Metropolis-Scale Road Network Datasets for Fine-Grained Urban Traffic Forecasting

Fedor Velikonivtsev
HSE University, Yandex Research
fvelikon@yandex-team.ru

Oleg Platonov
HSE University, Yandex Research
olegplatonov@yandex-team.ru

Gleb Bazhenov
HSE University, Yandex Research
gv-bazhenov@yandex-team.ru

Mikhail Seleznyov*
Applied NLP Group, Applied AI Institute
simplymike8@gmail.com

Liudmila Prokhorenkova
Yandex Research
ostroumova-la@yandex-team.ru

Abstract

Traffic forecasting on road networks is a complex task of significant practical importance and spatiotemporal graph neural networks (GNNs) have become the most popular approach for this task. The proper evaluation of traffic forecasting methods requires realistic datasets, but current publicly available benchmarks have significant drawbacks, including the absence of information about road connectivity for road graph construction, limited information about road properties, and relatively small size. Further, current datasets mostly contain information about intercity highways with sparsely located sensors, while city road networks arguably present a more challenging forecasting task due to much denser roads and more complex urban traffic patterns. In this work, we provide a more complete, realistic, and challenging benchmark for traffic forecasting by releasing datasets representing the road networks of two major cities, with the largest containing almost 100,000 road segments (more than a 10-fold increase relative to existing datasets). Our datasets contain rich road features and provide fine-grained data about both traffic volume and traffic speed, enabling building of more holistic traffic forecasting systems. We show that most current implementations of neural spatiotemporal models for traffic forecasting have problems scaling to datasets of our size. To overcome this issue, we propose an alternative approach to neural traffic forecasting that uses a GNN without a dedicated module for temporal sequence processing, thus achieving much better scalability, while also demonstrating stronger forecasting performance. We hope our datasets and modeling insights will serve as valuable resources for research in traffic forecasting. Our code and datasets are available on GitHub.

1 Introduction

Traffic forecasting on road networks is an important task with significant practical implications for urban planning, logistics optimization, and the daily experience of commuters (Li et al., 2018; Yu et al., 2018; Derrow-Pinion et al., 2021; Lim & Zohren, 2021; Jiang & Luo, 2022). In recent years,

*Work done during employment at Yandex.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Advancing Graph Machine Learning.

substantial efforts from the machine learning community have been dedicated to this challenge, with spatiotemporal graph neural networks (GNNs) emerging as the dominant methodology due to their inherent ability to model complex spatial and temporal dependencies (Cini et al., 2023).

However, the development and proper evaluation of advanced traffic forecasting methods depend critically on the availability of realistic and comprehensive benchmarks. Unfortunately, current publicly available traffic datasets have significant drawbacks that hinder progress in the field. In the existing traffic forecasting benchmarks (Jagadish et al., 2014; Li et al., 2018; Yu et al., 2018; Guo et al., 2019; Song et al., 2020; Liu et al., 2023), nodes represent sensors located on roads that measure traffic speed or volume, and edges are constructed based on location proximity (road travel distance between the sensors). These sensors are sparsely distributed and are mostly located on intercity highways, which leads to a number of limitations. First, the overall number of locations (road segments) with available measurements is relatively small, ranging from 207 to 8,600 in currently available datasets. Second, there is no graph structure based on the road connectivity available between the sensors. Thus, in the existing datasets, graph edges are heuristically constructed based on the road distances, without consideration for the natural graph structure arising from road segment adjacency. Finally, since sensors are typically located on intercity highways, their measurements fail to capture complex urban traffic within cities, which is a significant limitation, since traffic conditions within large cities affect daily commutes of millions of people.

To address these problems, our work provides a realistic and challenging benchmark specifically tailored for urban traffic forecasting. We release novel datasets representing the detailed road networks of two major cities. The largest of these datasets encompasses information for almost 100,000 distinct road segments of a large city with approximately 5.5 million residents. Our datasets contain rich road features and provide fine-grained temporal data capturing both traffic volume and traffic speed, enabling the development and evaluation of more holistic and nuanced traffic forecasting systems.

Using our datasets, we examine several existing implementations of neural traffic forecasting models and show that most of them struggle to scale to data of this magnitude. To overcome this issue, we propose an efficient approach to neural traffic forecasting that uses a GNN without a dedicated module for temporal sequence processing, thus achieving much better scalability, while also demonstrating stronger forecasting performance.

We hope our datasets and modeling insights will advance research in traffic forecasting and the related fields of urban computing and smart city development. For a more detailed analysis, see the extended version of this paper (Velikonivtsev et al., 2025).

2 Limitations of current datasets

By far the most popular datasets for traffic forecasting are `METR-LA` and `PEMS-BAY` introduced by Li et al. (2018). In these datasets, nodes represent sensors located on roads that measure traffic speed, and edges are constructed based on location proximity (road travel distance between the sensors). `METR-LA` is based on data from loop detectors in the highways of Los Angeles County (Jagadish et al., 2014) and `PEMS-BAY` is based on data from California Department of Transportation (CalTrans) Performance Measurement System (PeMS, Chen et al., 2001). Some works also use other datasets collected from the same PeMS data source: these datasets may include different subsets of sensors or measurements during different periods of time, but the general structure of these datasets is mostly the same (Yu et al., 2018; Guo et al., 2019; Song et al., 2020). Most works on GNN-based traffic forecasting evaluate their models exclusively on `METR-LA`, `PEMS-BAY`, or other datasets obtained from the PeMS data.

We note that these standard datasets are extremely small: `METR-LA` has only 207 nodes (sensors), while `PEMS-BAY` has only 325 nodes. Other traffic forecasting datasets obtained from the PeMS data also typically have up to a few hundred nodes. Recently, a larger dataset based on PeMS data was proposed: `LargeST` (Liu et al., 2023) with 8,600 nodes, which is still relatively small compared to the amount of data that needs to be processed by traffic forecasting systems in large cities. The small size of standard datasets de-emphasizes model efficiency and leads to proposed models being very resource-intensive and thus not scalable to real-world applications, as we will discuss later.

