


R-CTM: A data-driven macroscopic simulation model for heterogeneous traffic

Jingyao Liu¹  | Zhigang Deng² | Tianlu Mao¹ | Zhaoqi Wang¹

¹Beijing Key Lab of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

²Department of Computer Science, University of Houston, Houston, Texas, USA

Correspondence

Tianlu Mao, Beijing Key Lab of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.
Email: ltm@ict.ac.cn

Funding information

National Key Research and Development Program of China, Grant/Award Number: 2017YFC0804900; National Natural Science Foundation of China, Grant/Award Number: 61532002

Abstract

There is a well-known trade-off between computational efficiency and computational accuracy in the field of traffic simulation. In this article, we propose a novel recurrent neural network based model with an integrated attention mechanism, called R-CTM, to simulate heterogeneous traffic flow with multiple types of vehicles. It can effectively extract the traffic flow patterns of spatial and temporal changes from training traffic data, which can be real-world traffic data or synthetic traffic data via microscopic simulation models. Through experiments and comparisons, we show that it can significantly outperform the state of the art methods in terms of simulation accuracy. Besides accuracy, we also demonstrate its scalability: its runtime consumption does not linearly increase with respect to the spatial extent.

KEYWORDS

cell transmission models, heterogeneous traffic flow, macroscopic traffic modeling, recurrent neural networks

1 | INTRODUCTION

Dynamic traffic flow models are an important component of a variety of applications including games, urban planning/visualization, evacuation simulation, and autonomous driving.¹ One of such applications is emergency evacuation planning for severe disasters such as hurricanes, nuclear leaks, and so forth. In such scenarios, people in a certain area need to be quickly and efficiently evacuated to one or more resettlement places by vehicles. An efficient and accurate simulation model can provide a powerful tool in evaluating and optimizing evacuation planning by generating realistic traffic flow on specific routes.

Over the years, many traffic simulation models have been developed. Microscopic models are mostly used to simulate the behavior of individual vehicles in the road network. However, the computational cost of microscopic methods increases rapidly with the number of vehicles and the size of the road network, making them difficult to be scaled to large-scale scenarios. On the other hand, macroscopic traffic models consider the traffic flow as compressible fluid and evaluate the traffic states with the help of aggregated variables, for example, average speed, flow, and density. Often, they design specific functions to describe the traffic flow and use data from real world for parameter calibration to achieve accurate simulation results. However, macroscopic methods generally have high requirements for parameter tuning to achieve accurate simulations. If the parameters are calibrated using data different from target simulation conditions, the simulations by them may contain noticeable distortions or artifacts. However, in many real-world applications, such as emergency evacuation planning, it is often impossible or even practically infeasible to collect such data in the same or

similar situations/environments in the real world for model calibration. Furthermore, such errors tend to accumulate and increase in large-scale scenarios.

Traffic reconstruction methods based on computer graphics can reflect the traffic flow moving pattern. The trajectories of vehicles in the traffic flow are reconstructed from traffic detector data in such methods. Sewall et al.² proposed a method to reconstruct and visualize continuous traffic flow from the spatio-temporal data provided by traffic sensors. Chao et al.³ synthesize new vehicle trajectories through a fusion of texture synthesis and traffic behavior rules, using a limited set of vehicle trajectories as input samples. Such approaches can realistically reproduce the traffic flow in a real road network. However, such methods recover the details of traffic flow changes from the a priori knowledge of detectors essentially. It is not suitable for applications such as urban road planning and emergency evacuation, where such priori knowledge is not available.

Neither microscopic models nor macroscopic models can effectively simulate large-scale heterogeneous traffic flow, which often consists of mixed types of vehicles with different sizes and driving patterns. To tackle this challenge, in this article, we propose a deep learning based macroscopic traffic simulation model for heterogeneous traffic flow that contains mixed different types of vehicles. Our model, called R-CTM, is based on the concept of the cell transmission models (CTM). Given an initial vehicle distribution and speed conditions, our model can accurately and efficiently simulate large-scale heterogeneous traffic flow. It captures subtle nonlinear features from the data and further effectively integrates spatial and temporal information so that the aforementioned limitations of both the microscopic and the macroscopic methods can be alleviated or even avoided.

The main contributions of this article can be summarized below.

- To the best of our knowledge, our R-CTM model is the first *deep learning* based *macroscopic* traffic simulation model based on the cell transmission framework.
- Our model significantly improves the simulation accuracy, compared to existing macroscopic methods and existing machine learning based methods with almost the same computational time. Moreover, the computational time consumption of our model does not increase linearly with respect to the spatial extent.
- Different from existing macroscopic models that highly depend on parameter tuning, our model can simulate the same road environment (i.e., number of lanes, speed limit, etc.) and traffic conditions (i.e., vehicle types, speed distributions, etc.) without the recalibration of the parameters, even when many factors including vehicle density and vehicle proportions are changed.

