Responses to Comments for Paper T-ITS-22-05-1162

Dear Editors & Reviewers:

We would like to express our sincere gratitude to the editor for handling our manuscript "MT-STNet: A Novel Multi-Task Spatiotemporal Network for Highway Traffic Flow Prediction" and to three anonymous reviewers for their insightful comments and valuable suggestions.

Based on all comments and suggestions, the authors conducted major revisions to the paper, which are summarized as follows:

- have redefined the 'long-term' prediction to 'multi-step' forecasting;
- have extended the prediction steps from six to twelve horizons;
- have added comparisons of computational cost in the training and inference phases;
- have redefined the direct distance matrix according to the network science;
- have added more details about parameters, such as λ;
- have improved the font sizes of all figures;
- have added a more detailed embedding process for the embedding layer;
- have provided more details about the significance of multi-task learning;
- have provided sufficient spatial information description for monitoring stations;
- have applied the proper ARIMA (SARIMA) as one of the baselines in experiments and provided detailed processes and codes;
- have added detailed dataset descriptions to the manuscript and GitHub page;
- have added more relative references to the manuscript, such as SARIMA, the reviewer proposed;
- have added the codes and parameters of the proposed model and baselines to my personal GitHub page (https://github.com/zouguojian/Traffic-flow-prediction).

Please check the detailed responses for each comment below, which are marked in red color. All changes made to the manuscript are in yellow so that they will be easily identified. Should you have any questions, please contact us without hesitation. Once again, thank you very much for your work. We hope that the resubmitted manuscript is now suitable for publication in IEEE Transactions on Intelligent Transportation Systems.

Sincerely,
Dr. Guojian Zou

We thank the Editor for organizing the review for our submission. We have carefully followed the reviewers' comments and prepared a major revision in this submission. The detailed comments are addressed point-to-point as follows. We hope that the Editor and the anonymous reviewers find this revision satisfactory.

Section #1: The entire editorial decision letter from Editor-in-Chief

The review of your paper, "MT-STNet: A Novel Multi-Task Spatiotemporal Network for Highway Traffic Flow Prediction," T-ITS-22-05-1162, has been completed. The reviewers' comments and those of the Associate Editor are copied below. Based on these comments and recommendation of the Associate Editor your paper is not ready for publication in its present form, A properly revised version that takes care of the concerns and drawbacks pointed out by the reviewers and Associate Editor is potentially publishable.

Therefore, I suggest that you revise your paper along the lines described by the reviewers and resubmit the paper. Please include a description on how you took into account the reviewers' comments in preparing your revision. We would then hope to determine a publication decision soon thereafter. Please note that if your revision is not submitted within the next 3 months (90 days), your paper will be treated as a new paper.

You may want to resubmit your paper as a REGULAR PAPER (suggested length: 10 Transactions pages, authors' biographies included). The Associate Editor will determine whether your paper is best suited to be considered as a Regular Paper or shortened to a Short Paper.

Thank you for submitting your manuscript to the Transactions on Intelligent Transportation Systems.

Response: Thank you for your suggestion. We have meticulously revised the original manuscript to address the concerns and drawbacks highlighted by the reviewers. Specifically, we have made revisions to our paper based on the suggestions provided by the reviewers (which are highlighted in yellow in the resubmitted manuscript). Furthermore, we have included a detailed description in Section #2 that outlines how we took into account the reviewers' comments in preparing the revised version of our paper. We appreciate the invaluable feedback provided by the reviewers and Associate Editor.

Section #2: The entire editorial decision letter from reviewers 1, 2 and 3

Reviewer: 1

Recommendation: **Prepare A Major Revision For A New Review**

Comments:

In this paper, the author proposed a multi-task spatiotemporal network for highway traffic flow prediction, named MT-STNet. The composition of this model is novel, and the current manuscript is well organized. However, the authors have to consider the following remarks.

(1) The authors claimed that this model focused on long-term prediction. However, according to the description in Section IV-C, the prediction horizon is set as 6. In previous studies on long-term prediction, this value is always assigned as 50 steps [1], 24 hours [2], etc. Therefore, the prediction mode in this manuscript cannot really be regarded as long-term prediction. I suggest the author conduct a more detailed literature review and remove such statements.

Reference:

[1] Wang, Z., Su, X., & Ding, Z. (2020). Long-term traffic prediction based on lstm encoder-decoder architecture. IEEE Transactions on Intelligent Transportation Systems, 22(10), 6561-6571.

[2] Peng, H., Du, B., Liu, M., Liu, M., Ji, S., Wang, S., ... & He, L. (2021). Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning. Information Sciences, 578, 401-416.

Response: Thank you so much for your professional suggestion. The authors have read the references the reviewer provided and rethought and discussed the definition of traffic prediction. Dear reviewer, we have extended the prediction horizons from six-time steps to twelve in this paper, consistent with classic methods in traffic flow forecasting, such as Graph WaveNet [1] and GMAN [2]. In addition, the authors deem the reviewer gives us a new insight into how to define 'long-term' in the traffic prediction tasks; the task proposed in this paper is then called 'multi-step traffic prediction.', which avoids the definition falling into disputation, helping the audience to read. Moreover, thanks again to the reviewer for providing this crucial suggestion, and we have replaced all 'long-term' in the article with 'multi-step.'

[1] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019, August). Graph wavenet for deep spatial-temporal graph modeling. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (pp. 1907-1913).

[2] Zheng, C., Fan, X., Wang, C., & Qi, J. (2020, April). Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 1234-1241).

(2) GNN-based models in the baselines are always proposed on the benchmark of METR-LA or PEMS-BAY. Hence, these models always utilize traffic states of the last 12 steps as input and predict future traffic flow of the following 12 steps. These model may be good at "long-term" prediction such as more than 6 steps. So, the current prediction comparison (i.e. 6-step prediction) is unfair for these baselines, especially under the statement of long-term prediction. What are the prediction performance of MT-STNet and these baselines more than 6 steps?

Response: Thanks for your insightful suggestion. We have extended the prediction steps from six to twelve horizons, according to the reviewer's suggestion. In addition, some popular traffic prediction models in recent years, such as Graph WaveNet, MTGNN, ST-GRAT, RGSL, etc., have been added to the baselines. Moreover, we have updated the experimental results of the proposed method and baselines and added them to Fig. 8 and Tables II, III, IV, and Vof the manuscript. Note that more detailed experimental results have been added to the manuscript, including average error for the next twelve horizon forecasting and what gains for each time step. Furthermore, we added more comparisons from new baselines and the proposed method and summarized new findings from them. Therefore, these comparisons and analyses have been updated in Sec. V. Thanks again to the reviewer for providing this professional suggestion.

Table R1 The prediction results of the MT-STNET and baselines on the entrance toll task

Model	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SARIMA	5.941	9.214	60.559%	5.982	9.432	64.337%	5.892	9.228	62.815%
SVR	4.214	6.522	39.194%	4.609	7.113	42.947%	5.412	8.393	51.237%
LSTM_BILSTM	4.351	6.763	44.938%	4.899	7.724	49.670%	6.142	10.046	58.980%
DELA	4.101	6.253	41.926%	4.449	6.827	46.651%	5.229	8.184	59.904%
T-GCN	4.208	6.461	43.834%	4.293	6.647	42.554%	4.675	7.404	45.111%
STGNN	4.083	6.334	37.417%	4.435	6.937	40.515%	5.157	8.271	45.855%
DCRNN	3.815	5.867	33.469%	3.914	6.074	34.234%	4.179	6.650	35.667%
AGCRN	3.804	5.883	34.451%	3.910	6.084	35.492%	4.114	6.489	37.241%
ASTGCN	4.010	6.180	35.861%	4.275	6.578	43.248%	4.939	7.770	49.410%
MSTGCN	4.059	6.217	40.705%	4.415	6.799	44.060%	5.188	8.199	50.652%
Graph-WaveNet	3.836	5.893	35.720%	4.009	6.221	37.979%	4.160	6.515	38.841%
GMAN	3.883	5.914	41.163%	3.941	6.014	41.401%	4.060	6.236	42.677%
ST-GRAT	3.980	6.141	33.529%	4.342	7.017	35.170%	5.478	9.390	47.838%
MTGNN	3.827	5.924	34.029%	3.977	6.193	34.865%	4.321	6.832	38.513%
RGSL	3.740	5.938	38.627%	3.852	6.129	39.687%	4.125	6.699	40.560%
MT-STNet (ours)	**3.756**	**5.801**	**35.436%**	**3.807**	**5.885**	**35.520%**	**3.951**	**6.114**	**36.341%**