To obtain a graph structure, previous works (Li et al., 2018; Yu et al., 2018; Wu et al., 2019; Liu et al., 2023) connect two sensors if the road network distance between them is below a certain heuristically chosen threshold. The real road graph structure cannot be used, since sensors are very sparsely

Table 1: Dataset characteristics

dataset	speed	volume	# nodes	# node attributes	real road connectivity	reference
METR-LA	✓	✗	207	3	✗	Li et al. (2018)
PEMS-BAY	✓	✗	325	3	✗	
PeMSD7(M)	✓	✗	228	6	✗	Yu et al. (2018)
PeMSD7(L)	✓	✗	1,026	0	✗	
PEMS03	✗	✓	358	1	✗	Song et al. (2020)
PEMS04	✗	✓	307	0	✗	
PEMS07	✗	✓	883	0	✗	
PEMS08	✗	✓	170	0	✗	
LargeST	✗	✓	8,600	9	✗	Liu et al. (2023)
city-traffic-M	✓	✓	53,530	26	✓	ours
city-traffic-L	✓	✓	94,009	26	✓	

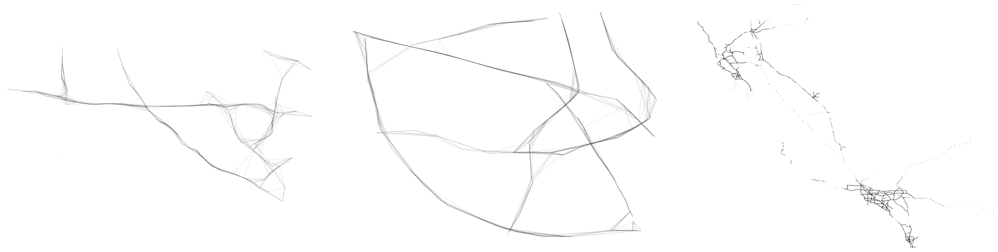
located and thus do not provide information for most existing roads. Thus, the only option is to use a heuristic for constructing a graph in the absence of information about the real road network connectivity. As a result, the real network topology is not provided with any of the standard datasets, which is a significant limitation.

Further, in all currently available traffic forecasting datasets, sensors (graph nodes) are sparsely distributed and only cover a relatively small number of roads. We provide the visualizations of the geographic distribution of sensors in METR-LA, PEMS-BAY, and LargeST datasets in Figure 1. It can be seen that sensors in these datasets are sparse and most of them have only two direct neighbors in the road graph (the sensors right before and after them on the same road), with only a small share of sensors located near intersections. This limits the possibility of using these datasets to study complex traffic patterns. The reason for this is that these datasets mostly focus on large but sparsely located intercity highways. At the same time, densely located city streets are almost not represented in these datasets. However, urban traffic is arguably more complex, presents unique patterns, and is more challenging to forecast. The importance of traffic data available on every road segment of a city was recently discussed by Xu et al. (2024). Since obtaining such data is challenging, the authors rely on sparse data from open public data sources and use a complex procedure to generate an estimate of road-level city traffic. In contrast, we have access to GPS signals from cars for all road segments in the considered cities, which provides much more reliable traffic estimates.

3 CityTraffic datasets

In our work, we present the first openly available datasets for large-scale and fine-grained study of urban traffic. We collect two spatiotemporal graph datasets from two major cities: city-traffic-M with more than $50,000$ nodes and city-traffic-L with almost $100,000$ nodes. These datasets differ significantly from the previous traffic forecasting datasets in what the graphs represent and how they are constructed. While previous datasets only have information about traffic at the locations of sensors, which are only placed at some roads and are generally sparse, the information in our datasets was obtained from GPS measurements rather than sensors, and therefore the measurements are available at a fine-grained level of individual road segments. Thus, our graphs have nodes corresponding to *all road segments in the two considered cities*. Further, while previous datasets construct edges heuristically based on travel distance between sensors, our graph has edges representing actual road connectivity, which can provide much more information. In our graphs, a directed edge connects two road segments if they are incident to each other and moving from one segment to the other is permitted by traffic rules. Next, our datasets have rich node features describing the properties of road segments, including speed limits — important information absent from all widely used traffic forecasting datasets. Our datasets are also the first providing information on traffic volume and traffic speed simultaneously, allowing for a more holistic approach to traffic forecasting. Thus, our datasets represent a realistic setting of traffic forecasting by a traffic monitoring system, which contrasts with the previous datasets that only roughly approximate it due to incomplete data. Some characteristics of our and existing datasets are shown in Table 1.

What makes our datasets fundamentally different from the currently widely used ones is that they focus on urban traffic with its high road density and complex patterns and dynamics. We provide



(a) `METR-LA` (b) `PEMS-BAY` (c) `LargeST`

Figure 1: Visualization of existing traffic forecasting datasets. Nodes correspond to sensors; graph structure is heuristically constructed based on road distances; layout is defined by sensor locations.

(a) `city-traffic-M` (b) `city-traffic-L`

Figure 2: Visualization of the proposed datasets. Nodes correspond to road segments; graph structure is defined by road adjacency; layout is defined by segment locations.

the visualizations of our datasets in Figure 2. It can be seen that our road networks are much more interconnected and present more complex structural patterns than in the previous datasets.

For each road segment, we provide two dynamic variables: current traffic speed and volume, both estimated using high-resolution GPS signals transmitted by vehicles. This data is provided at a 5-minute granularity, spanning from July 1st, 2024 to November 1st, 2024. When no vehicles with GPS enabled passed on a certain road segment, missing values for traffic speed may appear. For example, in `city-traffic-L`, the proportion of missing speed values ranges from 5% to 25%, depending on the time of day (higher missing value proportions typically appear at nighttime, when there are few vehicles on the roads). This level of missing data is consistent with challenges that real-world traffic forecasting systems face.