2 | RELATED WORK

We can roughly divide existing traffic simulation models into two categories:¹ macroscopic models and microscopic models. Among existing macroscopic traffic simulation models, the Lighthill–Whitham–Richards (LWR) model^{4,5} is the most popular one, which models the motion of traffic flow in a hydrodynamic term. Based on the LWR model, Daganzo⁶ proposed the CTM, which is a numerical method to solve differential equations of the LWR model. Based on the above CTM framework, researchers have introduced various extensions to improve its applicability and accuracy. To produce platoon dispersion phenomena, LWR was extended to multi-class LWR,⁷ which introduces the notion of classes defining heterogeneous driving on a freeway. This method roughly divides traffic flow into two categories: passenger vehicles (PV) and heavy vehicles (HV). Compared with HV, PV has a shorter length and a faster speed. Researchers have developed a few multi-class CTM models for simulating traffic flow with a certain percentage of HVs on freeways.^{8–14} Also, to produce platoon dispersion in general topology, CTM was extended to a multi-class CTM, known as M-CTM.¹⁵ Based on the reported results, M-CTM can produce platoon dispersion without compromising the model's computational complexity. Liu et al.¹³ modified the CTM model using methods from the lagged CTM to reproduce the moving bottlenecks caused by buses within the CTM framework. Tiaprasert et al.¹⁶ proposed FM-CTM, standing for the multi-class CTM with FIFO property, to improve the accuracy of the multi-class CTM model.

In recent years, machine learning algorithms, especially deep learning algorithms, have made remarkable progress in the field of spatio-temporal data mining.¹⁷ The neural networks algorithms that are widely used for spatio-temporal data mining is the recurrent neural networks (RNNs).¹⁸ The RNN has been proved to be powerful for processing time-series data and achieved great success in many applications, including speech recognition,^{19,20} natural language processing,²¹

target tracking,²² and so forth. In transportation applications, deep learning has also increasingly demonstrated its usefulness.

It is noteworthy that although our method is inspired by the CTM framework, our model should not be technically categorized as an extension of the CTM model. The main reason is that proposed model breaks the Courant–Friedrichs–Lewy condition, by allowing the traffic to move through multiple cells during one time step.

3 | METHODOLOGY

3.1 | Pipeline

We divide the road into cells, following the traditional CTM approach. The length of each cell is taken as a fixed value l , which is set to 50 m in practice. Because we simulate traffic in a freeway scenario, each cell is connected sequentially in the spatial order. The number of the outflow vehicles in each cell is the number of the inflow vehicles in its next cell. There are multiple types of vehicles driving on the road, denoted as v_0, v_1, \dots, v_n . Each type of vehicle is of a certain length l_{v_i} , having a random speed that follows a Gaussian distribution.

For each cell s , we focus on the following variables that reflect its traffic state: the numbers of vehicles flows into and out of cell s between time $t - 1$ and t for vehicle type v , denoted as $I_{t,s}^v, O_{t,s}^v$, respectively; the number of vehicle type v in the cell s at time t , denoted as $N_{t,s}^v$. Further, we can write these variables as follows:

$$X_{s,t} = [I_{t,s}^{v_0}, O_{t,s}^{v_0}, N_{t,s}^{v_0}, I_{t,s}^{v_1}, O_{t,s}^{v_1}, N_{t,s}^{v_1}, \dots]^T.$$

Our model can be conceptually regarded as a state machine. Each cell has a corresponding state vector h_t^s to represent its current state. It should be noted that the state vector here does not have any actual physical meaning compared to X . It is an intermediate variable that contains key information within the model. It is calculated based on $X_{s,t}, h_{t-1}^i, i \in \mathcal{N}_s \cup \{s\}$. \mathcal{N}_s denotes the neighborhood of the cell s , as shown in Figure 1, it includes all nodes within the maximum distance that can be traveled from or to cell s with the max speed v_{\max} in a single time step Δt . The simulation steps of our model can be described formally as follows:

$$\begin{aligned} \hat{\ell} &= [h_{t-1}^i, \forall i \in \mathcal{N}_s \cup \{s\}], \\ h_t^s &= \mathcal{M}(X_{s,t}, \hat{\ell}), \\ \hat{X}_{s,t+1} &= \mathcal{F}(h_t^s). \end{aligned}$$

In the above equations, $\mathcal{M}(\cdot)$ is a state estimation function, and $\mathcal{F}(\cdot)$ is a prediction function, which is explained in detail later, and $\hat{X}_{s,t+1}$ is the predicted value of the input at time step $t + 1$.

Before the simulation starts, the initialization at the first k steps is created in advance. In other words, the first k input variables need to be set according to the preset vehicle distribution, which is called *preset inputs* in this work. Starting from time step $k + 1$, the predicted value at each time step is used as the input of the next time step. In emergency evacuation applications, all preset inputs can be set to 0, which indicates that evacuation starts in an empty road. In this work, k is empirically set to 4.

