Fast-Forwarding of Vehicle Clusters in Microscopic Traffic Simulations

Philipp Andelfinger
TUMCREATE Ltd and
Nanyang Technological University
philipp.andelfinger@gmail.com

David Eckhoff
TUMCREATE Ltd and
Technische Universität München
david.eckhoff@tum-create.edu.sg

Wentong Cai
Nanyang Technological University
aswtcai@ntu.edu.sg

Alois Knoll
Technische Universität München and
Nanyang Technological University
knoll@in.tum.de

ABSTRACT

State fast-forwarding has been proposed as a method to reduce the computational cost of microscopic traffic simulations while retaining per-vehicle trajectories. However, since fast-forwarding relies on vehicles isolated on the road, its benefits extend only to situations of sparse traffic. In this paper, we propose fast-forwarding of vehicle clusters by training artificial neural networks to capture the interactions between vehicles across multiple simulation time steps. We explore various configurations of neural networks in light of the trade-off between accuracy and performance. Measurements in road network simulations demonstrate that cluster fast-forwarding can substantially outperform both time-driven state updates and single-vehicle fast-forwarding, while introducing only a small deviation in travel times.

ACM Reference Format:
Philipp Andelfinger, David Eckhoff, Wentong Cai, and Alois Knoll. 2020. Fast-Forwarding of Vehicle Clusters in Microscopic Traffic Simulations. In *2020 SIGSIM Principles of Advanced Discrete Simulation (SIGSIM-PADS'20), June 15–17, 2020, Miami, FL, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

The per-vehicle modeling approach of microscopic traffic simulation is known to entail a substantial computational burden. By representing some of the vehicles in aggregate, mesoscopic and macroscopic modeling approaches reduce simulation running times at the cost of reduced accuracy of the results [7]. In particular, when considering vehicles in aggregate, it becomes impossible to study interactions among individual vehicles in detail or to follow a vehicle's trajectory throughout an entire trip.

State fast-forwarding [1] (in the following referred to as *single-vehicle fast-forwarding*) has been proposed as an approach to accelerate microscopic traffic simulations while still retaining the microscopic nature of the simulation. Fast-forwarding determines so-called independence intervals during which individual vehicles are guaranteed not to interact with other vehicles. The mobility of such isolated vehicles can be predicted accurately at low computational cost, which allows the vehicles to be advanced to the point in simulated time at which the earliest interaction with other vehicles may occur, avoiding intermediate computations. This allows common and established models originally formulated in a time-driven manner to be executed in a partially event-driven mode.

Even though single-vehicle fast-forwarding achieves considerable running time reductions in simulations of sparse traffic, more congested scenarios allow for only modest reductions. In the present paper, we extend the fast-forwarding approach to clusters of multiple vehicles (*cluster fast-forwarding*). While this extension seems natural, the vehicle interactions within clusters do not permit the straightforward numerical solution of the underlying models used by single-vehicle fast-forwarding. Instead, cluster fast-forwarding relies on artificial neural networks to predict the lanes, positions, and velocities of vehicles in a cluster after a number of time steps. In contrast to existing hybrid microscopic-macroscopic approaches, predictions are only applied in situations where interactions with vehicles outside a cluster have been ruled out, which allows for accurate predictions. The contributions of this paper are as follows:

(1) We introduce the approach of cluster fast-forwarding and describe the constraints under which clusters can be advanced into the simulated future.
(2) We explore the hyperparameter space of artificial neural networks to predict future vehicle states, considering the trade-off between accuracy and cost.
(3) We evaluate the performance gains of cluster fast-forwarding over time-driven microscopic simulation and single-vehicle fast-forwarding on a grid road network.

