SUMMIT: A Simulator for Urban Driving in Massive Mixed Traffic

Panpan Cai*, Yiyuan Lee*, Yuanfu Luo, David Hsu

(*a*) Singapore-Highway (*b*) Magic-Roundabout (*c*) Meskel-Intersection

(*d*) Singapore-Highway in SUMMIT (*e*) Magic-Roundabout in SUMMIT (*f*) Meskel-Intersection in SUMMIT

Fig. 1: Benchmark scenes in the real world and corresponding scenes in SUMMIT.

***Abstract*— Autonomous driving in an unregulated urban crowd is an outstanding challenge, especially, in the presence of many aggressive, high-speed traffic participants. This paper presents SUMMIT, a high-fidelity simulator that facilitates the development and testing of crowd-driving algorithms. By leveraging the open-source OpenStreetMap map database and a heterogeneous multi-agent motion prediction model developed in our earlier work, SUMMIT simulates dense, unregulated urban traffic for heterogeneous agents at any worldwide locations that OpenStreetMap supports. SUMMIT is built as an extension of CARLA and inherits from it the physical and visual realism for autonomous driving simulation. SUMMIT supports a wide range of applications, including perception, vehicle control and planning, end-to-end learning. We provide a context-aware planner together with benchmark scenarios and show that SUMMIT generates complex, realistic traffic behaviors in challenging crowd-driving settings.**

I. Introduction

The vision of using autonomous driving to improve the safety and convenience of our daily life is coming closer. However, driving in *unregulated*, crowded urban environments, like in uncontrolled roads or unsignalised intersections in less-developed countries (Fig. 1), remains an open problem. Human participants can be fairly aggressive in these scenarios. One may disregard or be unaware of traffic rules, leading to behaviors like close following, inappropriate overtaking, illegal turning and crossing, etc. The road condition can become highly chaotic when involving many participants. Technical challenges for driving in unregulated urban crowds come from the complexity of both crowd behaviors and map environments. Traffic agents can be significantly different from each other. Cars, buses, bicycles, and motorcycles have different geometry, kinematics, and dynamics. Human participants also have different behavioral types - being conservative or aggressive, attentive or distracted, etc. In terms of the map environment, urban roads can have complex and versatile layouts: multi-lane roads, intersections, roundabouts, etc. Road structures significantly influence the motion of traffic agents and thus generate very different crowd behaviors in different locations. Such environments raise enormous difficulties for perception, control, planning, and decision-making of robot vehicles.

High-quality data for developing, training, and testing crowd-driving algorithms are, however, difficult and expensive to acquire due to the cost of devices, regulations and safety constraints. Although there are publicly available data

*The authors contributed equally.

The authors are with School of Computing, National University of Singapore, 117417 Singapore. {caipp, leeyiyuan, yuanfu, dyhsu}@comp.nus.edu.sg.

TABLE I: Comparison between SUMMIT and existing driving simulators.

Simulator	Real -world Maps	Unregulated behaviors	Dense Traffic [1]	Realistic Visuals & Sensors
SimMobilityST [5]	✓	×	✓	×
SUMO [6]	✓	×	✓	×
TORCS [7]	×	✓	×	✓
Apollo [8]	×	×	×	✓
Sim4CV [9]	×	×	×	✓
GTAV [10]	×	×	×	✓
CARLA [4]	×	×	×	✓
AutonoViSim [11]	×	✓	✓	✓
Force-based simulator [12]	×	×	✓	✓
SUMMIT (ours)	✓	✓	✓	✓

sets like KITTI [1], BDD100K [2], Oxford RobotCar [3], etc., that provide real-world driving data with rich sensor inputs, these data are not *interactive*, i.e., one cannot model the reactions of exo-agents to the robot's decisions. Driving simulators offer the capability of generating a virtually unlimited amount of interactive driving data. However, existing driving simulators do not capture the full complexity of unregulated urban crowds such as complex road structures and traffic behaviors, and are thus insufficient for testing or training robust driving algorithms. We aim to fill this gap.