Table R2 The prediction results of the MT-STNet and baselines on the exit toll task

Model	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SARIMA	5.886	8.974	53.816%	6.025	9.705	55.907%	6.588	12.056	62.007%
SVR	4.118	6.496	36.771%	4.558	7.246	40.155%	5.473	8.803	48.131%
LSTM_BILSTM	4.225	6.649	42.442%	4.838	7.783	47.105%	6.218	10.127	57.198%
DELA	4.007	6.160	40.304%	4.395	6.841	45.175%	5.335	8.287	59.809%
T-GCN	3.948	6.196	38.519%	4.076	6.442	37.411%	4.490	7.178	39.779%
STGNN	3.911	6.159	35.291%	4.293	6.912	38.470%	5.138	8.527	44.911%
DCRNN	3.548	5.493	30.741%	3.728	5.852	31.172%	3.996	6.387	32.157%
AGCRN	3.599	5.611	32.873%	3.755	5.963	33.525%	3.945	6.305	34.825%
ASTGCN	3.790	5.936	33.040%	4.049	6.367	38.639%	4.696	7.528	42.901%
MSTGCN	3.854	5.945	38.565%	4.239	6.622	41.942%	5.091	8.181	49.853%
Graph-WaveNet	3.588	5.586	32.219%	3.714	5.861	33.498%	3.920	6.303	35.042%
GMAN	3.730	5.768	37.797%	3.812	5.949	38.262%	3.974	6.259	39.942%
ST-GRAT	3.709	5.686	31.505%	4.104	6.583	32.018%	5.092	8.325	44.342%
MTGNN	3.597	5.641	32.244%	3.799	6.043	32.822%	4.130	6.670	35.738%
RGSL	3.631	5.856	34.811%	3.739	6.087	35.504%	4.070	6.852	36.477%
MT-STNet (ours)	**3.685**	**5.853**	**32.075%**	**3.729**	**5.948**	**32.249%**	**3.859**	**6.155**	**33.340%**

Table R3 The prediction results of the MT-STNet and baselines on the gantry task

Model	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SARIMA	9.392	18.089	55.416%	9.672	18.855	56.380%	10.431	20.454	58.952%
SVR	5.774	8.480	30.719%	6.445	9.602	34.574%	8.008	12.153	44.858%
LSTM_BILSTM	5.918	8.837	32.796%	6.728	10.297	36.322%	8.822	14.017	44.705%
DELA	5.460	8.107	29.135%	5.853	8.903	30.878%	6.806	10.861	36.005%
T-GCN	5.651	8.341	30.439%	5.870	8.782	30.214%	6.571	10.155	32.445%
STGNN	5.600	8.471	28.324%	6.175	9.589	30.462%	7.486	12.026	34.977%
DCRNN	5.050	7.423	27.087%	5.372	7.997	28.037%	5.817	8.933	28.117%
AGCRN	5.218	7.712	27.075%	5.397	8.049	28.194%	5.730	8.730	29.373%
ASTGCN	5.386	8.017	27.724%	5.779	8.733	31.181%	6.775	10.587	35.537%
MSTGCN	5.419	8.007	29.794%	5.969	8.969	32.631%	7.257	11.199	40.257%
Graph-WaveNet	5.161	7.655	26.544%	5.367	8.071	27.731%	5.733	8.813	28.859%
GMAN	5.178	7.554	29.078%	5.281	7.746	29.511%	5.512	8.175	30.763%
ST-GRAT	5.378	7.931	25.471%	6.111	9.457	28.087%	7.843	13.029	38.446%
MTGNN	5.206	7.698	26.929%	5.502	8.226	27.810%	6.063	9.370	30.527%
RGSL	5.329	8.007	28.138%	5.536	8.490	28.805%	6.081	9.764	29.911%
MT-STNet (ours)	**5.128**	**7.531**	**26.698%**	5.206	7.651	**26.941%**	**5.406**	**7.962**	**27.701%**

Table R4 The prediction results of the MT-STNet and baselines on the whole dataset

Model	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SARIMA	7.962	15.058	56.182%	8.159	15.704	57.975%	8.704	17.128	60.379%
SVR	5.166	7.794	33.463%	5.741	8.765	37.210%	7.039	10.947	46.680%
LSTM_BILSTM	5.302	8.098	36.911%	6.025	9.417	40.880%	7.824	12.676	49.761%
DELA	4.929	7.446	33.661%	5.312	8.183	36.559%	6.230	9.951	45.018%
T-GCN	5.056	7.643	34.494%	5.233	8.010	33.902%	5.820	9.179	36.221%

STGNN	4.994	7.706	31.358%	5.491	8.678	33.874%	6.602	10.800	38.909%
DCRNN	4.533	6.818	28.982%	4.786	7.295	29.799%	5.163	8.106	30.305%
AGCRN	4.644	7.034	29.564%	4.805	7.349	30.579%	5.087	7.929	31.889%
ASTGCN	4.824	7.339	30.265%	5.168	7.955	34.868%	6.035	9.583	39.546%
MSTGCN	4.866	7.339	33.512%	5.348	8.189	36.549%	6.457	10.167	44.034%
Graph-WaveNet	4.613	6.994	29.349%	4.798	7.370	30.755%	5.093	7.990	31.911%
GMAN	4.660	6.956	33.007%	4.750	7.131	33.409%	4.947	7.505	34.746%
ST-GRAT	4.798	7.236	28.133%	5.398	8.551	30.166%	6.876	11.634	41.333%
MTGNN	4.641	7.036	29.274%	4.892	7.499	30.089%	5.369	8.476	33.019%
RGSL	4.681	7.250	31.379%	4.850	7.641	32.125%	5.299	8.710	33.161%
MT-STNet (ours)	4.596	6.936	29.363%	4.663	7.046	29.563%	4.839	7.325	30.397%

(3) In Table 4, please add more computational cost comparisons between MT-STNet and other advanced baselines.

Response: Thanks to the reviewer's suggestion. We have provided detailed total parameters, time cost in the training and inference phases, and GPU memory usage in the training and test stages, as shown in Table R5 (added in Table VI of the manuscript). Moreover, more detailed comparisons and conclusions are added in the Sec. V *Computation Cost*. For example, "we prefer a faster, more efficient, low-complexity model that uses less GPU memory while maintaining accurate prediction. Therefore, MT-STNet is proposed as having superior performance, and its model complexity is less than that of DCRNN, AGCRN, Graph-WaveNet, GMAN, MTGNN, and RGSL. In the training phase, because of the difference in data loading, the GPU memory usage of MT-STNet is higher than that of the optimal baseline, DCRNN. In contrast, in the inference phase, the GPU memory usage is minimal, and the time cost outperforms the top two optimal baselines, DCRNN and GMAN, as shown in Table VI. MT-STNet provides multi-step forecasts in a single pass, reducing the time required for inference compared with DCRNN and GMAN. The computation cost further validates the superiority of MT-STNet in multi-step highway traffic flow prediction."