Finally, for each road segment, we provide 26 static attributes that describe various properties of the segment, including its length, speed limit, coordinates of the segments' endpoints, quality of road surface, indicators of the presence of a masstransit lane, a crosswalk, restrictions for certain types of vehicles, and so on. Road attributes are a mixture of numerical and categorical features. More detailed information about the proposed datasets can be found in Appendix B and C.

Finally, for each road segment, we provide 26 static attributes that describe various properties of the segment, including its length, speed limit, coordinates of the segments' endpoints, quality of road surface, indicator to masstransit lane, presence of crosswalk, restriction for certain types of vehicles, and so on. Road attributes are a mixture of numerical, categorical, and binary features. More detailed information about the proposed datasets as well as the extended analysis can be found in Appendix B.

4 Experiments

In this section, we evaluate the scalability and forecasting performance of existing neural spatiotemporal models on our large and fine-grained traffic datasets. We benchmark several established architectures and highlight their limitations in handling datasets of our size and complexity. To address these limitations, we then introduce a simple but effective model that scales well to large dataset sizes and also outperforms existing baselines in forecasting accuracy. In all experiments, we make forecasts for 12 steps (1 hour) ahead and report the error averaged over the 12 steps and all road segments. We provide more details of our experimental setup in Appendix D.

4.1 Models

Simple graph-agnostic baselines First, we evaluate several naive baselines to establish reference points for model performance. These baselines rely on simple heuristics derived from past traffic values. The simplest of these baselines is the *previous* strategy, which predicts the most recently observed value at each road segment. We also consider baselines that use the daily and weekly periodicity in traffic patterns, which is commonly observed in urban traffic dynamics. Specifically, we predict traffic speed/volume by using the corresponding value either one day or one week ago from the target timestamp. We refer to these methods as *previous 1 day/week ago*. Next, we include simple statistical baselines such as the global *mean*, *median*, as well as *node-wise mean* and *node-wise median* which are the mean and median computed independently for each road segment. These naive baselines do not use the graph structure. Further, we evaluate a linear model which is a simple learnable graph-agnostic baseline.

Spatiotemporal baselines For our experiments, we have selected four popular models from the literature that are frequently used by other works on graph-based time series forecasting and that could scale to our datasets (see below). To process the temporal dimension of the data, they utilize either recurrence or convolution mechanisms:

- DCRNN (Li et al., 2018) — a diffusion convolutional recurrent neural network that exploits recurrent cells supplied with a graph convolution operation;
- GRUGCN (Gao & Ribeiro, 2022) — a combination of recurrent temporal encoder and graph convolutional spatial encoder, which are stacked consecutively;
- STGCN (Yu et al., 2018) — a spatiotemporal graph neural network that is composed of alternating temporal and graph convolution operations;
- GWN (Wu et al., 2019) — a spatiotemporal graph neural network that stacks graph convolutions and causal dilated temporal convolutions.

For our experiments, we adapt the implementations from the `LargeST` repository (Liu et al., 2023).

A scalable traffic forecasting approach Our datasets are much larger than the ones currently used in the literature. Thus, they present a significant scaling challenge to deep learning models. We investigated the models available in Torch Spatiotemporal (Cini & Marisca, 2022) as well as in the codebase of `LargeST` (Liu et al., 2023), the largest previous traffic forecasting dataset, and found that *only four models listed above* can be trained on `city-traffic-M` on a GPU with 80GB VRAM. However, even these models require very long training time. This led us to investigate the sources of the inefficiency of the currently available methods and look for ways to design more scalable models.

The GNN-based models for traffic forecasting proposed in previous works typically use recurrence, convolution, or attention mechanisms to process the temporal dimension of the data. However, these mechanisms are relatively resource-intensive since they maintain a separate vector representation for each timestamp in the lookback window for each node in the graph. Thus, for a dataset with n graph nodes, a lookback window of t timestamps, and a hidden dimension of size d, each layer of such models requires at least $\mathcal{O}(ntd)$ memory. While the aforementioned mechanisms differ in their required number of operations (and their ability to parallelize them), for all of them it is at least linear in the number of vector representations, which is $\mathcal{O}(nt)$, and each of these representations is involved in at least one matrix-vector multiplication, so each layer also performs at least $\mathcal{O}(ntd^2)$ operations. Thus, for datasets with a large number of nodes or a necessity to use a long lookback window, the time and memory requirements of such models quickly become prohibitive.

However, in the time series literature, several recent works have been exploring an alternative direction that allows processing the temporal dimension much more efficiently (Oreshkin et al., 2019; Zeng et al., 2023; Zhang et al., 2022; Das et al., 2023; Li et al., 2023; Yi et al., 2024). These works concatenate all past time series values in the lookback window into a single input vector and transform it into a single vector representation (e.g., with one linear layer). This vector representation is then processed with an MLP-based model (Zeng et al. (2023) do not use an MLP at all and directly make predictions with just one linear layer). Despite the simplicity of this approach, it has been shown that it can compete with other models or even outperform them, all while being significantly, sometimes orders of magnitude, more efficient.

In this work, we propose to adapt this approach to graph-based traffic forecasting. Specifically, we take the idea of encoding each time series in a multivariate dataset into a single vector representation with a linear layer and adapt it to graph-based forecasting setting by replacing the following MLP with a GNN. Since, crucially, this approach requires maintaining only a single vector representation per graph node (in contrast to t vector representations required by other methods), in the case of graph-based traffic forecasting, it has per-layer memory complexity of only $\mathcal{O}(nd)$, which allows it to efficiently scale to much larger datasets, such as the ones we propose in our work.

Our proposed model consists of a linear layer that encodes the temporal information of a single time series into a latent vector representation and a multilayer GNN that allows representations of different time series to interact according to the graph connectivity. According to the categorization of temporal graph models introduced by Gao & Ribeiro (2022), models using our approach are *time-then-graph* models (in contrast to more popular *time-and-graph* models), but their component for processing the temporal dimension is extremely simplified (e.g., to a single linear layer) for the purpose of efficiency.