We develop a new simulator, SUMMIT, that generates high-fidelity, interactive data for unregulated, dense urban traffic on complex real-world maps. SUMMIT uses real-world maps fetched from online sources to provide a virtually unlimited source of complex environments. Given arbitrary locations, the simulator automatically generates crowds of heterogeneous traffic agents with sophisticated, unregulated behaviors. The simulator leverages road contexts of real-world maps to guide the behaviours of traffic agents topologically and geometrically in order to construct realistic traffic conditions. We implemented SUMMIT based on CARLA [4] to leverage the high-fidelity physics, rendering, ans sensors. Through a python-based API, SUMMIT reveals rich sensor data, semantic information, and road contexts to external algorithms, enabling the application in a wide range of fields such as perception, vehicle control and planning, end-to-end learning, etc. We provide both qualitative and quantitative results to show that SUMMIT can generate complex, realistic mixed traffic in real-world urban environments. We also developed a built-in context-aware planner as a reference for future crowd-driving algorithms. The code of SUMMIT will be released open-source upon acceptance of the paper.

II. Related Work

A. Existing Driving Simulators

Driving simulators have been extensively applied to boost the development of autonomous driving systems. Recent simulators (Table I) have brought realistic visuals and sensors, but do not capture the complexities of urban environments and unregulated traffic behaviors.

Multi-car simulators like TORCS [7], [13], [14] focus on interactions between multiple robot-vehicles. These simulators suit the study of complex interactions between agents, but can hardly scale up to crowded urban scenes. CARLA [4], Sim4CV [9], and GTA [10] explicitly feature detailed physics modeling and realistic rendering for end-to-end learning. CARLA also provides a rich set of sensors such as cameras, Lidar, depth cameras, semantic segmentation, etc. However, these simulators rely on predefined maps, limiting the variety of environments. The simulated traffic also have relatively low density and simple rule-based behaviors. Another class of simulators [6], [5], [11], [12] feature traffic simulation and control in urban environments. Among them, SUMO [6] and SimMobilityST [5] support real-world maps but use simple rule-based behaviors, while another class [11], [12] apply more sophisticated motion models but are restricted to predefined maps. We aim to model the complexities in both urban maps and traffic behaviors in an automatic and unified framework.

B. Crowd Simulation Algorithms

Existing crowd simulation algorithms, e.g., social force and velocity obstacles, can in principle be applied to generate crowd behaviors in urban environments. Social force [15], [16], [17], [18] assume that traffic-agents are driven by attractive forces exerted by the destination and repulsive forces exerted by obstacles. Social force can simulate large crowds, but the quality of interactions are constrained by model simplicity. Velocity Obstacle (VO) [19] and Reciprocal Velocity Obstacle (RVO) [20], [21], [22] compute collision free motion by optimizing in the feasible velocity space. Variants such as GVO [23], NH-ORCA [24], B-ORCA [25], PORCA [26] explicitly handle non-holonomic traffic agents. Some variants model behavioral types of crowd agents such as patience [26] and attention [27]. A recent model GAMMA [28] can simulate heterogeneous traffic agents with different geometry, kinematics, and behavioral types in a unified, velocity-space framework. The behavior model in SUMMIT extends the framework of GAMMA to encode topological road contexts such as lanes and pedestrian sidewalks to closely represent real-world scenarios.

III. SUMMIT Simulator

SUMMIT focuses on simulating complex unregulated behaviours of dense urban traffic in complex real-world maps. It is designed for generating high-fidelity interactive data to facilitate the development, training, and testing of crowd-driving algorithms. SUMMIT automatically generates massive mixed traffic using topological road contexts and optimization-based unregulated crowd behaviors. SUMMIT fetches real-world maps from the OpenStreetMap [29], and constructs two topological graphs: a lane network for vehicles, and a sidewalk network for pedestrians. These networks

[1] We only check-mark simulators explicitly featuring crowd behaviours.