Table R5 Computation cost during the training and inference phases (* means the model train one time on the whole training set)

Model	Parameters	Training / (100 iterations) (batch size =128)		Inference (batch size =1)	
		Time Cost	GPU Memory Usage	Time Cost	GPU Memory Usage
SARIMA*	-	39,612.092 (min)	-	5,507.866 (min)	-
SVR*	-	571.626 (min)	-	108.240 (min)	-
LSTM_BILSTM	1,121,089	0.810 (min)	8691MiB	3.022 (min)	531MiB
DELA	120,423	0.180 (min)	7277MiB	0.601 (min)	1965MiB
T-GCN	37,844	0.064 (min)	1523MiB	0.305 (min)	501MiB
STGNN	617,985	0.165 (min)	2827MiB	1.179 (min)	1619MiB
DCRNN	372,353	2.144 (min)	4153MiB	13.299 (min)	1615MiB
AGCRN	750,240	0.152 (min)	3105MiB	2.108 (min)	2035MiB
ASTGCN	74,312	0.348 (min)	2479MiB	0.701 (min)	1789MiB
MSTGCN	50,956	0.327 (min)	2241MiB	0.596 (min)	1787MiB
Graph-WaveNet	306,580	0.124 (min)	2869MiB	0.647 (min)	1759MiB
GMAN	916,801	1.359 (min)	16899MiB	2.205 (min)	531MiB
ST-GRAT	2,238,849	0.900 (min)	17789MiB	15.033 (min)	1639MiB
MTGNN	204,668	0.121 (min)	2801MiB	0.600 (min)	1727MiB
RGSL	871,312	0.347 (min)	3945MiB	4.407 (min)	1681MiB
MT-STNet	192,771	0.450 (min)	8707MiB	2.405 (min)	523MiB

(4) In the direct distance matrix, the authors set the distance between unconnected nodes as -1. However, in network science, distances between unconnected nodes are always assigned as infinity. Please explain the reason for this setting.

Response: Thanks to the reviewer's expert suggestion. The physical structure information is one of the contributions to our work, and the distance is a factor we must consider. In this work, we use positive and negative values to classify the direct distance between nodes in the first version, which helps the computer or program understand the distance difference. As the reviewer mentioned, existing studies assign distances between unconnected nodes as infinity. To follow the network science, we change the distance metrics proposed in this paper, and then, if there is no connection between nodes v_i and v, the direct distance $s_{v_i,v}$ is infinity. Therefore, the direct distance between nodes in the experiment is then reconstructed, and the new experimental results are given in Tables II, III, IV, and V of the manuscript (i.e., Tables R1, R2, R3, and R4 in this response). We have modified the distance description in Sec. IV, and uploaded the results to Sec. V. Finally, thanks again to the reviewer for providing this crucial suggestion.

(5) In Eq. (17), please introduce more details about how to choose the value of λ.

Response: Thanks for your valuable suggestion. The value of λ represents a relevance that can be obtained by the inner product of the query vector of node v at time step t_{P+j} and the key vector of node v at time step t, which computes automatically according to the input variables, i.e., query and key vectors. Especially the first variable is the query vector obtained by using a nonlinear transformation function f_q transfer the source input historical spatiotemporal correlation to the target vector, which can reference the Transformer [1]; similar to the query vector, the second variable (key vector) is then obtained through the nonlinear transformation function f_k converts the future representation. The above description has been added in the generative inference Section.

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

(6) The current figures in Section IV need to improve. All the font sizes are too small, making it difficult to read.

Response: Thanks for your professional suggestion. We have modified the font size in the figures according to the reviewer’s suggestion.

Reviewer: 2

Recommendation: **Accept With Minor Changes**

Comments:

(There are no comments. Please check to see if comments were included as a file attachment with this e-mail or as an attachment in your Author Center.)

What are the contributions of this paper?: The article proposes a novel multi-task spatiotemporal network (MT-STNet) for highway traffic flow prediction. The proposed model addresses three critical issues: 1) encoding spatiotemporal correlation and physical structure information of the traffic network; 2) eliminating the problem of long-term prediction error propagation; and 3) exploiting the benefits of multi-task learning for prediction. The experimental results demonstrate that the proposed model outperforms the baseline model. However, several relevant expressions can be further improved to enhance the clarity and comprehensiveness of the article.

Response: Thanks to the reviewer's expert suggestion. The authors have rechecked the entire work and improved the expressions. For example, most classical baselines and new relative works are added to the related work, and more details are updated. In addition, we added more experiments to Sec. V, and more comparisons and the contribution of each component are added. Moreover, all improved contexts are highlighted in yellow. Thanks again to the reviewer for providing this crucial suggestion.

What are some ways in which the paper could be improved? Please supply any additional important references that you feel the author omitted which should be noted in the paper.: These improvements are as follows:

(1) The Embedding Layer section should provide more information about how multi-dimensional timestamps are mapped and embedded. While Fig. 3 reflects the process, the specific details of the mapping and embedding process require clarification.

Response: Thanks to the reviewer's expert suggestion. Minute of day and day of week embeddings mean dense representations for the integer type data, as shown in Fig. R1. Besides, there is road station embedding, which is used to present the station index. In this paper, the integer variable is first transferred into one-hot codes and then mapped into a dense matrix, a float-dense vector; different types of variables meet the same values, such as timestamp 2021.08.06 06:00 and road station index 6, which leads the model impossibly pick out the difference between these variables. This problem is solved by the embedding method, which is proven in the existing works and widely used in traffic prediction systems. For instance, the embedding method has been applied in our recently published paper [1], and we have cited this reference in Sec. IV. As shown in Fig. R1 (Left), the embedding process can be divided into three steps: first, allocating the index to each variable. For example, 7 days a week, we give each day an index from 0 to 7; second, we transfer the index into a one-hot code. For instance, day 2 ($x \in R$) is transferred into a one-hot vector ($x' \in R^{7\times1}$), the value of the corresponding index in the one-hot is set to one, and vice versa; then map the one-hot to dense embedding matrices ($\mathrm{M} \in R^{7\times64}$) via multiplication (${x'}^{\intercal}\mathrm{M} \in R^{1\times64}$), and the matrices are updated adaptively in the training phase. In addition, Fig. R1 has been added in Sec. IV. In the existing studies, the time information and station index are usually used as essential internal factors input to the model for traffic prediction [2]. Similar to timestamp embedding, two types of real-valued embeddings can represent node degrees and directed edges between nodes, and the embedding process does not repeat description because the process details are the same. Thanks a lot to the reviewer for providing this significant suggestion, which helped us improve the entire quality of this paper.

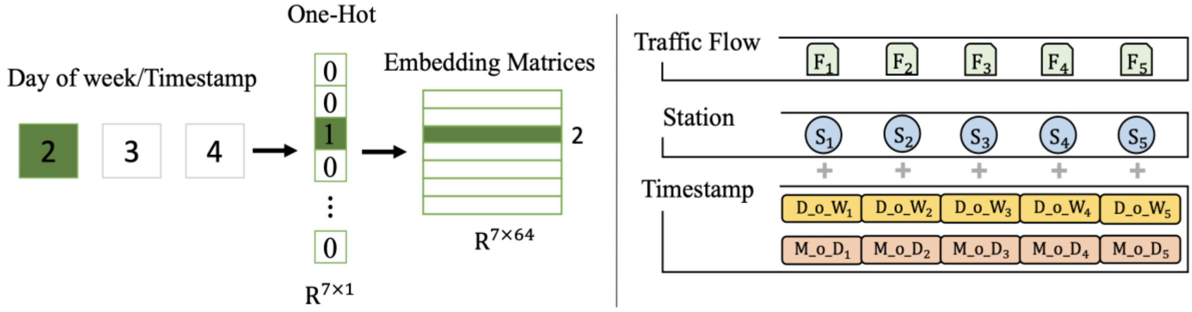

Fig. R1. Left: Example of the day of week embedding. Right: The node representation consists of three types of primary data information: traffic flow embedding, station embedding, and timestamp embedding (including day of week and minute of day, i.e., D_o_W and M_o_D).

[1] Zou, G., Lai, Z., Ma, C., Tu, M., Fan, J., & Li, Y. (2023). When Will We Arrive? A Novel Multi-Task Spatio-Temporal Attention Network Based on Individual Preference for Estimating Travel Time. IEEE Transactions on Intelligent Transportation Systems.

[2] Zheng, C., Fan, X., Wang, C., & Qi, J. (2020, April). Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 1234-1241).

(2) The article needs to provide more details about the definition and value of subtasks. This includes whether there are more important main tasks, whether there is a correlation between subtasks, and how the performance of multiple models compares to single-task models.