Our model is trained using a simulated dataset. Deep learning models often require a large amount, evenly distributed dataset for training. Practically, traffic flow data acquired from real world are often too small to train deep learning models, besides their uneven distribution. Therefore, we generate our training traffic dataset through microscopic simulation models as the alternative. In order to ensure the accuracy of the obtained simulations, we train the microscopic simulation



FIGURE 1 The white cell is the current cell, the blue ones are its neighboring cells, and the green ones are the irrelevant cells. The states of the cells of interest (white and blue) are the input to the state estimation function.

model using data generated under the same road network conditions (e.g., the number of lanes, speed limit, etc.) and traffic flow conditions (e.g., size, speed distribution, etc.). It should be noted that our trained model does not need to be retrained for different vehicle densities, vehicle type ratios, road lengths, and so forth.

3.2 | Model details

As mentioned above, there are two main parts in our model. One is the state estimation function $\mathcal{M}(\cdot)$, and the other is the prediction function $\mathcal{F}(\cdot)$. Next, we describe the computational process for both of them in detail.

The state estimation function is designed to compute the current cell state. In our model, the traffic state estimation function gives the state of the current cell based on the current cell's state and the states of the surrounding cells at the previous time step. It consists of two main components: a multiple attention module for processing spatial information, and an LSTM module for processing temporal information.

The computing process of the state estimation function is illustrated in Figure 2. The superscript of the letters in the figure represents the cell number, the subscript indicates time stamp, and h and c denote the hidden state and the cell state in LSTM, respectively. The model collects the hidden states of a few cells associated with the current cell at the previous time step and processes the collected hidden states into spatial information through a multi-head attention layer.

After that, the spatial information first goes through two gate structures to control both the hidden state and the cell state of the current cell, and then enters into a conventional LSTM cell for further calculation. The above procedure can be formalized as follows:

$$\begin{aligned} h_{sp}^s &= \text{ATTENTION}(\dots, h_{t-1}^{s-1}, h_{t-1}^{s+1}, \dots), \\ \hat{h}_{t-1}^s &= h_{t-1}^s \circ \sigma_g(W_{sf} h_{sp}^s + b_{sf}), \\ \hat{c}_{t-1}^s &= c_{t-1}^s \circ \sigma_g(W_{si} h_{sp}^s + b_{si}), \\ h_t^s, c_t^s &= \text{LSTM}(\hat{h}_{t-1}^s, \hat{c}_{t-1}^s, x_t^s). \end{aligned}$$

Next, we explain the structure of the attention layer in detail. The purpose of the attention mechanism is to reasonably distribute the weights of different inputs. In our model, the multiple attention layers need to assign weights to spatially relevant cells' states. As mentioned previously, each cell is relevant with those cells (including itself) within l distance in both the forward and backward directions. l is calculated as $l = V_{\max} \times \Delta t$, where V_{\max} is the maximum speed, and Δt is the length of a time step.

We added learnable parameters to act as the spatial information of upstream or downstream cells of the boundary cells, since the boundary cells have no upstream or downstream cells. The detail is illustrated in

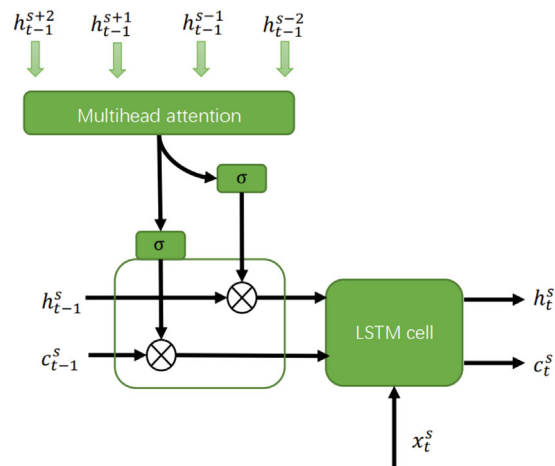


FIGURE 2 The estimation function structure of our R-CTM model

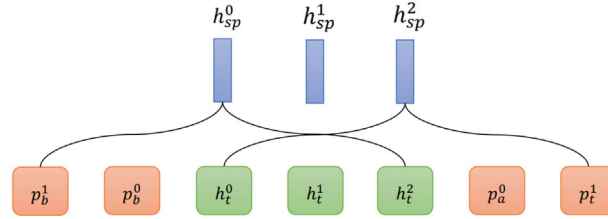


FIGURE 3 The spatial relationship of parameters

Figure 3. These parameters play an important role to the accuracy of our model, as shown in the supplemental materials.

The attention mechanism used in this work is computed as follows:

$$\begin{aligned}
 e_{ij} &= \text{LeakyReLU}(W_a[h_i, h_j]), \\
 \alpha_{ij} &= \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{j \in \mathcal{N}_i} \exp(e_{ij})}, \\
 h'_i &= \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} h_j\right).
 \end{aligned}$$

Here h'_i represents the result of one head in the multi-head attention layer.

