, 120:108174.

Hamilton, W. L. (2020). Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*.

Jepsen, T. S., Jensen, C. S., and Nielsen, T. D. (2020). Relational fusion networks: Graph convolutional networks for road networks. *IEEE Transactions on Intelligent Transportation Systems*, 23:418–429.

Jepsen, T. S., Jensen, C. S., Nielsen, T. D., and Torp, K. (2018). On network embedding for machine learning on road networks: A case study on the danish road network. *2018 IEEE International Conference on Big Data (Big Data)*, pages 3422–3431.

Karimi, K. (2012). A configurational approach to analytical urban design: 'space syntax' methodology. *URBAN DESIGN International*, 17:297–318.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

OpenStreetMap (2017). OpenStreetMap contributors. `https://www.openstreetmap.org`.

Parthasarathi, P., Levinson, D. M., and Hochmair, H. H. (2013). Network structure and travel time perception. *PLoS ONE*, 8.

Sanchez-Lengeling, B., Reif, E., Pearce, A., and Wiltschko, A. B. (2021). A gentle introduction to graph neural networks. *Distill*. https://distill.pub/2021/gnn-intro.

Silva, T. H. and Silver, D. (2024). Using graph neural networks to predict local culture. *arXiv*.

Strano, E., Viana, M. P., da Fontoura Costa, L., Cardillo, A., Porta, S., and Latora, V. (2013). Urban street networks, a comparative analysis of ten european cities. *Environment and Planning B: Planning and Design*, 40:1071 – 1086.

Ying, R., You, J., Morris, C., Ren, X., Hamilton, W. L., and Leskovec, J. (2018). Hierarchical graph representation learning with differentiable pooling. In *Neural Information Processing Systems*.

Yu, T., Yin, H., and Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *International Joint Conference on Artificial Intelligence*.

Zhang, L. and Long, C. (2023). Road network representation learning: A dual graph-based approach. *ACM Transactions on Knowledge Discovery from Data*, 17:1 – 25.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., and Li, H. (2020). T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21:3848–3858.