Response: Thanks for your professional suggestion. Multi-task learning (MTL) is a learning paradigm in machine learning, and its aim is to leverage useful information contained in multiple related tasks to help improve the generalization performance of all the tasks [1]. (1) The target prediction task is highway traffic flow prediction, divided into several subtasks because we want to distinguish the flow heterogeneity, but our target is unchanged. Therefore, we deemed that the subtasks are equal and do not prefer one of them. For the loss function, we distribute the same weights to each subtask. In future work, we may add a relative task to our work, such as traffic speed prediction, which needs to add different loss weights to both tasks because flow forecasting is our target main task. (2) The highway network is a closed-loop graph, which means the total traffic flow equals the flow from entry tolls and subtracts the flow that leaves exit tolls. However, for the highway network, there is tremendous traffic flow heterogeneity. Three root causes: (1) the difference in speed limitation between ramps and main roads; (2) trip purpose; and (3) the types of roads, especially ramps and main roads. Therefore, highway traffic flow forecasting is divided into three categories: traffic flow passing the entrance toll, gantry, and exit toll, regarded as multi-task learning. The total traffic flow passing entrance and exit tolls will affect the flow passing the gantries of the main roads; the total traffic flow passing entrance exit tolls and gantries indirectly affect the flow passing the entrance tolls because main road traffic conditions limit the flow of entrance tolls; and the total traffic flow passing entrance tolls and gantries directly affects how many vehicles could be passing exit tolls. The correlation between subtasks is actually affecting each other. (3) In the Sec. V *Influence of Each Component*, we remove the multi-task learning and use the single-task model applied in these three tasks, and the performance is given in Table VII of the manuscript. From Table VII, we can see that ‘The accuracy is weakened by discarding traffic heterogeneity in the highway system. For instance, compared with MT-STNet, the variant without multi-task learning increased MAE, RMSE, and MAPE by 0.472%, 0.096%, and 1.046%, respectively, for horizon six; 0.320%, 0.169%, and 5.421% for the next twelve time steps. The experiments verify that multi-task learning is an appropriate technique for handling flow heterogeneity in traffic networks, which divides traffic prediction into several relative subtasks, sharing underlying knowledge.' Thanks again to the reviewer for providing this crucial suggestion.

[1] Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering, 34(12), 5586-5609.

(3) Fig. 7 shows a generalized study area that does not provide sufficient spatial location and interrelationship of 66 monitors. The article should provide more information about how the physical network structure between monitors is acquired and learned.

Response: Thanks for your insightful suggestion. (1) Dear reviewer, we have uploaded the spatial location and the connectivity between stations. For the station, we provided detailed information, including longitude, latitude, and distance between two connected stations (as shown in Fig. R2), such as station G000664001001610010, the Chinese name is "京藏望远枢纽至永宁上行门架," as shown in Fig. R3 left. In addition, all of the station information has been uploaded to the personal GitHub page (https://github.com/zouguojian/Traffic-flow-prediction/tree/main/MT-STNet/data/). The station is then mapped into an index (as shown in Fig. R3 right) orderly for the embedding generation, as mentioned in Sec. IV.

Start Station	起点桩号	Longitude	Latitude	End Station	终点桩号	Longitude	Latitude	Distance (m)
四十里店收费站	K1160+260	106.3434259	38.65589963	四十里店至京藏四十里店枢纽	K1160+130	106.346548	38.648498	866.53
四十里店至京藏四十里店枢纽	K1160+130	106.346548	38.648498	四十里店枢纽		106.351866	38.64227	832.39
四十里店枢纽		106.351866	38.64227	四十里店枢纽到银川北枢纽门架	K1163+186	106.336273	38.622721	2561.09
四十里店南枢纽到银川北枢纽门架	K1163+186	106.336273	38.622721	银川北枢纽	K58+564	106.3316	38.578268	4959.59
银川北枢纽	K58+564	106.3316	38.578268	银川北主线收费站	K54+800	106.2829845	38.57812899	4226.05
银川北枢纽	K58+564	106.3316	38.578268	银川北匝道收费站（贺兰收费站）	K1168+200	106.3331651	38.56864417	1078.74
银川北枢纽	K58+564	106.3316	38.578268	银川北枢纽到贺兰山路收费站门架	K1175+722	106.3369005	38.51238957	7339.83
银川北枢纽到贺兰山路收费站门架	K1175+722	106.3369005	38.51238957	贺兰山路收费站	K1177+100	106.3317006	38.50169292	1272.55
贺兰山路收费站	K1177+100	106.3317006	38.50169292	贺兰山路收费站到银川东枢纽门架	K1178+836	106.3350869	38.48446767	1937.9
贺兰山路收费站到银川东枢纽门架	K1178+836	106.3350869	38.48446767	银川东枢纽	K1510+226	106.332031	38.431496	5896.19
银川东枢纽	K1510+226	106.332031	38.431496	银川东收费站	K1511+500	106.3146453	38.43600761	1595.31
银川东枢纽	K1510+226	106.332031	38.431496	银川东枢纽到白鸽枢纽门架	K1187+780	106.3105628	38.41086277	2959.85
银川东枢纽	K1510+226	106.332031	38.431496	银川东枢纽到掌政枢纽门架	K1508+700	106.3360068	38.41453518	1917.49
银川东枢纽到白鸽枢纽门架	K1187+780	106.3105628	38.41086277	白鸽枢纽	K2392+237	106.303789	38.403961	968.14
白鸽枢纽	K2392+237	106.303789	38.403961	银川南收费站	K1189+700	106.2908391	38.4010491	1173.96
白鸽枢纽	K2392+237	106.303789	38.403961	白鸽枢纽到望远枢纽门架	K1191+100	106.288999	38.3862921	2349.67
白鸽枢纽到望远枢纽门架	K1191+100	106.288999	38.3862921	望远枢纽		106.276588	38.33799	5478.8
望远枢纽		106.276588	38.33799	望远枢纽到永宁收费站门架	K1201+370	106.2373622	38.30641865	4901.9
望远枢纽到永宁收费站门架	K1201+370	106.2373622	38.30641865	永宁收费站	K1206+850	106.2357992	38.29375097	1415.17
望远枢纽		106.276588	38.33799	望远枢纽到掌政枢纽门架	K1199+500	106.3391155	38.38970528	7925.19
望远枢纽到掌政枢纽门架	K1199+500	106.3391155	38.38970528	掌政枢纽	K1190+033	106.355675	38.409192	2603.48

Fig. R2. Detailed station information in the highway network.

Station	Chinese Name
G008564001000310010	机场北至机场南上行门架
G008564001000320010	机场南至机场北下行门架
G008564001000210010	银昆石坝枢纽至机场北上行门架
G008564001000220010	机场北至银昆石坝枢纽下行门架
G002064001000410010	青银石坝枢纽至水洞沟上行门架
G002064001000420010	水洞沟至青银石坝枢纽下行门架
G002064001000320010	青银石坝枢纽至青银掌政枢纽下行门架
G002064001000310010	青银掌政枢纽至青银石坝枢纽上行门架
G002064001000210010	青银银川东枢纽至青银掌政枢纽上行门架
G002064001000220010	青银掌政枢纽至青银银川东枢纽下行门架
G000664001001610010	京藏望远枢纽至永宁上行门架

Station	Index
G008564001000310010	26
G008564001000320010	27
G008564001000210010	28
G008564001000220010	29
G002064001000410010	30
G002064001000420010	31
G002064001000320010	32
G002064001000310010	33
G002064001000210010	34
G002064001000220010	35
G000664001001610010	36

Fig. R3. The relationship among Station, Chinese Name, and Index.

Moreover, the interrelationship of 66 monitors (i.e., N = 66). The connection is given in the adjacent matrices $A \in R^{N \times N}$ on the GitHub page (https://github.com/zouguojian/Traffic-flow-prediction/blob/main/MT-STNet/data/adjacent_fully.csv), as shown in Fig. R4 left. More precisely, when A_{v_i,v_j} is 'one,' indicating a directed connection edge between node v_i and node v_j (and vice versa), while 'zero' indicates no directed connection. For example, the node pairs (0, 60) and (0, 55) have connections, and the direction is from 0 to 60 and 0 to 55. The distance value of each pair has been given in Fig. R2, and if there is no connection between nodes v_i and v, the direct distance is infinity.