Our approach can use any GNN architecture. For our experiments, we use GNNs with two popular spatial graph convolution mechanisms: mean aggregation, which was popularized in modern GNNs by Hamilton et al. (2017), and transformer-like multihead attention neighborhood aggregation that has been popularized in GNNs by Shi et al. (2021) (note that this is local attention over graph neighbors, not global attention over all nodes). We refer to these models as GNN-`Mean` and GNN-`TrfAttn`. Following Platonov et al. (2023); Bazhenov et al. (2025), we augment our GNNs with skip connections (He et al., 2016), layer normalization (Ba et al., 2016), and MLP blocks, which often significantly improve their performance.

We show that our approach, despite its simplicity and efficiency, usually leads to better forecasting quality than prior methods. We also show that its efficiency allows it to use much longer lookback windows with a negligible impact on computational cost (since only the size of a single linear layer is affected), which often further improves the forecasting performance. We hope that these findings will encourage further development of efficient methods for traffic modeling and graph-based spatiotemporal forecasting in general.

4.2 Results

We compare the performance of the considered models; the results are shown in Table 2. Following previous studies, we use the lookback window of 12. Among the considered naive baselines, the best results for the task of traffic volume prediction are achieved by the predictor taking the value one week ago relative to the target timestamp. For speed prediction, the best naive predictor employs the latest known value. These metric values should serve as a necessary sanity check to ensure that the designed models actually capture useful information for the given forecasting task. Thus, as expected, the linear model consistently outperforms the presented naive baselines, which demonstrates that using historic observations is essential for precise traffic forecasting. More advanced spatiotemporal methods, in turn, have better performance than all graph-agnostic approaches, which indicates that using structural information about the road network is important for accurate traffic forecasting. Among the considered graph-aware methods, the best results are almost always achieved by the proposed GNN-`TrfAttn` model. These results suggest that models with more flexible mechanism for aggregating structural information, such as Transformer self-attention, have more potential for generalizing to complex traffic networks, so they should be especially considered when developing more effective backbones for spatiotemporal traffic forecasting. We provide additional ablations on lookback window size and runtime measurements in Appendix E.

Table 2: Performance of baselines and spatiotemporal models, MAE on the test set is reported. `OOM` indicates experiments which exceeded GPU memory (80GB).

		city-traffic-L		city-traffic-M	
		volume	speed	volume	speed
naive baselines	mean	9.413	11.828	2.848	11.704
	median	7.577	11.551	2.063	11.161
	node-wise mean	5.355	5.912	1.527	5.448
	node-wise median	5.297	5.818	1.491	5.375
	previous	2.641	4.576	0.957	4.240
	previous 1 day ago	2.808	5.827	0.988	5.550
	previous 1 week ago	2.540	5.700	0.926	5.476
	linear model	2.284 ± 0.000	4.229 ± 0.001	0.806 ± 0.000	3.951 ± 0.001
spatiotemporal	DCRNN	2.212 ± 0.054	3.988 ± 0.012	0.765 ± 0.007	3.704 ± 0.014
	GRUGCN	2.255 ± 0.011	4.074 ± 0.014	0.765 ± 0.011	3.717 ± 0.020
	STGCN	OOM	OOM	0.777 ± 0.011	3.663 ± 0.016
	GWN	2.368 ± 0.006	4.516 ± 0.008	0.792 ± 0.004	4.204 ± 0.083
	GNN-Mean	2.038 ± 0.021	3.753 ± 0.005	0.737 ± 0.004	3.397 ± 0.011
	GNN-TrfAttn	2.050 ± 0.029	3.724 ± 0.010	0.733 ± 0.006	3.353 ± 0.007

5 Conclusion

Our work makes two contributions to the field of traffic forecasting. First, we introduce two large-scale datasets for fine-grained urban traffic forecasting: `city-traffic-M` and `city-traffic-L`. These datasets address critical limitations of existing benchmarks by providing the detailed coverage of dense urban roads rather than sparse intercity highways; actual road network connectivity instead of heuristically defined graphs; rich road segment features including speed limits; and information about both traffic volume and traffic speed. By capturing the complex road structure and traffic conditions of two major cities, we provide the community with the data needed for the development and rigorous evaluation of holistic traffic forecasting systems. Second, our empirical analysis reveals scalability issues in existing neural traffic forecasting models, which complicate their application in real-world systems. To address these issues, we propose an efficient GNN-based approach that achieves both much better scalability and superior forecasting performance.

Future opportunities The proposed traffic datasets open several interesting avenues for future research. The first direction is the development of efficient traffic forecasting methods. While our proposed model shows decent performance, there is a continuous need to develop even better GNN architectures or alternative deep learning models that can efficiently process large urban road networks without sacrificing forecasting quality. Also, the presence of real connectivity structure and rich node features enables the development of models that can effectively exploit such information. Moreover, each of our datasets contains two dynamic target variables — traffic volume and traffic speed, which can be used to investigate the performance of forecasting models in multitask settings. Fine-grained forecasting methods developed and evaluated with the help of our datasets can be integrated into adaptive traffic control systems, dynamic routing algorithms for logistics and navigation, and long-term urban infrastructure planning tools.

Acknowledgments

We thank Ivan Gorin, Aleksei Istomin, and Alexandr Ruchkin for their help with collecting the data for our datasets.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bai, L., Yao, L., Li, C., Wang, X., and Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems*, 33:17804–17815, 2020.

Bazhenov, G., Platonov, O., and Prokhorenkova, L. GraphLand: Evaluating graph machine learning models on diverse industrial data. *Advances in Neural Information Processing Systems*, 2025.

Chen, C., Petty, K., Skabardonis, A., Varaiya, P., and Jia, Z. Freeway performance measurement system: mining loop detector data. *Transportation research record*, 1748(1):96–102, 2001.

Cini, A. and Marisca, I. Torch Spatiotemporal, 3 2022. URL https://github.com/TorchSpatiotemporal/tsl.