By concatenating the results of all the heads, we obtain the aforementioned spatial information as follows:

$$h_{sp} = \text{CONCAT}(h'_1, h'_2, \dots).$$

The prediction function does not have a complex structure as the state estimation function. We first use a fully connected layer to predict the number of vehicles that leave the current cell s at the next time step based on the current state h_t^s , that is,

$$O_{s,t+1} = FC(h_t^s).$$

Here FC denotes the fully connected layer, which is a linear function with parameters, $O_{s,t+1}$ is a vector containing the number of outflows for all vehicle types, that is, $O_{s,t+1} = [O_{s,t+1}^{v_0}, O_{s,t+1}^{v_1}, \dots]$. Since each cell is sequentially connected, every cell's outflows go to the next cell. So we can easily obtain the input at the next time step through a simple transformation as follows:

$$\begin{aligned}
 I_{s,t+1} &= O_{s-1,t+1}, \\
 N_{s,t+1} &= N_{s,t} + i_{s,t+1} - O_{s,t+1}.
 \end{aligned}$$

The number of inflow vehicles in the first cell $I_{0,t+1}$ takes the ground-truth value, which is also the variable to control the flow of traffic on the road during simulations.

The loss function designed in our model needs to be accurate not only on the output flow of a cell but also on the number of vehicles in it. In the following formula, α and β are the hyperparameters that need to be calibrated.

$$L = \alpha \text{Loss}(O_t, \hat{O}_t) + \beta \text{Loss}(N_t, \hat{N}_t).$$

During the training process, model parameters are optimized by the Adam²³ to find the parameters with the minimal loss on the training dataset. In our experiments, the loss function used in our work is the absolute error as follows:

$$\text{Loss}(y, \hat{y}) = \|y - \hat{y}\|_2^2.$$

4 | EXPERIMENTS AND COMPARISONS

To keep the text brief, we attach the dataset, training, implementation details, and a ablation study in the supplementary material.

4.1 | Quantitative metrics

We compare the simulation capabilities of different methods using the following two quantitative metrics: the first, called *cell_error*, which is the same as RMSE in Reference 16. It is the root mean square error between the number of vehicles predicted within a single cell \hat{N}_s^v and the data obtained through SUMO simulation N_s^v , which is computed as:

$$\text{cell_error}_{pv} = \sqrt{\frac{\sum_t^T \sum_s^I (\hat{N}_{t,s}^{pv} - N_{t,s}^{pv})^2}{T \cdot I}},$$

$$\text{cell_error} = \text{cell_error}_{pv} + \text{cell_error}_{hv}.$$

The other, called *seg_error*, is the error between the number of vehicles predicted $\sum_s \hat{N}_{t,s}^v$ and the ground-truth value obtained through SUMO simulation $\sum_s N_{t,s}^v$ over the entire road, which is computed as:

$$\text{seg_error}_{pv} = \sqrt{\frac{\sum_t^T (\sum_s \hat{N}_{t,s}^{pv} - \sum_s N_{t,s}^{pv})^2}{T}},$$

$$\text{seg_error} = \text{seg_error}_{pv} + \text{seg_error}_{hv}.$$

In the above equations, I and T are the total numbers of the observed cells and time steps, respectively. The errors within a single cell can directly show the accuracy of the model. And, the errors on the whole road can show how the errors are accumulated in the simulation space. In this way, we can quantitatively compare the simulation qualities of different models with error diagrams and space-time heat maps.

4.2 | Comparison with existing methods

To evaluate the effectiveness of our model, we compare our model with several state of the art methods, including a conventional CTM with mixed vehicle types, FM-CTM,¹⁶ a modified multilayer perceptron model (MLP),²⁴ LSTM model,²⁵ and the social-LSTM model²² on the aforementioned synthetic dataset and the real-world dataset.

4.2.1 | Comparison on the synthetic data

We compare the simulation results by different methods with the synthetic data (as the ground-truth). As we can see from Figure 4, the MLP and LSTM methods fail to capture the nonlinear characteristics of the data and cannot fit the overall trend of traffic flow during a long simulation. This shows that macroscopic traffic data generally contain complex nonlinear characteristics.

Table 1 shows the simulation errors by the FM-CTM, social-LSTM, and our model (R-CTM). As shown in this table, considering that the maximum number of vehicles in a single cell is around 20, all of the three models achieve relatively high accuracies at the individual cell level. In contrast, at the segment level, our R-CTM model achieves significantly smaller errors than both the FM-CTM and the social-LSTM models.