The highway is a connected topological network in the physical space, as shown in Fig. R4 right (i.e., it is the source map, and the blue direction arrow is manually added between stations compared with Fig. 7 in the manuscript.). We map the highway network in the physical space to the logical space that the computer can understand. The mapping work can be interpreted as abstracting the intelligent sensors of electronic toll collection (ETC) stations and gantries as nodes in the graph and abstracting road segments as edges connecting nodes, as shown in Fig. 1 in the manuscript. Note that Fig. 1 in the manuscript is used instead of Fig. R4 right because it is simple and easy to understand for audiences. The above details are used to illustrate the interrelationship of 66 monitors and the contents have been described in Sec. III and V.

src_FID	nbr_FID
0	60
0	55
1	37
1	59
2	51
2	44
3	53
3	50
3	17
4	16
4	50
4	53
5	48
5	57
6	46
6	49
7	45
7	42
7	34
8	43
8	38
9	32
9	28
9	30
10	31
10	65
11	26
11	29
12	62

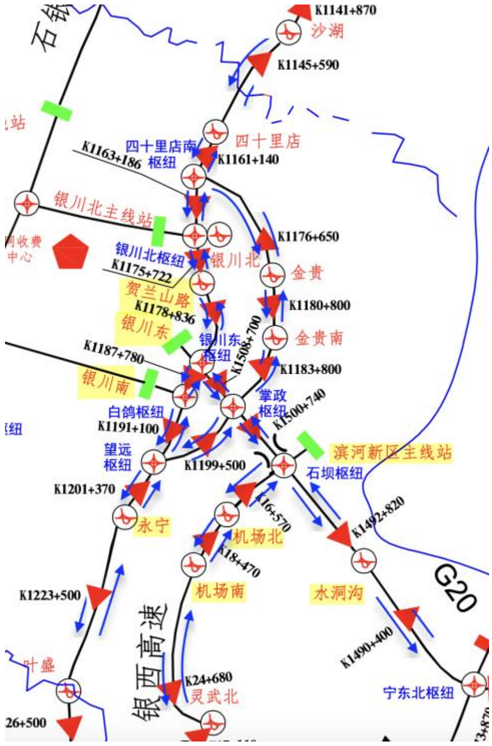

Fig. R4. The connectivity between monitoring stations.

(2) The physical network structure between monitors is provided above, such as station location, abstracting process, connectivity between stations, and distance between stations. In addition, the node degree and the shortest path between stations can be calculated via statistics and the Dijkstra algorithm according to the connectivity and distance information. Therefore, an ST-Physical Block is designed to model the spatio-temporal correlations of the highway network according to the physical network structure. For instance, the graph convolutional network (GCN) is used, which utilizes equations (12) and (13) in the manuscript to model the static spatial correlation according to the inputs of connectivity matric and traffic flow. As described in the manuscript, the traffic flow of a monitoring station in the traffic network is affected by other monitoring stations, and the influence weight changes dynamically with time. This property is defined as dynamic spatial correlation, and a spatial attention approach is developed to adaptively model the correlations between different stations of the traffic network. Additionally, considering the positive effects of the inherent physical information of the traffic network on the calculation of dynamic spatial correlations, including station in- and out-degrees, shortest path, and shortest path distance. The above paragraphs provide most information about how the physical network structure between monitors is acquired and learned.

Note that considering the page limitation of the manuscript, the authors have **uploaded all response details of these questions** to the GitHub page to help our audiences and reviewers know the whole physical network structure, including what information is required and learned from resources. Furthermore, all essential files, such as distance and connectivity, are added to GitHub. Thanks again to the reviewer for providing this crucial suggestion.

Reviewer: 3

Recommendation: **Prepare A Major Revision For A New Review**

Comments:

(1) If the standard terminology from traffic condition forecasting literature and practice, the authors are presenting a short-term forecasting method, not a long-term method. Because ARIMA is a parametric model, the authors must provide details on model form, fitted parameter values, and parameter standard error values. From the limited information that is given, it is clear that the authors are not using a proper ARIMA model form. The Van Der Voort et al. (1996) paper they cite presented a hybrid approach where a trained-Kohonen map classifier was used to classify short-duration time series segments into one of several supposedly stationary categories with forecasts provided by a separate ARIMA model that had been fitted to the various time series subcategories. Therefore, this paper is not a sufficient reference for proper ARIMA modeling. Also, another paper in the authors' reference list, Okutani and Stephanedes (1984) contains a well-founded assertion that renders the complex attempt to deal with non-stationarity by using a classifier to subset the time series into stationary snippets unnecessary. Okutani and Stephandes state that a first weekly difference is sufficient to derive a stationary transformation of the raw time series. The implication of this is that the appropriate ARIMA form for traffic condition data series (whether flow, speed, density or any other traffic stream metric) is Seasonal ARIMA (SARIMA) with a one week seasonal period. The authors will find the necessary information to understand and apply SARIMA in Williams and Hoel (2003) for which the full citation is provided.

Response: Thanks for your insightful suggestion. (1) In previous studies, such as Graph WaveNet [1], GMAN [2], and ST-GRAT [3], the twelve-time steps forecasting is defined as 'long-term prediction.' In addition, the authors deem the reviewer gives us a new sight into how to define 'long-term' in the traffic prediction tasks; the task proposed in this paper is then called 'multi-step traffic prediction.', which avoids the definition falling into disputation, helping the audience to read. Moreover, thanks again to the reviewer for providing this crucial suggestion, and we have replaced all 'long-term' in the article with 'multi-step.' (2) As the reviewer mentioned that ARIMA is a parametric model, the details on model form, fitted parameter values, and parameter standard error values are then provided in this response. In addition, we have given the parameters optimization process of ARIMA and SARIMA, including training and test codes. Especially, authors and audiences could reproduce the codes in their personal device and other datasets we provided. (3) We used traditional ARIMA to predict traffic flow for each monitoring station but neglected the seasonal property, and it is unscientific compared with SARIMA. After consulting relative references according to the reviewer's proposal, a more proper ARIMA named SARIMA is more adaptive as a baseline for traffic flow than traditional ARIMA. Therefore, we used SARIMA as one of the baselines to predict traffic flow, and the overall process refers to the professional tutorial (i.e., the link is https://zhuanlan.zhihu.com/p/531987920). For the highway traffic flow prediction in this paper, there are seven hyperparameters that need to be confirmed in the training phase: the (p, d, q) order of the model for the number of AR parameters, differences, and MA parameters; and the (P, D, Q, s) order of the seasonal component of the model for the AR parameters, differences, MA parameters, and periodicity. The detailed process is given in the following steps (in the GitHub web, we are the first to provide the detailed program of ARIMA and SARIMA for traffic flow forecasting),

[1] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019, August). Graph wavenet for deep spatial-temporal graph modeling. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (pp. 1907-1913).
[2] Zheng, C., Fan, X., Wang, C., & Qi, J. (2020, April). Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 1234-1241).
[3] Park, C., Lee, C., Bahng, H., Tae, Y., Jin, S., Kim, K., ... & Choo, J. (2020, October). ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In Proceedings of the 29th ACM international conference on information & knowledge management (pp. 1215-1224).

Step 1. We read the source data from the database (train.csv). For example, we randomly choose a monitoring station, such as station 43, and visualize one-week historical values on the plots, as shown in Visualization I.