Cini, A., Marisca, I., Zambon, D., and Alippi, C. Graph deep learning for time series forecasting. *arXiv preprint arXiv:2310.15978*, 2023.

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., and Yu, R. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

Derrow-Pinion, A., She, J., Wong, D., Lange, O., Hester, T., Perez, L., Nunkesser, M., Lee, S., Guo, X., Wiltshire, B., et al. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pp. 3767–3776, 2021.

Gao, J. and Ribeiro, B. On the equivalence between temporal and static equivariant graph representations. In *International Conference on Machine Learning*, pp. 7052–7076. PMLR, 2022.

Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 922–929, 2019.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., and Shahabi, C. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

Jiang, W. and Luo, J. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 207:117921, 2022.

Lan, S., Ma, Y., Huang, W., Wang, W., Yang, H., and Li, P. DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International Conference on Machine Learning*, pp. 11906–11917. PMLR, 2022.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

Li, Z., Qi, S., Li, Y., and Xu, Z. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023.

Lim, B. and Zohren, S. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Liu, X., Xia, Y., Liang, Y., Hu, J., Wang, Y., Bai, L., Huang, C., Liu, Z., Hooi, B., and Zimmermann, R. LargeST: A benchmark dataset for large-scale traffic forecasting. *Advances in Neural Information Processing Systems*, 36:75354–75371, 2023.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., and Prokhorenkova, L. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? *International Conference on Learning Representations (ICLR)*, 2023.

Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., and Sun, Y. Masked label prediction: Unified message passing model for semi-supervised classification. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2021.

Song, C., Lin, Y., Guo, S., and Wan, H. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 914–921, 2020.

Velikonivtsev, F., Platonov, O., Bazhenov, G., and Prokhorenkova, L. Fine-grained urban traffic forecasting on metropolis-scale road networks. *arXiv preprint arXiv:2510.02278*, 2025.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019.

Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., and Xiong, H. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020.

Xu, X., Zheng, Z., Hu, Z., Feng, K., and Ma, W. A unified dataset for the city-scale traffic assignment model in 20 US cities. *Scientific data*, 11(1):325, 2024.

Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., and Niu, Z. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.

Yu, B., Yin, H., and Zhu, Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

Zhang, T., Zhang, Y., Cao, W., Bian, J., Yi, X., Zheng, S., and Li, J. Less is more: Fast multivariate time series forecasting with light sampling-oriented MLP structures. *arXiv preprint arXiv:2207.01186*, 2022.

Zheng, C., Fan, X., Wang, C., and Qi, J. GMAN: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1234–1241, 2020.

A Background on traffic forecasting

The goal of traffic forecasting is to predict future traffic conditions (e.g., traffic speed and/or traffic volume) based on historical observations. Typically, observations are provided by sensors located at specific road segments. Traditional approaches that rely on statistical models, such as ARIMA or Kalman filters, often fall short in capturing the complex, nonlinear spatiotemporal dependencies present in real-world traffic systems. Recent advances in deep learning, particularly in representation learning on graphs and sequences, have led to a surge of interest in neural methods for traffic forecasting, aiming to model spatial and temporal components jointly and more effectively.

One of the pioneering works in this direction is Diffusion Convolutional Recurrent Neural Network (DCRNN, Li et al., 2018), which formulates the traffic forecasting problem as a spatiotemporal sequence modeling task, representing the traffic network as a directed graph and utilizing diffusion convolution over the graph structure to capture spatial dependencies, integrated with a recurrent neural network (RNN) to model the temporal component. This work was one of the first to use GNNs in traffic forecasting, so it became the groundwork for many subsequent methods.

Further, Yu et al. (2018) proposed Spatiotemporal Graph Convolutional Network (STGCN) that replaces RNNs with temporal convolutional layers, resulting in improved computational efficiency. This architecture employs separate modules for spatial and temporal components, alternating between graph convolutions for aggregating local spatial information and temporal convolutions for processing sequential information.

Later works sought to address the limitations of previous models by introducing more intricate and flexible mechanisms. For instance, Attention-based Spatial-Temporal Graph Convolutional Networks (ASTGCN, Guo et al., 2019) incorporate spatial and temporal attention to dynamically weigh the importance of different nodes and time steps, potentially improving the model's ability to focus on specific patterns. Graph WaveNet (GWN, Wu et al., 2019) introduces adaptive adjacency matrices and dilated temporal convolutions to enable the model to learn latent spatial structure and long-range temporal dependencies more efficiently. Another work in this direction is Adaptive Graph Convolutional Recurrent Network by (AGCRN, Bai et al., 2020) that learns node embeddings and constructs adaptive graphs dynamically, decoupling model performance from reliance on predefined graph structures.

Further, Zheng et al. (2020) also introduced a fully attention-based architecture in Graph Multi-Attention Network (GMAN), avoiding both recurrent and convolutional components, and combining spatial and temporal attention to dynamically model the spatiotemporal patterns at each time step. Together with other examples, such as Spatial-Temporal Transformer Networks (STTNs, Xu et al., 2020) and Dynamic Spatial-Temporal Aware Graph Neural Network (DSTAGNN, Lan et al., 2022), these works mark a trend in the field towards attention-based models and even more sophisticated methods for capturing complex dependencies in the data.

As can be seen, many recent models incorporate multiple complex components, such as hierarchical attention or adaptive adjacency learning, which can significantly complicate implementation and introduce overheads in computation. Consequently, scaling to large traffic networks with tens of thousands of road segments can become a significant challenge for these models, since implementing and training them efficiently is a non-trivial task, and the real-time deployment of such models can be hindered by their computational complexity.

For most of the discussed models, there are publicly available implementations that have been introduced by the authors of the original works or provided by the authors of existing traffic forecasting benchmarks such as LargeST (Liu et al., 2023). However, as we discuss further, the currently available traffic datasets do not allow us to thoroughly evaluate these implementations and ensure their practical usability for large-scale traffic forecasting, since they do not provide the real road network topology or detailed information about road properties to reliably test the performance of traffic forecasting models.