Figure 5 shows the number of vehicles on the road in the simulation results by different methods, compared to the ground truth. As we can see from this figure, the social-LSTM model can accurately predict the traffic flow variation pattern in low density scenarios, but the error is more noticeable in high density scenarios. The simulation results by the FM-CTM model are roughly similar to the ground truth that was generated by the SUMO microscopic model. But we

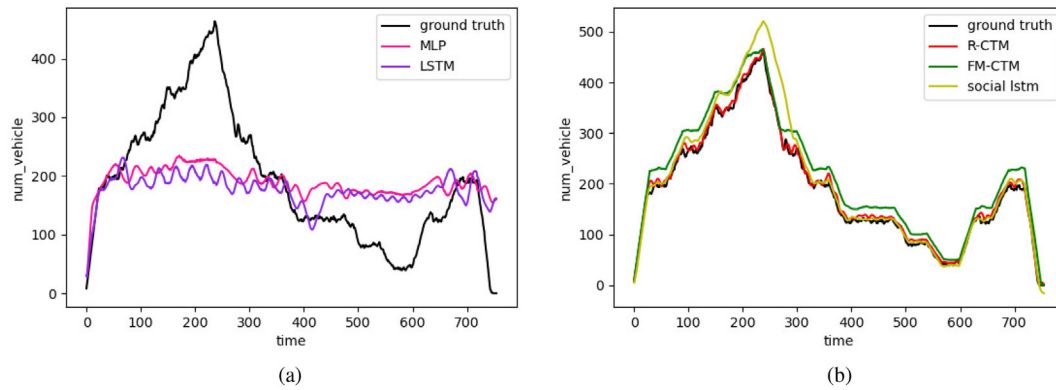


FIGURE 4 (a) The results of the MLP and LSTM models tested on synthetic traffic data with 30% HVs. (b) The results of FM-CTM, social-LSTM, and R-CTM models under the same conditions

TABLE 1 Test errors on the synthetic data

HV's ratio	Cell error			Seg error		
	R-CTM	FM-CTM	Social-LSTM	R-CTM	FM-CTM	social-LSTM
5%	1.70	1.91	1.96	6.93	26.89	17.63
10%	1.68	1.92	2.01	7.02	27.44	20.45
15%	1.71	2.0	2.00	7.99	26.51	21.37
20%	1.72	2.01	2.03	9.64	26.74	25.22
25%	1.69	2.06	1.99	11.09	26.53	24.41
30%	1.75	2.1	1.98	11.85	25.67	23.93

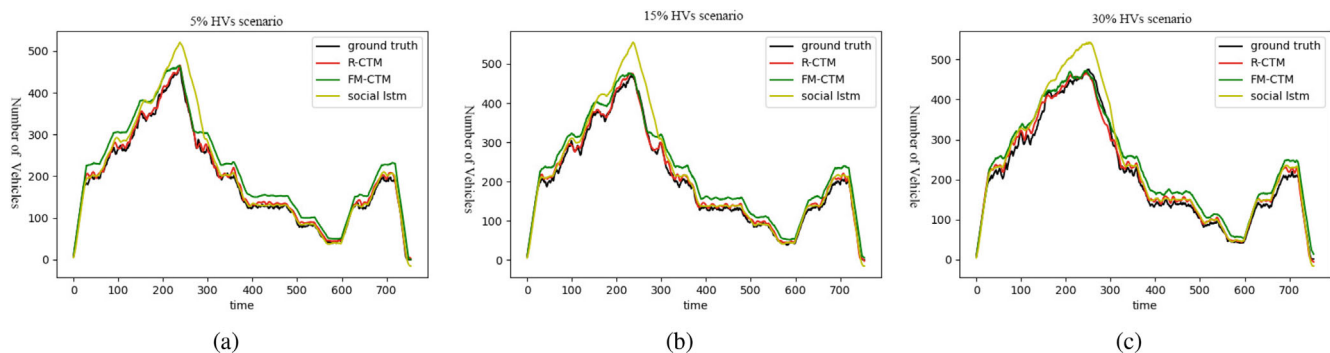


FIGURE 5 Comparisons between the simulated results by three different models and the SUMO simulation results for the case containing (a) 5%, (b) 15%, and (c) 30% HVs in the traffic flow, respectively

can also observe noticeable errors at certain places. Finally, the simulation results by our R-CTM model can accurately approximate the ground truth in all the scenarios. Numerically, the cell error of our R-CTM model is 15% smaller than both the FM-CTM model and the social-LSTM model. At the segment level, the error by our R-CTM model is smaller than 50% of those by the FM-CTM and the social-LSTM models. This suggests that our R-CTM model can significantly improve the accuracy of simulations, and produce more realistic simulation results.

Figure 6 shows the simulation error for different vehicle types when there is 30% HVs in the traffic flow. It can be seen that the error of FM-CTM mainly originates from the difference in the number of HVs. The error of the social-LSTM model, on the other hand, is independent of the vehicle type, and it increases when the vehicle density is increased.