Visualization I

```
# -- coding: utf-8 --

import pandas as pd
import numpy as np
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_squared_error
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from statsmodels.tsa.stattools import adfuller
from tqdm import tqdm_notebook
from itertools import product
import warnings
import datetime
from statsmodels.tsa.arima.model import ARIMA
from pmdarima.model_selection import train_test_split

def plot(ts):
    results = adfuller(ts)
    results_str = 'ADF test, p-value is: {}'.format(results[1])

    grid = plt.GridSpec(2, 2)
    ax1 = plt.subplot(grid[0, :])
    ax2 = plt.subplot(grid[1, 0])
    ax3 = plt.subplot(grid[1, 1])

    ax1.plot(ts)
    ax1.set_title(results_str)
    plot_acf(ts, lags=int(len(ts) / 2 - 1), ax=ax2)
    plot_pacf(ts, lags=int(len(ts) / 2 - 1), ax=ax3)
    plt.show()

# 1. Read the original data from the database and observe the results of 
the visualization
data = pd.read_csv("train.csv")
filtered_data = data[data.iloc[:, 0] == 43]
y = filtered_data.iloc[:, 5]

# 2. visualization
print('Visual display, including stationarity test, autocorrelation, and 
partial autocorrelation plots')
# If p is less than 0.05, why is no difference needed?
# It means that the data is relatively stable, but the seasonal difference 
is needed.
plot(y.values[:2016])
```

We can download these codes and datasets from GitHub and try them on your local computer, and the results are given in Fig. R5.

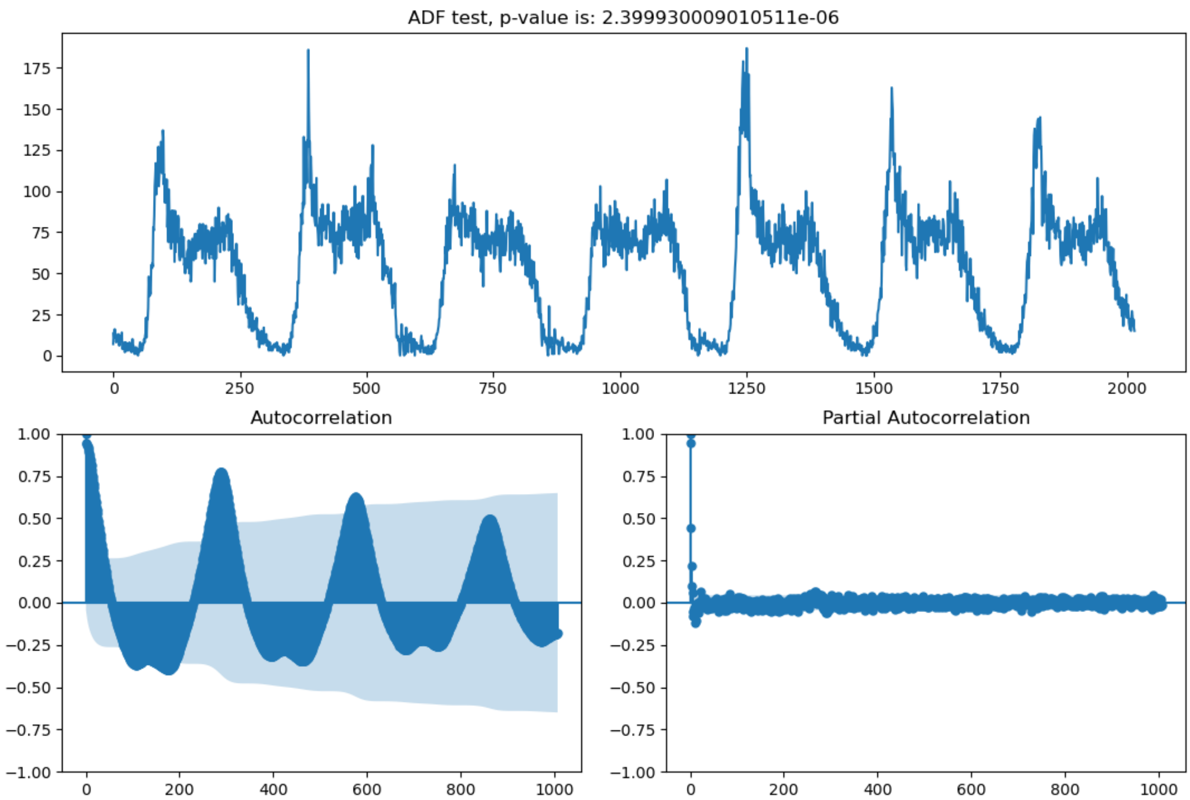

Fig. R5 Results of ADF, autocorrelation, and partial autocorrelation in example samples.

The ADF test is stable (the p-value is approximately equal to 0 and much less than 0.05). Indeed, there is no upward or downward trend overall; the mean remains unchanged, and the variance is also relatively stable, with $d = 0$. However, we can see from Fig. R5 that there is apparent periodicity, and the periodicity is $s = 12 * 24$. If ARIMA is used, performing a step of seasonal difference is still necessary.

Step 2. In addition, to find the optimal ARIMA, the AR parameters (p) and parameters MA (q) values are optimized via the find_pd(ts, d, max_p, max_q) function. The find_pd function tries different possible parameter values in ARIMA to find the optimal performance on training datasets. The whole program ARIMA is given, and readers and audiences can download it to local personal devices for training and testing; the codes are shown in the following,

ARIMA II

```
from pmdarima.model_selection import train_test_split
def version_arima_with_manual(ts):
    """
    ARIMA
    """
    # period
  periods = 12 * 24
    # seasonal difference
  ts_diff = ts - ts.shift(periods)

    # data splitting
  train, test = train_test_split(ts_diff, train_size=0.8)
    test_list = test.values.tolist()
    label_list = ts.values.tolist()[train.shape[0]:]
    print(train.shape, test.shape)

    # model training
  p, q, _ = find_pq(train)
```

```
    model = ARIMA(train.tolist(), order=(p, 0, q)).fit()
    print(model.summary())

    # model predicting
    history = train.values.tolist()
    pre_s = []
    label_s = []
    total = 0
    for t in range(0, test.shape[0]-12, 12):
        try:
            start_time = datetime.datetime.now()
            yhat = model.forecast(steps=12)
            pre_s += yhat.tolist()
            label_s += label_list[t:t+12]
            history = history + test_list[t:t+12]
            model = ARIMA(history, order=(p, 0, q)).fit()
            end_time = datetime.datetime.now()
            total_time = end_time - start_time
            if total==0:
                total = total_time
            else:
                total+=total_time
        except:
            pass
    print(np.array(pre_s).shape)
    print("Total running times is : %f" % total.total_seconds())

    # forecasting results
  prior = ts.values.tolist()[:train.shape[0]][-periods:]
    for i in range(0, len(pre_s), 12):
        for j in range(12):
            pre_s[i + j] = pre_s[i + j] + prior[j] # adding period to
predicted values
        prior = prior[12:] + pre_s[i: i + 12] # last value adds into prior
list, and removes first index from prior

    # model evaluation on three metrics
  metric(pred = np.array(pre_s), label = np.array(label_s))

    # visualization
  plt.figure(figsize=(12, 4))
    plt.plot(label_s, label='Observed', color='black')
    plt.plot(pre_s, label='ARIMA', color='red')
    plt.ylabel("Traffic flow", font1)
    plt.title("Monitoring station 43", font1)
    plt.legend()
    plt.grid(True)
    plt.show()

    return np.reshape(pre_s, [-1, 12]), np.reshape(label_s,[-1, 12])  #
[None, 12], [None, 12]
```

After the model training process and hyperparameters selection, the fitting performance of ARIMA in the test dataset is given in Fig. R6.