B Dataset details

The data used in our benchmark is collected from a widely-used online map and navigation service that estimates traffic congestion and travel time using high-resolution GPS signals transmitted by

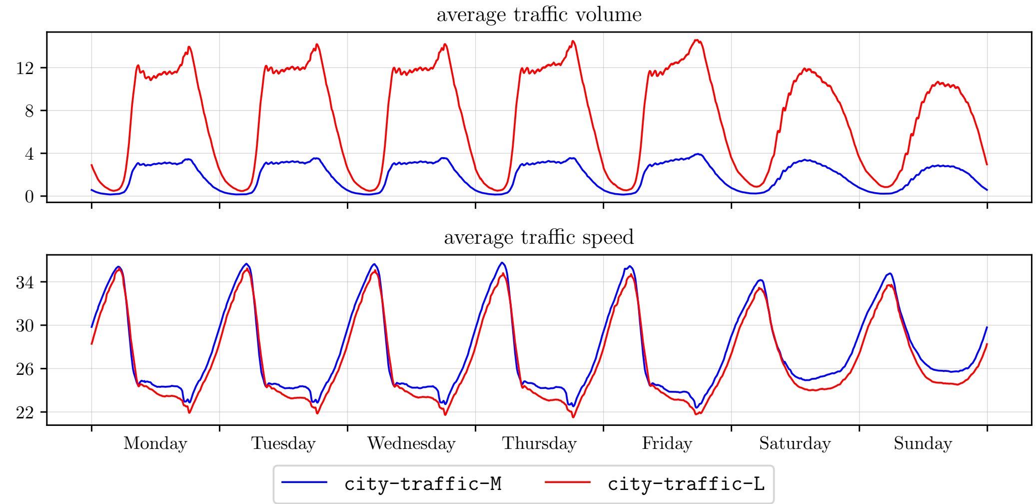


Figure 3: The weekly dynamics of target variables averaged across all roads in the proposed datasets.

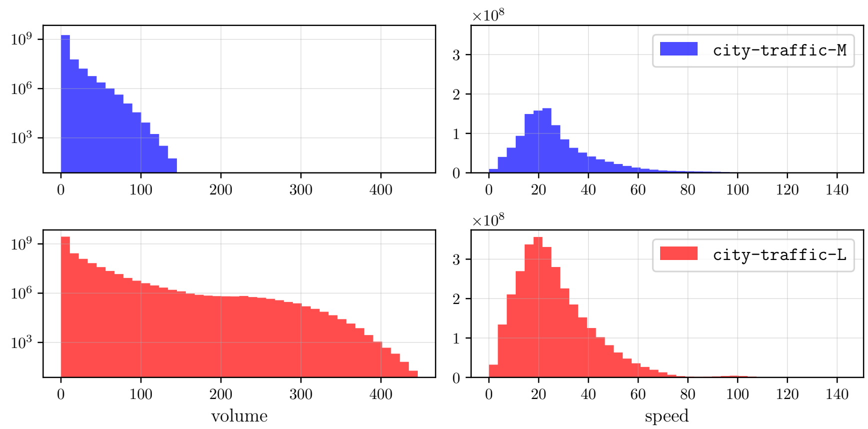


Figure 4: The histograms of traffic volume and speed in the proposed datasets.

vehicles. To select the road segments, we take the central geographic point within each city, consider a circular area of a 15 kilometer radius, and include all road segments located within this area in the dataset. The obtained set of road segments includes the city itself and may also cover some nearby roads.

The traffic volume is estimated based on the number of vehicles that traverse each road segment during a specific time interval, as inferred from aggregated GPS traces. It is important to note that the number of traverses represents an estimate rather than the actual traffic flow, as it is derived solely from vehicles equipped with GPS. Consequently, the reported values systematically underestimate the true traffic volume, but represent the dynamic of the traffic volume well. The speed estimation is also derived from these GPS signals, using a proprietary internal algorithm developed by the service provider. This GPS-based approach offers a significant advantage over traditional loop-detector or camera-based systems by providing fine-grained, diverse, and city-wide coverage without the sparsity typical for fixed sensor infrastructure.

Some characteristics of our datasets are reported in Table 3. Each node in the dataset represents an individual road segment and has a set of 26 attributes. The full list of attributes is provided below:

- `category` — functional category of the road segment (e.g., major arterial, residential, service);

Table 3: Characteristics of new `city-traffic` datasets. Timestamps are in UTC+0 timezone.

	`city-traffic-M`	`city-traffic-L`
# nodes	53,530	94,009
# edges	121,236	164,424
is directed	✓	✓
# timestamps	35,449	35,449
# train timestamps	26,208	26,208
# validation timestamps	4,032	4,032
# test timestamps	5,209	5,209
train start	`Jul 1st 2024 00:00`	`Jul 1st 2024 00:00`
validation start	`Sep 30th 2024 00:00`	`Sep 30th 2024 00:00`
test start	`Oct 14th 2024 00:00`	`Oct 14th 2024 00:00`
test end	`Nov 1st 2024 02:00`	`Nov 1st 2024 02:00`
avg. in-degree	2.264	1.749
avg. out-degree	2.264	1.749
avg. node degree (undirected)	3.652	2.970
Gini coefficient of degree distribution	0.9	0.9

- `edge_type` — encodes the type of connection between the road segments;
- `speed_mode` — type of speed regulation pattern allowed on the segment (e.g., high-speed corridor, restricted-speed street);
- `speed_limit` — the maximum legal speed limit on the segment;
- `region_id` — identifier of the administrative or city district containing the segment;
- `can_bind_to_reverse_edge` — indicates whether the segment allows binding to a reverse-direction edge;
- `dismount_bike` — indicates if cyclists are required to dismount on the segment;
- `has_masstransit_lane` — indicates if the segment has a dedicated lane for public or mass transit;
- `ends_with_crosswalk` — indicates if the segment ends with a pedestrian crosswalk;
- `ends_with_toll_post` — indicates if the segment ends with a toll post;
- `is_in_poor_condition` — indicates whether the road surface is in poor condition;
- `is_paved` — indicates whether the segment is paved;
- `is_restricted_for_trucks` — indicates whether the segment is restricted for trucks;
- `is_toll` — indicates whether the segment is a toll road;
- `access_[0...5]`* — boolean masks for road accessibility by different vehicle types;
- `length` — length of the road segment (in meters);
- `num_segments` — number of consecutive sub-segments composing the road segment;
- `x_coordinate_start` — latitude of the segment's start point;
- `y_coordinate_start` — longitude of the segment's start point;
- `x_coordinate_end` — latitude of the segment's end point;
- `y_coordinate_end` — longitude of the segment's end point.