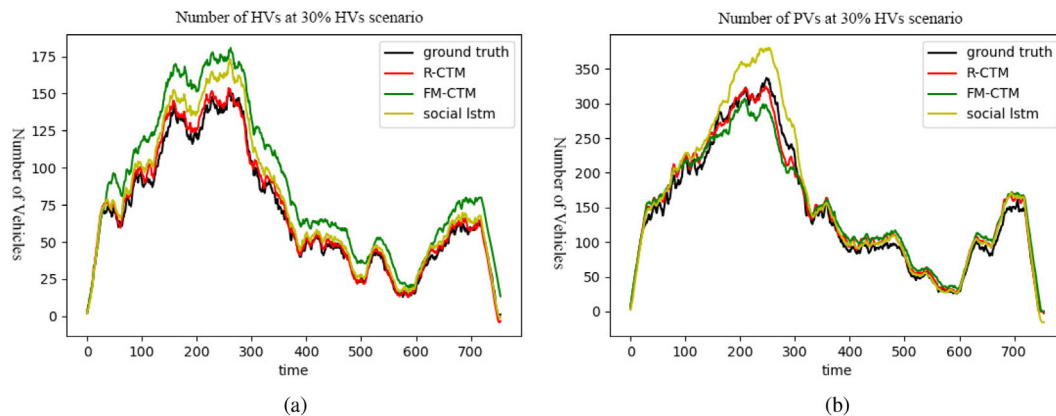


FIGURE 6 (a) The number of HVs over time and (b) the number of PVs. Both figures are for the case where there is 30% HVs in the traffic flow.

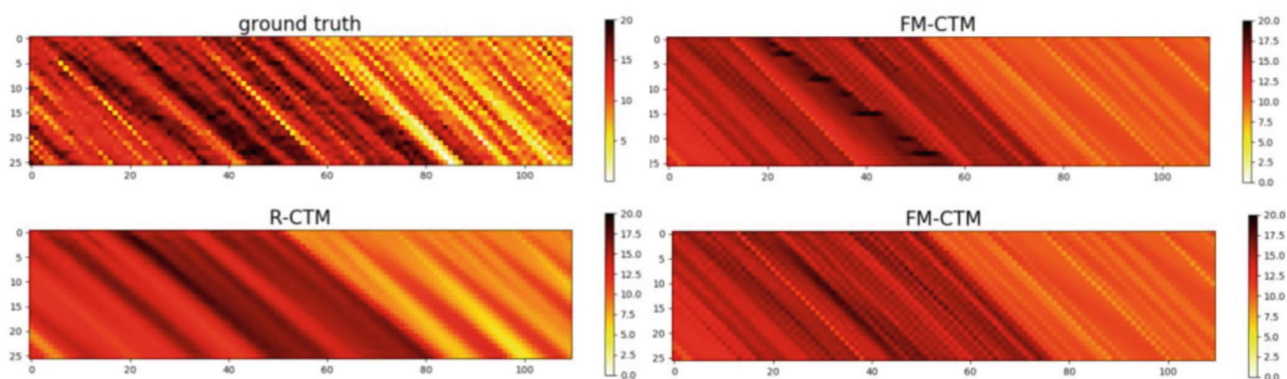


FIGURE 7 The top-left panel shows the ground truth and the bottom-left panel shows the result by our R-CTM model. The top-right panel shows the results by the FM-CTM model after parameter calibration on different datasets, and the bottom-right panel shows the result by the FM-CTM model after parameter calibration on the same dataset.

Discussion. FM-CTM is less effective in simulating HV traffic flow probably because HV vehicles are longer and have a larger variation of speeds, and thus the traffic flow shows more nonlinearity when it contains more HVs. This is consistent with our real-life observation: large vehicles often move at slower speeds, which often may cause congestion on the road. The R-CTM model can more accurately simulate the variation of such large vehicles in the mixed traffic flow.

The social-LSTM model shows less accuracy when the traffic densities of both types of vehicles increase. This indicates there is a difference in traffic flow pattern between low density cases and high density cases. Our R-CTM model is more accurate because the multi-headed attention layer in our model can provide enough nonlinearity to well represent different variations under low or high traffic conditions.

Moreover, the empirically designed segmentation functions in traditional CTM model could distort simulation results. As shown in Figure 7, right top, there are some distortions in the FM-CTM simulation result. This is due to the vehicle congestion caused by setting the maximum throughput of any cell in the segmentation function. When we do parameter calibration specifically for this scenario, the maximum limit of flow is increased from 0.62 to 0.68 (per lane per second), and the congestion can be avoided (Figure 7, right bottom). This shows that the traditional CTM simulation models are limited by the embedded segmentation limitation. Because of this, the simulation scenarios that it can simulate are also limited. When the simulation condition changes significantly, the result by the FM-CTM model is likely to have distortion unless some critical parameters are carefully calibrated, such as in Reference 16, where parameters are calibrated separately for each cell. However, this is almost impossible or technically infeasible in practical simulation applications.