2 RELATED WORK

We consider time-driven microscopic traffic simulations in which vehicles are advanced on a road network by state updates carried out at fixed increments of simulated time. Commonly, two types of models are applied: a car-following model determines a vehicle's longitudinal movement based on its own velocity and displacement as compared to the vehicle ahead, while a lane-changing model

SIGSIM-PADS '20, June 15–17, 2020, Miami, FL, USA

ACM ISBN 978-1-4503-7592-4/20/06…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

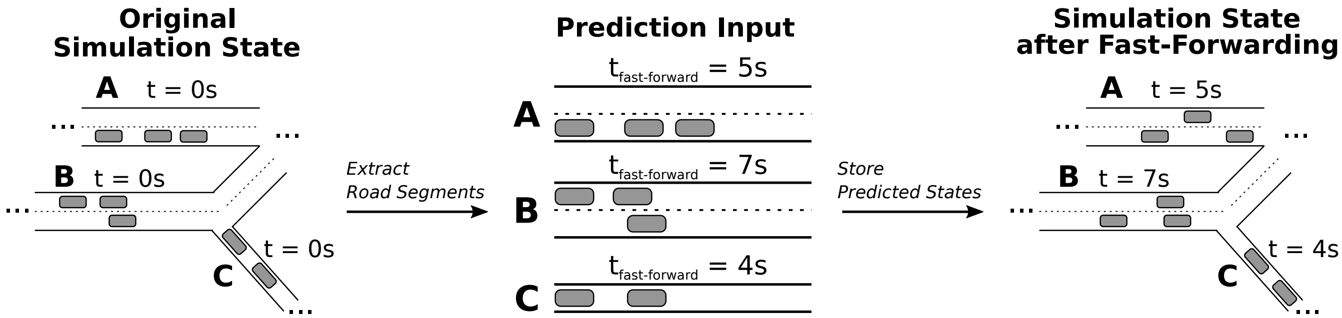

Figure 1: Overview of cluster fast-forwarding. Periodically, road segments that can be fast-forwarded to a future point in time are extracted from the road network. Predictions of the vehicles' future states are generated by neural networks and stored as part of the new simulation state.

determines the lateral movement depending on the space on the current and adjacent lanes and the expected acceleration of the current vehicle and its neighbors. The experiments presented in Section 4 will rely on the Intelligent Driver Model [9] for car-following and the MOBIL model [5] for lane-changing.

Hybrid microscopic-macroscopic modeling approaches apply detailed models to regions of interest in the simulation space, whereas other regions are simulated at a more abstract level to reduce the computational load (e.g., [2, 3]). The main challenge is the accuracy of the aggregation and disaggregation of simulated entities when crossing regions modeled at different levels of detail. Thus, the performance gains may come at the cost of substantial deviations compared to a purely microscopic reference simulation. Our work differs from existing hybrid approaches in the lack of a need for explicit aggregation and disaggregation. Further, only isolated clusters are fast-forwarded, enabling low deviations from the results of a reference simulation.

A variety of works have applied artificial neural networks to predict vehicle mobility based on sensor data such as video footage (e.g., [4, 6, 8]). While these works aim to predict future traffic situations directly from data, our objective is to closely reproduce the results generated by specific models instead of empirical data. The need to balance prediction accuracy and cost puts our focus on the dimensioning and parametrization of the artificial neural networks.

3 CLUSTER FAST-FORWARDING

The single-vehicle fast-forwarding approach proposed in our previous work exploits properties of common microscopic traffic models. Car-following models are frequently defined by differential equations expressing the relationship between a vehicle's velocity and distance to a leading vehicle and its acceleration. If a vehicle is isolated on the road network, the lack of coupling permits a straightforward and computationally inexpensive solution of the car-following equation. Further, the lane-changing behavior of isolated vehicles is trivially predictable. Simulation running times are reduced by efficiently advancing isolated vehicles multiple time steps into the simulated future.

In contrast, simulating a chain of multiple vehicles requires the solving of a coupled differential equation. To do so, common traffic simulators iteratively apply numerical integration over discretized simulation time. This time-driven approach also accommodates lane-changing decisions applied at discrete points in simulation time. Given a reference simulation of this type, advancing a cluster of vehicles from time t to $t + n$ requires a prediction of the coupled car-following behavior and all lane-changes that may have occurred in the meantime. Cluster fast-forwarding carries out these predictions using artificial neural networks trained on vehicle state data generated from large numbers of time-driven simulations. Both the input and output of each prediction is a set of state variables reflecting the vehicles' lane indices, displacements, and velocities at simulation times t and $t + n$. To limit the predictions to sufficiently small numbers of vehicles so that accurate predictions are still possible, we determine intervals where the considered vehicle clusters are isolated.