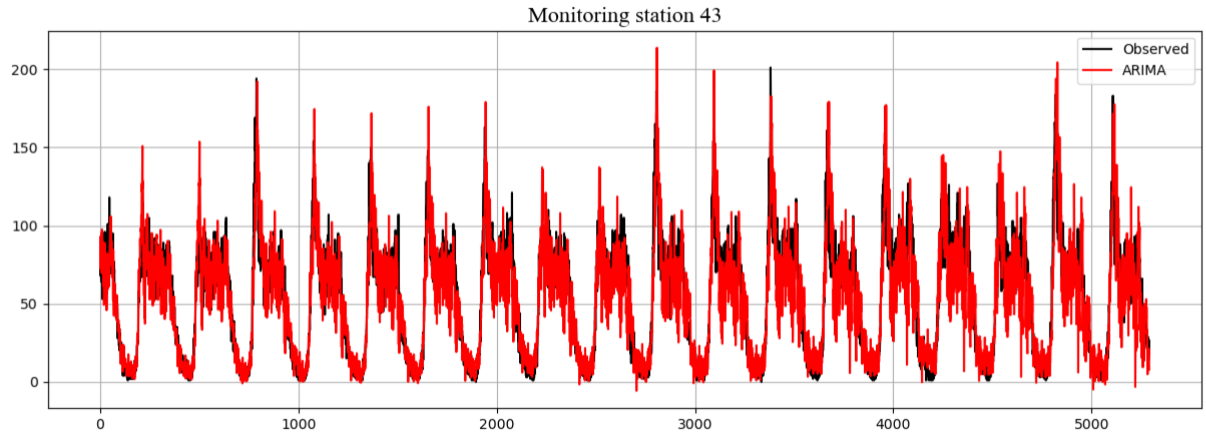

Fig. R6 Example of fitting performance for monitoring station 43 (ARIMA).

For the experiment of the example station 43 (i.e., monitoring sensor index), the MAE, RMSE, and MAPE are 13.999, 19.261, and 51.983%, respectively, and the real fitting performance is visualized in Fig. R6 for the next twelve-time steps forecasting. The optimal model parameters are shown in the following table,

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                21196
Model:                 ARIMA(4, 0, 3)   Log Likelihood              -83049.507
Date:                Sat, 14 Oct 2023   AIC                         166117.014
Time:                        22:17:04   BIC                         166188.668
Sample:                             0   HQIC                        166140.391
                              - 21196
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0029      0.472     -0.006      0.995      -0.927       0.922
ar.L1          0.9247      0.017     54.700      0.000       0.892       0.958
ar.L2          0.9591      0.007    128.207      0.000       0.944       0.974
ar.L3         -0.9219      0.016    -56.717      0.000      -0.954      -0.890
ar.L4          0.0285      0.007      4.005      0.000       0.015       0.042
ma.L1         -0.7277      0.016    -45.520      0.000      -0.759      -0.696
ma.L2         -0.9739      0.005   -210.973      0.000      -0.983      -0.965
ma.L3          0.7526      0.012     62.729      0.000       0.729       0.776
sigma2       165.2250      1.076    153.600      0.000     163.117     167.333
===================================================================================
Ljung-Box (L1) (Q):                   0.36   Jarque-Bera (JB):              7253.21
Prob(Q):                              0.55   Prob(JB):                         0.00
Heteroskedasticity (H):               1.23   Skew:                             0.04
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.86
===================================================================================
```

Step 3. Because periodicity exists in the dataset, the sessional difference needs to be handed, and the period is 12*24=288 with the difference $D = 1$. According to the above analysis for ARIMA, the values of difference d, seasonal difference D, and periodicity s are confirmed. For the SARIMA, four other parameters need to be confirmed, including AR parameters (p), MA parameters (q), and the seasonal component of the model for the AR parameters (P) and MA parameters (Q). Similar to ARIMA, to find the optimal SARIMA, the AR parameters (p), MA parameters (q), and the seasonal component of the model for the AR parameters (P) and MA parameters (Q) are optimized via the find_pq_PQ(ts, m, d, D, max_p, max_q, max_P, max_Q) function. The whole program SARIMA is given, and readers and audiences can also download it to local personal devices for training and testing (we are looking forward to receiving the reader's and the audience's response about their experience on similar datasets using SARIMA, sharing the issues the meet on the GitHub page, and the link is: https://github.com/zouguojian/Traffic-flow-prediction/tree/main/MT-STNet/baselines); the codes are shown in the following,

SARIMA III

```
def find_pq_PQ(ts, m, d, D, max_p=5, max_q=5, max_P=2, max_Q=2):
    best_p, best_q = 0, 0
    best_P, best_Q = 0, 0
    best_aic = np.inf

    for p in range(max_p):
        print('p is : ',p)
        for q in range(max_q):
            for P in range(max_P):
                for Q in range(max_Q):
                    model = SARIMAX(ts, order=(p, d, q), seasonal_order=(P,
D, Q, m)).fit(disp=-1)
                    aic = model.aic

                    if aic < best_aic:
                        best_aic = aic
                        best_p = p
                        best_q = q
                        best_P = P
                        best_Q = Q

    return best_p, best_q, best_P, best_Q, best_aic


def version_sarima_with_manual(ts):
    """
    SARIMA (statsmodels)
    """
    # period
  periods = 12 * 24

    # data splitting
  train, test = train_test_split(ts, train_size=0.8)
    test_list = test.values.tolist()

    # model training
  d, D = 0, 1
    p, q, P, Q, _ = find_pq_PQ(ts, periods, d=d, D=D)
    model = SARIMAX(train, order=(p, d, q), seasonal_order=(P, D, Q,
periods)).fit(disp=-1)
    # yhat = model.forecast(test.shape[0])
    print(model.summary())

    history = train.values.tolist()
    pre_s = []
    label_s = []
    total = 0
    for t in range(0, test.shape[0]-12, 12):
        print(t)
        try:
            start_time = datetime.datetime.now()
            yhat = model.forecast(steps=12)
            pre_s += yhat.tolist()
            label_s += test_list[t:t+12]
            history = history[122:] + test_list[t:t+12]
```

```
            model = SARIMAX(history, order=(p, d, q), seasonal_order=(P, D,
Q, periods)).fit(disp=-1)
            end_time = datetime.datetime.now()
            total_time = end_time - start_time
            if total==0:
                total = total_time
            else:
                total+=total_time
        except:
            pass
    print(np.array(pre_s).shape)
    print("Total running times is : %f" % total.total_seconds())

    # model evaluation
  metric(pred = np.array(pre_s), label = np.array(label_s))

    # visualization
  plt.figure(figsize=(12, 4))
    plt.plot(label_s, label='Observed', color='black')
    plt.plot(pre_s, label='SARIMA', color='red')
    plt.legend()
    plt.grid(True)
    plt.show()
```

After the model training process and parameters selection, the optimal SARIMA is obtained, and the fitting performance in the test dataset is given in Fig. R7.

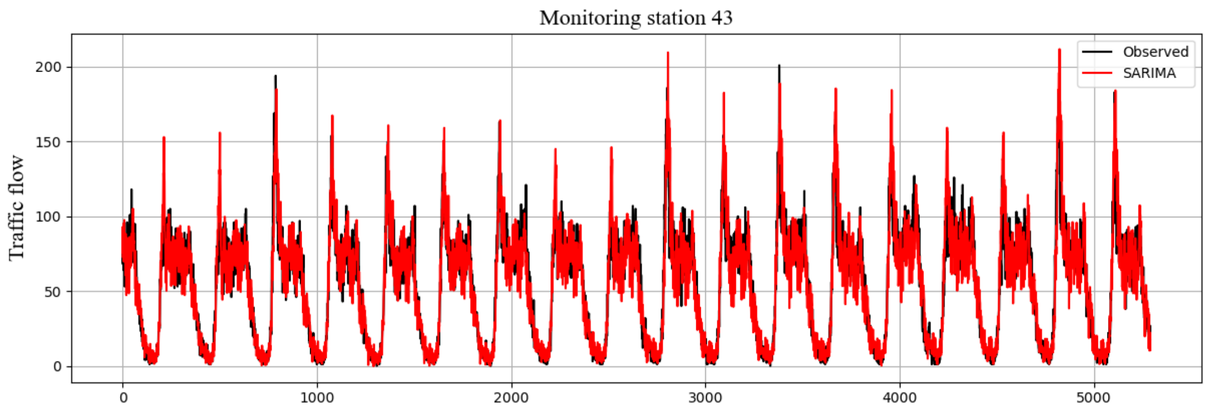

Fig. R7 Example of fitting performance for monitoring station 43 (SARIMA).

Same as ARIMA, for the experiment of the example station 43 (i.e., monitoring sensor index), the MAE, RMASE, and MAPE are 10.700, 15.029, and 37.362%, respectively, and the real fitting performance is visualized in Fig. R7 for the next twelve-time steps forecasting. Compared to ARIMA, SARIMA improved the MAE, RMSE, and MAPE by 23.566%, 21.972%, and 28.127%, respectively, for the next twelve horizons prediction. The experimental results demonstrate the seasonal. The optimal model parameters are shown in the following table,

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                flow   No. Observations:                21196
Model:             SARIMAX(3, 0, 3)x(0, 1, [1], 288)   Log Likelihood              -83068.844
Date:                             Tue, 10 Oct 2023   AIC                         166155.689
Time:                                     11:59:21   BIC                         166227.220
Sample:                                          0   HQIC                        166179.041
                                           - 21196
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.7754      0.034     51.923      0.000       1.708       1.842
ar.L2         -1.7070      0.037    -45.945      0.000      -1.780      -1.634
ar.L3          1.7489      0.037     47.911      0.000       1.677       1.820
ar.L4         -0.8303      0.031    -26.700      0.000      -0.891      -0.769
ma.L1         -1.5974      0.033    -47.978      0.000      -1.663      -1.532
ma.L2          1.5639      0.032     49.424      0.000       1.502       1.626
ma.L3         -1.5744      0.036    -44.320      0.000      -1.644      -1.505
ma.L4          0.6831      0.023     29.128      0.000       0.637       0.729
sigma2       163.4217      1.038    157.479      0.000     161.388     165.456
===================================================================================
Ljung-Box (L1) (Q):                   9.89   Jarque-Bera (JB):              6907.11
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):               1.19   Skew:                             0.04
Prob(H) (two-sided):                  0.00   Kurtosis:                         5.81
===================================================================================
```