TABLE 2 Runtime statistics (in seconds) of different methods on the same road with different lengths

Road length	2000 m	1750 m	1500 m	1250 m	1000 m	750 m	500 m
Our model	2.73	2.69	2.7	2.72	2.61	2.61	2.66
Social-LSTM	2.01	1.99	1.97	2.03	1.89	1.85	1.81
FM-CTM	2.81	2.15	1.81	1.50	1.18	0.93	0.61
SUMO	1041.2	900.6	791.2	630.1	533.4	370.2	277.8

TABLE 3 Test errors on the real-world dataset

Dataset	Cell error		Seg error	
	R-CTM	FM-CTM	R-CTM	FM-CTM
I80-1	2.72	2.66	11.04	14.20
I80-2	3.52	5.44	13.83	35.70
I80-3	4.97	5.32	22.90	31.35

By contrast, our R-CTM model does not require special parameter calibrations. Specifically, as long as the number of lanes is consistent and the vehicle driving characteristics are similar to those in the training dataset, our model can perform accurate simulations, without causing distortions.

Both our R-CTM model and the social-LSTM model require more computational resources than traditional CTM models. With the parallel computing support of many deep learning frameworks, our model can be executed in parallel. Table 2 shows the runtime consumptions (in seconds) of the R-CTM, social-LSTM, and the FM-CTM models running on a computer with GTX1060 GPU. The simulation scenario is a six-lane road with a variable length of one hour. We can see that despite our R-CTM model takes a longer time than both the social-LSTM and the FM-CTM models, but it is at least two orders of magnitudes faster than the SUMO microscopic simulation method. More importantly, the scalability of the R-CTM model is good: its runtime consumption does not increase linearly with respect to the spatial extent. In contrast, we can observe an approximately linear relation between the time consumption of the FM-CTM model and the length of the simulated road.

4.2.2 | Comparison on the real-world dataset

We also validated our model on the real-world dataset. Because the amount of the real-world dataset is too small to support the training of our R-CTM model, we used SUMO to generate synthetic data for training, based on the obtained distribution of vehicle speeds and the ratios of different vehicle types in the real-world dataset. Finally, we tested the trained R-CTM model on a test set of the real-world dataset. Table 3 shows the test errors by different methods. We can see that although the R-CTM model was trained on synthetic data, it still can achieve a better performance than FM-CTM at both cell level and segment level in most cases.

5 | CONCLUSION

In this article, we present a data-driven macroscopic traffic simulation model, called R-CTM. Its key idea is to replace empirically designed functions in traditional CTM methods with a neural network model. Based on the vanilla LSTM model, we specifically design two key components: additional parameters and a multi-headed attention layer. The multi-headed attention layer can effectively handle the spatial influence on the current cell and significantly improve the model's accuracy. Adding additional parameters at the beginning and the end of the road can effectively reduce errors at both ends of the road and also improve the accuracy. Through comprehensive comparisons with state of the art methods on both a synthetic traffic dataset and a real-world traffic dataset, we demonstrate that our R-CTM model can accurately simulate multi-class vehicles' flow patterns on a long road.

Our R-CTM model provides a new perspective of rethinking CTM models. Specifically, it does not rely on empirically designed functions to describe the features and phenomena of traffic but automatically learns them in a data-driven fashion. As the future work we plan to explore how data-driven models can reliably reproduce the features and phenomena of traffic flow on complex road networks, such as traffic junctions, ramps, and so forth.

It is important to note that our approach still has some limitations. Our proposed method breaks the Courant–Friedrichs–Lewy condition, by allowing the traffic to move through multiple cells during one time step, which is why it is not mathematically guaranteed that the number of traffic flows in the road is consistent. Second, due to the limitations of the machine learning model, the simulation for scenarios with differences from the training data requires retraining, such as inconsistent number of lanes or inconsistent vehicle travel speeds.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0804900 and the National Natural Science Foundation of China under Grant 61532002. We also would like to thank Prof. Maobin Hu of the University of Chinese Academy of Sciences for suggestions on this work.