Figure 1 illustrates one iteration of fast-forwarding at simulation runtime: for each road segment r, a scanning step determines the earliest point in time t_{sense} at which a vehicle on another road segment may sense r, as well as the earliest time t_{next} at which a vehicle on r senses the next road on its route. A simple approach to determine t_{sense} and t_{next} is to project the future movement of vehicles based on their maximum velocity according to the road segments' speed limits. Now, $t_{\text{fast-forward}} := \min(t_{\text{sense}}, t_{\text{next}})$ is a lower bound on the time at which vehicles on the current road segment may interact with others. Thus, in the fast-forwarding step, the vehicles can be advanced to at most $t_{\text{fast-forward}}$. The approach is parametrized with the period at which scanning and fast-forwarding is attempted, and the maximum amount of time by which vehicles may be advanced.

4 EXPERIMENTS

4.1 Neural Network Configuration

Cluster fast-forwarding relies on artificial neural networks to predict the future states of small numbers of vehicles on a road segment. However, a substantial number of road segments of varying speed limits, numbers of lanes, and sizes of independence intervals may be considered at the same time. To achieve an overall performance benefit, the prediction cost must be as low as possible, while still maintaining sufficient accuracy to not invalidate the simulation results. In this section, we explore a variety of neural network configurations in light of this trade-off.

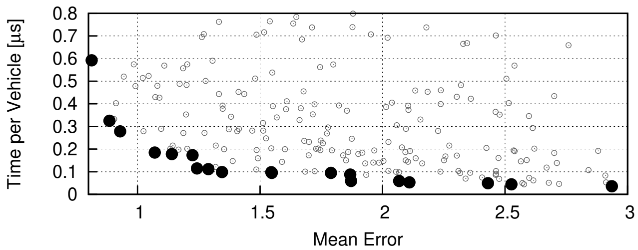

Figure 2: Scatter plot of prediction errors and per-vehicle prediction running times for different neural network configurations. Black points indicate the Pareto frontier of best-performing configurations.

The neural network training was carried out using the Python bindings provided by PyTorch using initial and final states of 10^6 simulations on road segments with randomly generated initial states. As cluster fast-forwarding is applied to individual road segments only, each of the simulations executed to generate training data simulate vehicles on a single road segment comprised of one or more lanes. A 9:1 split was used to form the training and test sets. After preliminary experiments, we chose the Adam optimizer for training. The relative reduction of the mean squared error was evaluated every 100 epochs, up to a maximum of 100 000 epochs. Once the relative reduction remained below 10^{-5} for more than 500 epochs, the training was terminated. All running time measurements were executed on an Intel Core i5-7400 CPU with 16 GiB of RAM, using the C++ bindings provided by the LibTorch library.

We explore the following parameter values for road segments: number of lanes $\in \{1, 2, .., 5\}$, number of vehicles $\in \{1, 2, .., 5\}$, speed limit $\in \{10, 20, .., 50\}$m/s, size of independence interval $\in \{10, 20, .., 300\}s$. For the vehicle counts and speed limits, we experiment with two training approaches: a) training a separate neural network for each possible parameter value, and b) training a joint network that receives the parameter value as an input. Similarly, we train separate and joint networks for car-following and lane-changing. A separate neural network is trained for each lane count.

Our experiments cover feedforward networks of the following configurations: total number of neurons in the hidden layers $\in \{30, 60, 120, 240, 480\}$, hidden layers $\in \{2, 3\}$, activation functions: Sigmoid, Tanh, ReLU, Leaky ReLU.

In Figure 2, we relate the prediction error to the average prediction time per vehicle. The error is determined as the geometric mean of the absolute errors in the predicted lane number $\in \{0, 1, ..\}$, the velocity in m/s, and the displacement on the lane in m. We also indicate the Pareto frontier of best-performing configurations. Table 1 shows the five Pareto-optimal configurations with the lowest mean errors, listing the type and total number of neurons in the hidden layers, number of hidden layers, and joint or separate networks for different models, numbers of vehicles, and speed limits. The results show the trade-off between accuracy and cost: separate networks for individual parameter values tend to provide higher accuracy at the cost of increased prediction times. The Pareto frontier contains only one configuration where a single neural network covers both models, all numbers of vehicles, and all speed limits. This network

Table 1: The five Pareto-optimal neural network configurations associated with the lowest mean error.