Note that we apply ordinal encoding to the `speed_limit` feature. Thus, we provide the correspondence of particular feature values and their ordinal codes: `NaN` $\to$ 0; $5\,km/h \to 1$; $20\,km/h \to 2$; $30\,km/h \to 3$; $40\,km/h \to 4$; $50\,km/h \to 5$; $60\,km/h \to 6$; $70\,km/h \to 7$; $80\,km/h \to 8$; $90\,km/h \to 9$; $100\,km/h \to 10$; $110\,km/h \to 11$.

*There is a separate attribute for each of the 6 masks.

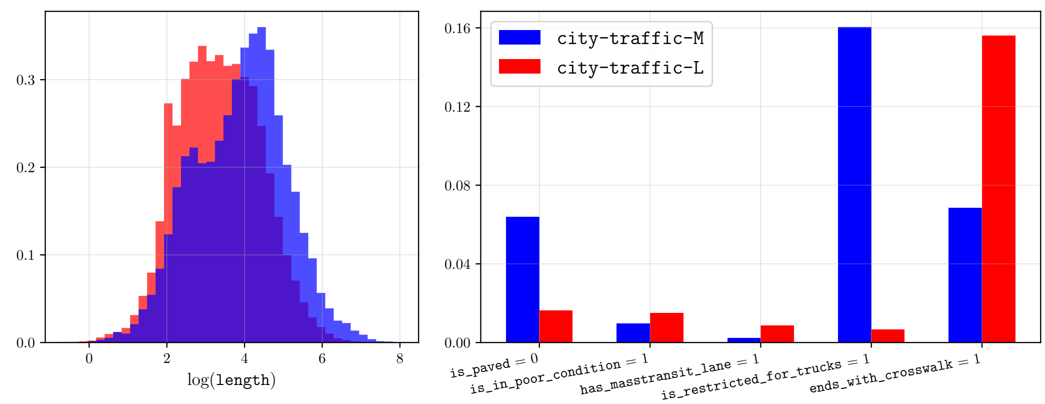


Figure 5: The distribution of some spatial features in the proposed datasets.

The comparative characteristics of our and existing datasets are shown in Table 1. In Figure 3, we visualize the behavior of dynamic variables. Specifically, we report traffic speed, traffic volume, and traffic density which is the ratio of volume to speed. For each variable and each city, we average the values over all road segments in the city. One can clearly see the daily and weekly traffic patterns, e.g., noticeable traffic jams in the evenings on working days. While the average speed in `city-traffic-M` and `city-traffic-L` is similar, traffic volumes differ significantly.

C Differences between `city-traffic-M` and `city-traffic-L`

While both datasets follow the same construction methodology, there are several notable differences between `city-traffic-M` and `city-traffic-L` due to the differences in the corresponding cities, which make the two datasets complementary benchmarks.

In terms of scale, `city-traffic-M` contains 53,530 road segments and 121,236 directed edges, while `city-traffic-L` is almost twice as large, with 94,009 segments and 164,424 edges. The larger size of `city-traffic-L` poses a particular challenge for the scalability of spatiotemporal models, as the number of graph nodes and edges directly determines memory and runtime costs.

In terms of topological properties, the two cities also vary significantly and have a different urban structure. `city-traffic-L` features a complex structure shaped by a large river crossing the metropolitan area, which has led to the development of multiple islands connected by bridges. This creates bottlenecks and high-traffic corridors that models must capture. By contrast, `city-traffic-M` lacks such a riverine structure; its road network is more uniform, with a grid-like arrangement and wide avenues even in the central districts. Average node degree of a road network also differs between the datasets: `city-traffic-M` has an average undirected degree of 3.65, while `city-traffic-L`'s average is 2.97. This reflects the higher density and branching structure of the smaller city versus the sparser but more geographically constrained connectivity of the larger one.

While the average traffic speed values are comparable between the two datasets, the average traffic volume differs significantly: `city-traffic-L` records substantially higher overall volume, reflecting the higher population of the city. The weekly dynamics, shown in Figure 3, indicates more pronounced rush-hour congestion patterns in `city-traffic-L`.

Both datasets provide the same 26 static attributes per segment. However, their distribution is different for the two proposed datasets. As Figure 5 shows, `city-traffic-L` has a greater fraction of paved roads, and there are also notably more road segments with crosswalks at their endpoints. On the other hand, `city-traffic-M` has longer continuous road segments on average, and the fraction of roads restricted for trucks is much greater.

Taken together, the two datasets provide complementary perspectives: `city-traffic-M` highlights fine-grained dynamics in a compact road network, while `city-traffic-L` captures large-scale, heterogeneous urban traffic with more complex network structure. This difference is essential for developing models for diverse city types.