In this response, for ARIMA and SARIMA methods, we have provided detailed data analysis, the optimal hyperparameters finding, and the whole program codes. The experiments demonstrate that the performance of SARIMA is higher than ARIMA regarding all metrics. But note that we have spent a lot of time on hyperparameter selection (i.e., we need 10 hours to find the optimal group of parameters in the training dataset for each monitoring station, and we know the size of the training dataset is 21196), and the computation cost is given in Table VI of the manuscript. Because of page limitation, we have uploaded all the above context to our GitHub page, but the experimental results have been added to Sec. V, and the reference Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of Transportation Engineering, 129(6), 664–672. https://doi.org/10.1061/(asce)0733-947x(2003)129:6(664) is added to Sec. I and II. Thanks again to the reviewer for providing this professional suggestion and allowing us to improve the quality of our work.

(2) Another serious weakness of the paper is that the authors do not describe their training and test datasets. The length of the respective datasets, time periods, missing data, descriptive statistics should all be provided. This is a matter of acceptable study publication practice. Readers must not only be able to properly assess the rigor or the study, readers with sufficient correlative expertise should also be able to reproduce the authors analysis. Neither is possible without sufficient detail on the data and the forecast methods.

Response: Thanks for your professional suggestion. We have provided a detailed description for training, validation, and test datasets according to the reviewer's suggestion, as shown in the following Table R6. We especially describe the data properties from six aspects: time period, mean, std, data missing rate, min, max, and length. First, the time period is defined, which consists of the data length. For example, in the validation dataset, the time period occupied 10% of the total source dataset, and the dataset length is 2650. Second, data missing is a general problem in our traffic system because of monitoring sensor faults and bad weather, which causes partial data records with nan in the source database. In this paper, we use an average of the same time as the nearest weeks to fill in the missing values. Third, the mean, std, min, and max metrics are used to describe the training, validation, and test datasets, respectively. Note that because the max value is different in training, validation, and test datasets, the max-min normalization function is abandoned in this work to avoid the max value being higher than the current in the training dataset. In addition, the max-min normalization function easily affects the practice application when the prediction model is deployed in the real-world system, according to our project experience. To solve this problem, the z-score normalization method is used in our work, the performance is verified, and project codes are uploaded to the Github page (https://github.com/zouguojian/Traffic-flow-prediction).

Table R6 Data description for training, validation, and test datasets.

Data	Description	
Training dataset	Time period	June 1, 2021-August 3, 2021
	Mean	29.850
	Std	31.831
	Data missing rate	3.009%
	Min	0
	Max	320
	Length	18547
Validation dataset	Time period	August 3, 2021-August 13, 2021
	Mean	25.392
	Std	27.821
	Data missing rate	3.129%
	Min	0
	Max	314
	Length	2650
Test dataset	Time period	August 13, 2021-August 31, 2021
	Mean	28.294
	Std	30.857
	Data missing rate	3.076%
	Min	0
	Max	319
	Length	7949

Thanks again for the reviewer's suggestion. Like the existing works [1], [2], [3], we have given the data description in the Sec. V, including where data is from, how to define the network and calculate the distance between monitoring sensors, what traffic elements we have, and what the period and granularity are. To convince the audiences and readers to reproduce our work directly and apply it in the other dataset or relative tasks, this response file is converted to a PDF file and uploaded to the GitHub page.

[1] Ma, C., Dai, G., & Zhou, J. (2021). Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method. IEEE Transactions on Intelligent Transportation Systems, 23(6), 5615-5624.
[2] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... & Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. IEEE transactions on intelligent transportation systems, 21(9), 3848-3858.
[3] Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019, July). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 922-929).

What are the contributions of this paper?: The paper purports to present a new "long-term" traffic flow prediction model.

(3) What are some ways in which the paper could be improved? Please supply any additional important references that you feel the author omitted which should be noted in the paper.: The authors must correct there characterization of their method as "long-term" forecasting. The five, ten, and 30-minute forecasts time frames are all considered "short-term" in literature and practice.

Response: Thanks to the reviewer's suggestion. The reviewer's suggestion is similar to reviewer 1 proposed questions (i.e., Q1 and Q2). The 'long-term' has appeared in high frequency in existing studies, such as GMAN [1], DCRNN [2], and Graph-WaveNet [3], which is why we used the 'long-term' in this paper. According to reviewers' emphasis and previous knowledge, ‘long-term’ is not the best definition for our prediction task. To avoid the definition falling into disputation, preventing audiences from talking about the boundary between ‘long-term’ and ‘short-term’ predictions, a new designation named ‘multi-step’ forecasting is thought in this article to replace the old one. The meaning of ‘multi-step’ is utilizing the historical values to predict multiple time steps traffic flow, and the time horizons are extended from six to twelve to be consistent with some popular traffic prediction models in recent years, such as Graph WaveNet [3], MTGNN [4], etc. Finally, thanks again to the reviewer for providing this crucial suggestion.

[1] Zheng, C., Fan, X., Wang, C., & Qi, J. (2020, April). Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 1234-1241).

[2] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018, February). Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In International Conference on Learning Representations.

[3] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019, August). Graph wavenet for deep spatial-temporal graph modeling. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (pp. 1907-1913).

[4] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2020, August). Connecting the dots: Multivariate time series forecasting with graph neural networks. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 753-763).

(4) The authors must also correct their ARIMA forecast method.

Response: Thanks to the reviewer's suggestion. Dear reviewer, according to your professional idea, we have used the seasonal ARIMA (SARIMA) instead of the original ARIMA. The detailed implements are given in question (1), such as hyperparameters selection, and the prediction performance has been added to Sec. V of the manuscript. In addition, we have uploaded both ARIMA and SARIMA methods to my GitHub page (https://github.com/zouguojian/Traffic-flow-prediction/tree/main/MT-STNet/baselines).

(5) If you are suggesting additional references they must be entered in the text box provided. All suggestions must include full bibliographic information plus a DOI. If you are not suggesting any references, type N/A.: Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of Transportation Engineering, 129(6), 664–672. https://doi.org/10.1061/(asce)0733-947x(2003)129:6(664).

Response: Thank you so much for your professional suggestion. The author has added the recommended reference to the Sec. I and II. In addition, the Seasonal ARIMA (SARIMA) has been used as a baseline in the experiment to replace the traditional ARIMA.