Neurons	Hidd. layers	Models	Veh.	Speed Lim.	Error	Time [μs]
240 Sigmoid	2	sep.	sep.	sep.	0.81	0.59
120 TanH	2	sep.	sep.	sep.	0.89	0.33
60 TanH	2	sep.	sep.	sep.	0.93	0.28
120 Sigmoid	2	sep.	sep.	joint	1.07	0.18
120 TanH	2	joint	sep.	sep.	1.14	0.18

Table 2: Road network simulation performance with purely time-driven execut. and fast-forwarding (FF).

#Vehicles, Gen. rate	Running time, Speedup		
	Time-driven	Single-vehicle FF	Cluster FF
10 000, 1	40.7s	33.0s, 1.23	20.5s, 1.99
10 000, 16	48.6s	42.3s, 1.15	26.7s, 1.82
20 000, 1	91.9s	76.0s, 1.21	44.4s, 2.07
20 000, 16	117.6s	104.4s, 1.13	77.3s, 1.52
40 000, 1	196.1s	163.4s, 1.20	91.7s, 2.14
40 000, 16	238.5s	223.8s, 1.07	207.6s, 1.15

achieved a comparatively large mean error of 2.94, but a low prediction time of 0.036μs. As the five configurations with the lowest error rely on only two layers and at most 240 neurons, we see that larger networks do not necessarily provide higher accuracy.

For ease of implementation of the road network experiments presented in the next section, we chose the configuration shown in the last row of Table 1, which considers car-following and lane-changing jointly.

4.2 Road Network Simulations

To determine the performance benefits of cluster fast-forwarding, we performed traffic simulations using a simplified variant of the city-scale microscopic traffic simulator CityMoS [12]. All simulations and predictions were executed sequentially, making use of a single core of an Intel Core i5-7400 CPU with 16 GiB of RAM.

We consider a grid network comprised of 64x64 four-way intersections connected by roads 300m in length, each with 4 lanes per direction. When including the short road segments on the intersections, the average road segment length is 87.0m. We vary the total number of vehicles and the number of vehicles generated per time step. Vehicles follow the shortest path between origin-destination pairs drawn uniformly at random from the links of the road network. Each run terminates after 10 hours of simulated time, or once all vehicles have reached their destination. The time step size is 0.1s. We configured the scanning period and horizon to values in $\{2, 4, 8, 16, 32\}$s and $\{2, 4, 8, 16, 30\}$s of simulated time, omitting combinations where the scanning period is smaller than the horizon. We show averages across 3 runs each for the best-performing parameter combinations. As shown in our previous work, auto-tuning could be applied to choose parameters at runtime.

Table 2 lists the running time and speedup over purely time-driven runs. The peak number of vehicles varied between 6 241 and 38 465. Generally, cluster fast-forwarding achieves higher speedup than single-vehicle fast-forwarding, particularly at large vehicle

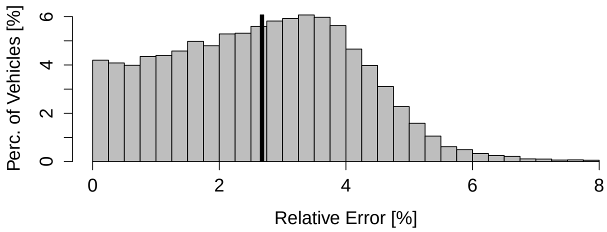

Figure 3: Histogram of the relative percentage error in the vehicles' travel times with cluster fast-forwarding compared to purely time-driven simulation.

counts. Cluster fast-forwarding benefits from more frequent opportunities to fast-forward agents, which can be measured by the total number of time-driven vehicle state updates avoided throughout a simulation run. For instance, with 40 000 vehicles and a vehicle generation rate of 1, the purely time-driven reference simulation carried out 2.61×10^8 state updates. Single-link fast-forwarding reduced the number of updates by 13.7%, while cluster fast-forward reduced the number by 49.8%.