ORCID

Jingyao Liu  <https://orcid.org/0000-0002-9033-7066>

REFERENCES

1. Chao Q, Bi H, Li W, Mao T, Wang Z, Lin MC, et al. A survey on visual traffic simulation: models, evaluations, and applications in autonomous driving. *Comput Graph Forum*. 2020;39:287–308.
2. Sewall J, Van Den Berg J, Lin M, Manocha D. Virtualized traffic: reconstructing traffic flows from discrete spatiotemporal data. *IEEE Trans Vis Comput Graph*. 2010;17(1):26–37.
3. Chao Q, Deng Z, Ren J, Ye Q, Jin X. Realistic data-driven traffic flow animation using texture synthesis. *IEEE Trans Vis Comput Graph*. 2017;24(2):1167–78.
4. Lighthill MJ, Whitham GB. On kinematic waves ii. a theory of traffic flow on long crowded roads. *Proc Royal Soc Lond Ser A Math Phys Sci*. 1955;229(1178):317–45.
5. Richards PI. Shock waves on the highway. *Oper Res*. 1956;4(1):42–51.
6. Daganzo CF. The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transp Res B Methodol*. 1994;28(4):269–87.
7. Wong GCK, Wong SC. A multi-class traffic flow model—An extension of lwr model with heterogeneous drivers. *Transp Res A Policy Pract*. 2002;36(9):827–41.
8. Ngoduy D. Multiclass first-order traffic model using stochastic fundamental diagrams. *Transportmetrica*. 2011;7(2):111–25.
9. Van Lint JWC, Hoogendoorn SP, Schreuder M. Fastlane: new multiclass first-order traffic flow model. *Transp Res Rec*. 2008;2088(1):177–87.
10. Ngoduy D, Liu R. Multiclass first-order simulation model to explain non-linear traffic phenomena. *Phys A Stat Mech Appl*. 2007;385(2):667–82.
11. Szeto WY, Jiang Y, Sumalee A. A cell-based model for multi-class doubly stochastic dynamic traffic assignment. *Comput Aided Civil Infrastr Eng*. 2011;26(8):595–611.
12. Mesa-Arango R, Ukkusuri SV. Modeling the car-truck interaction in a system-optimal dynamic traffic assignment model. *J Intell Transp Syst*. 2014;18(4):327–38.
13. Liu H, Wang J, Wijayarathna K, Dixit VV, Waller ST. Integrating the bus vehicle class into the cell transmission model. *IEEE Trans Intell Transp Syst*. 2015;16(5):2620–30.
14. Qian ZS, Li J, Li X, Zhang M, Wang H. Modeling heterogeneous traffic flow: a pragmatic approach. *Transp Res B Methodol*. 2017;99:183–204.
15. Tuerprasert K, Aswakul C. Multiclass cell transmission model for heterogeneous mobility in general topology of road network. *J Intell Transp Syst*. 2010;14(2):68–82.
16. Tiaprasert K, Zhang Y, Aswakul C, Jiao J, Ye X. Closed-form multiclass cell transmission model enhanced with overtaking, lane-changing, and first-in first-out properties. *Transp Res C Emerg Technol*. 2017;85:86–110.
17. Wang S, Cao J, Yu P. Deep learning for spatio-temporal data mining: a survey. *IEEE Trans Knowl Data Eng*. 2020. <https://doi.org/10.1109/TKDE.2020.3025580>.
18. Boden M. A guide to recurrent neural networks and backpropagation. The Dallas project; 2002.
19. Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. *Proceedings of the International Conference on Machine Learning, Beijing, China; 2014*. p. 1764–72.
20. Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada; IEEE, 2013*. p. 6645–9.

21. Mikolov T, Karafiát M, Burget L, Černocký J, Khudanpur S. Recurrent neural network based language model. Proceedings of the 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan; 2010.
22. Alahi A, Goel K, Ramanathan V, Robicquet A, Fei-Fei L, Savarese S. Social lstm: human trajectory prediction in crowded spaces. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV; 2016. p. 961–71.
23. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
24. Gardner MW, Dorling SR. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmospheric Environ.* 1998;32(14-15):2627–36.
25. Schmidhuber J, Hochreiter S. Long short-term memory. *Neural Comput.* 1997;9(8):1735–80.

AUTHOR BIOGRAPHIES



Jingyao Liu is currently pursuing the degree with the Department of Institute of Computing Technology, Chinese Academy of Sciences. He is supervised by Dr. Zhaoqi Wang. He focuses on deep learning methods in traffic modeling and simulation.



Zhigang Deng is Moores Professor of Computer Science at University of Houston, Texas, USA. His research interests include computer graphics, computer animation, virtual humans, human computer conversation, and robotics. He earned his Ph.D. in Computer Science at the Department of Computer Science at the University of Southern California in 2006. Prior to that, he also completed a B.S. degree in Mathematics from Xiamen University (China), and M.S. in Computer Science from Peking University (China). Besides serving as the general or program co-chair for CASA 2014, SCA 2015, and MIG 20222, he has been an associate editor for *IEEE Transactions on Visualization and Computer Graphics*, *Computer Graphics Forum*, *Computer Animation and Virtual Worlds Journal*, and so forth. He is a distinguished member of ACM and a senior member of IEEE.



Tianlu Mao received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2009, where she is also working as associate professor. Her research interests include artificial intelligence, modeling, and simulation.



Zhaoqi Wang is a researcher and a director of Ph.D. students with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include virtual reality and intelligent human computer interaction. He is a senior member of the China Computer Federation.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Liu J, Deng Z, Mao T, Wang Z. R-CTM: A data-driven macroscopic simulation model for heterogeneous traffic. *Comput Anim Virtual Worlds.* 2022;e2091. <https://doi.org/10.1002/cav.2091>