Table 4: Effect of lookback horizon on model performance, MAE on the test set is reported.

	lookback	CityTraffic-L volume	CityTraffic-L speed	CityTraffic-M volume	CityTraffic-M speed
GNN-TrfAttn	12	2.042 ± 0.027	3.818 ± 0.004	0.748 ± 0.008	3.504 ± 0.010
	24	2.033 ± 0.026	3.778 ± 0.006	0.751 ± 0.007	3.457 ± 0.013
	36	2.017 ± 0.010	3.773 ± 0.015	0.744 ± 0.009	3.431 ± 0.010
	48	2.021 ± 0.021	3.761 ± 0.000	0.743 ± 0.005	3.428 ± 0.008
	72	2.021 ± 0.016	3.743 ± 0.002	0.743 ± 0.013	3.414 ± 0.009

Table 5: Training time in hours for different models across two datasets and two lookback window sizes. TLE indicates models that did not converge within a 250 hours time limit.

	CityTraffic-L		CityTraffic-M	
Lookback	12	48	12	48
DCRNN	7.52	31.17	5.13	21.06
GRUGCN	2.24	7.63	1.24	4.12
GWN	6.84	27.81	4.17	17.13
STGCN	26.55	TLE	6.38	211.19
GNN-Mean	1.45	1.79	0.77	0.90
GNN-TrfAttn	1.88	2.09	1.06	1.18

D Experimental setup

We use learnable node embeddings for road segments in addition to their static features. We also use additional temporal features such as day of the week, week of the year, and month of the year. We encode these features both with one-hot encoding and with periodic trigonometric functions.

To ensure comparability across experiments, we fix the effective batch size (number of timestamps at which the prediction is made) to 30 across all datasets and models and adjust gradient accumulation steps as needed. All models are trained using the AdamW optimizer with a fixed learning rate of 0.0003. Training is performed for 5 epochs, and each training run is repeated 3 times to compute the mean and standard deviation.

All experiments are conducted on a single NVIDIA A100 GPU with 80GB of VRAM and 120GB of system RAM. For models that exceed this memory limit, we try to decrease the number of model parameters — if the model still fails after several attempts, we report OOM for it.

Our datasets are available at Kaggle, and our code is provided in our GitHub repository.

E Ablations

Effect of lookback window In the next series of experiments, we vary the lookback window among the following options: $[12, 24, 36, 48, 72]$ and consider the best-performing and efficient model GNN-TrfAttn. As can be seen from Table 4, better results can usually be achieved for larger lookback windows, which proves that more complete information about how the target variable changed in the past is important for more accurate predictions in the future. At the same time, these results show that even such a simple module for processing the temporal component as a linear projection of historical variables into latent space of GNN model allows it to scale to greater amounts of data, while preserving computational efficiency.

Scalability of the models We report the total training time in hours for all evaluated models across different datasets and lookback window sizes of 12 and 48 in Table 5. As the lookback window increases from 12 to 48, the considered sequential models, especially DCRNN, GWN, and STGCN, exhibit significantly worse scaling behavior. In case of STGCN on CityTraffic-L dataset with a lookback of 48, training fails to complete within 250 hours. This poor scalability is attributed to the need to maintain and process an explicit temporal state for each input timestamp, which grows linearly with the lookback size. In contrast, our proposed models GNN-`Mean` and GNN-`TrfAttn` require consistently low training time across all configurations. This demonstrates that such a non-sequential full-batch design is significantly more scalable and computationally efficient, particularly as the temporal input dimension grows. These results highlight the importance of scalability aspect for practical application of traffic forecasting models.

F Limitations

While our benchmark provides novel and valuable data for fine-grained urban traffic forecasting, it has certain limitations. First, we acknowledge that cities in different countries may exhibit different traffic patterns. Consequently, the conclusions drawn from the two cities included in our benchmark may not be directly generalizable to urban environments with substantially different traffic dynamics. Additionally, as our benchmark spans only four months of data, it may not facilitate the evaluation of models designed to capture long-term annual trends. However, we contend that for urban traffic, which is characterized by rapidly changing conditions, the ability to capture local trends, such as recent traffic conditions on a specific road segment and its adjacent segments, is often more critical.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 3 introduces our new traffic datasets, and Section 4 reports the experimental results with our proposed scalable approach.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Appendix F.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the detailed description of our experimental setup, including the full list of hyperparameters, optimization settings, and evaluation protocols in Appendix D. Our source code is released on GitHub.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our source code can be accessed in the GitHub repository, along with the detailed instructions for reproducing all experiments, installing the necessary environment, preprocessing data, and conducting evaluation. Our datasets are shared via Kaggle, and our repository includes the guidelines for downloading and preparing the data.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides a complete specification of the experimental setup in Appendix D. We also describe the full list of used baselines and the evaluation protocol, including model hyperparameters and training settings. Predefined data splits are also provided for each of the introduced datasets. We disclose all commands to reproduce results in the repository.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide the mean and standard deviation of results in all numerical reports, namely, in Tables 2 and 4. The considered naive baselines are deterministic, thus they are reported without standard deviations.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As reported in Appendix D, our experiments are conducted on a single NVIDIA A100 GPU with 80GB of VRAM and 120GB of system RAM.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research aligns with the NeurIPS Code of Ethics, as we ensure ethical conduct in every aspect of the study, including data handling, experimental procedures, and potential societal impact.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work contributes a realistic and large-scale benchmark for urban traffic forecasting, which can support the development of models for improving urban mobility, reducing congestion, and lowering emissions, thereby offering clear societal benefits in sustainability and public infrastructure planning. However, we also recognize potential negative impacts, such as the misuse of traffic prediction systems for surveillance, traffic control biases, or inequitable infrastructure deployment. While our benchmark is constructed from anonymized and aggregated GPS traces, we encourage future users to consider privacy-preserving practices and fairness implications in downstream applications.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not involve models or datasets that pose a high risk for misuse. The released dataset is derived from aggregated and anonymized GPS data, and the models are standard forecasting architectures with no generative or dual-use capabilities.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in our work, including baseline model implementations and prior datasets, are properly cited in the main text. Our empirical study involves only publicly available resources with compatible licenses and respects all terms of use.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release two new large-scale urban traffic forecasting datasets and share our source code for reproducing experimental results. The datasets are accompanied by detailed documentation and usage guidelines. The repository also contains the necessary instructions.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.