To evaluate the effect of the cluster fast-forwarding on the fidelity of the simulation results, Figure 3 shows a histogram of the relative percentage error in the individual vehicles' travel times for a total number of 40 000 vehicles and a generation rate of 1. The average travel time in the time-driven simulation was 646.82s. The vertical line indicates the mean relative percentage error of 2.67%. The 99% quantile is 6.46%. Single-vehicle fast-forwarding entailed a lower relative error of 0.02%, but as shown above, achieved much lower performance than cluster fast-forwarding.

5 CONCLUSIONS

We proposed cluster fast-forwarding as a method to accelerate microscopic traffic simulations using neural network-based predictions of vehicle states on non-interacting road segments. Experiments with different neural network configurations show the trade-off between prediction accuracy and running time. Further, our simulation experiments show that cluster fast-forwarding successfully improves on the limited performance gains of single-vehicle fast-forwarding for higher traffic densities. The resulting reduction in simulation running times comes at the cost of a slight increase in the deviation from the results of a purely time-driven simulation.

The cluster fast-forwarding approach is amenable to parallelization among road segments without affecting the results. Parallel traffic simulations on hardware accelerators [10, 11] typically hold the simulation state in accelerator memory, which may enable hardware acceleration of the predictions without introducing additional data transfers.

ACKNOWLEDGEMENTS

This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

REFERENCES

[1] Philipp Andelfinger, Yadong Xu, Wentong Cai, David Eckhoff, and Alois Knoll. 2018. Fast-Forwarding Agent States to Accelerate Microscopic Traffic Simulations. In *Proceedings of the Conference on Principles of Advanced Discrete Simulation*. ACM, 113–124.

[2] Emmanuel Bourrel and Jean-Baptiste Lesort. 2003. Mixing Microscopic and Macroscopic Representations of Traffic Flow: Hybrid Model Based on Lighthill–Whitham–Richards Theory. *Transportation Research Record* 1852, 1 (2003), 193–200.

[3] Wilco Burghout, Haris Koutsopoulos, and Ingmar Andreasson. 2005. Hybrid Mesoscopic-Microscopic Traffic Simulation. *Transportation Research Record: Journal of the Transportation Research Board* 1934 (2005), 218–255.

[4] Xiaomo Jiang and Hojjat Adeli. 2005. Dynamic Wavelet Neural Network Model for Traffic Flow Forecasting. *Journal of Transportation Engineering* 131, 10 (2005), 771–779.

[5] Arne Kesting, Martin Treiber, and Dirk Helbing. 2007. General Lane-changing Model MOBIL for Car-following Models. *Transportation Research Record: Journal of the Transportation Research Board* (2007), 86–94.

[6] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[7] Kai Nagel and A Schleicher. 1994. Microscopic Traffic Modeling on Parallel High Performance Computers. *Parallel Comput.* 20, 1 (1994), 125–146.

[8] Yongxue Tian and Li Pan. 2015. Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network. In *International Conference on Smart City/SocialCom/SustainCom*. IEEE, 153–158.

[9] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. 2000. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E* 62, 2 (February 2000), 1805–1824.

[10] Jiajian Xiao, Philipp Andelfinger, David Eckhoff, Wentong Cai, and Alois Knoll. 2018. Exploring Execution Schemes for Agent-Based Traffic Simulation on Heterogeneous Hardware. In *Proceedings of the International Symposium on Distributed Simulation and Real Time Applications*. IEEE, 1–10.

[11] Jiajian Xiao, Philipp Andelfinger, David Eckhoff, Wentong Cai, and Alois Knoll. 2019. A Survey on Agent-based Simulation Using Hardware Accelerators. *ACM Comput. Surv.* 51, 6 (2019), 131:1–131:35.

[12] Daniel Zehe, Suraj Nair, Alois Knoll, and David Eckhoff. 2017. Towards CityMoS: A Coupled City-Scale Mobility Simulation Framework. *5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication* (2017), 26.