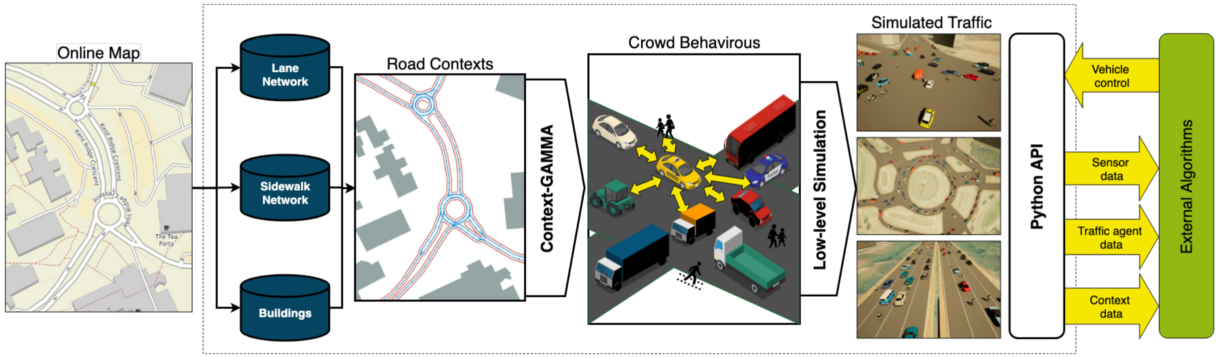


Fig. 2: An overview of SUMMIT that simulates massive mixed traffic at any location in the world.

form a representation of the road contexts. Then, our behavior model, Context-GAMMA, takes road contexts as input to guide the traffic behaviors geometrically and topologically. In the micro-scope, Context-GAMMA uses velocity-space optimization to generate sophisticated, unregulated crowd behaviors. The low-level structure of SUMMIT is based on CARLA, retaining its desirable features such as high-fidelity physics, realistic rendering, weather control, and rich sensors. Fig. 2 provides an overview of SUMMIT.

A. Representing Real-world Maps

1) Lane Network: A lane network in SUMMIT defines the connectivity of the road structure at the fidelity of individual lanes. The network consists of directed lane segments and connections between them. SUMMIT relies on SUMO [6] to automatically convert OSM maps to lane networks. The extensive suite of network editing tools provided by SUMO can also be leveraged to improve and customize maps. The lane network interface allows users to locate traffic agents on the lane network and retrieve connected lane segments. The interface closely follows CARLA's waypoint interface, so that CARLA users can easily adapt to it.

2) Sidewalk Network: A sidewalk network in SUMMIT defines the behaviors of pedestrians, which usually walk along road edges and occasionally cross roads. The network contains sidewalks near road edges defined as poly-lines and connections between sidewalks defined as cross-able roads. The sidewalk poly-lines are extracted from the geometry of roads. Similar to the lane network, the sidewalk network interface allows users to locate pedestrians on the network and retrieve the opposite sidewalk for road-crossing.

3) Occupancy Map Interface: We additionally provide an occupancy map interface to expose drive-able regions for the ego-vehicle. An occupancy map is the top-down projection of the road geometry, aligned with the ego-vehicle's location and heading direction. It can be used either for collision checking in control and planning algorithms or as bird-view input to neural networks.

4) Landmarks: SUMMIT also makes use of landmark data in OSM maps such as buildings and forests to provide structurally rich and realistic visuals. We additionally support randomization of the landmark textures to generate more versatile visual inputs and enable techniques such as domain randomization [30].

B. Crowd behavior Modelling

SUMMIT uses Context-GAMMA, a context-aware crowd behavior model, to generate sophisticated interactive behaviors of traffic agents. Context-GAMMA extends GAMMA [28] to incorporate road contexts and models them as constraints in velocity space. For completeness, we briefly introduce GAMMA, and present the extensions in Context-GAMMA.

GAMMA formulates the motion of traffic agents as constrained geometric optimization in velocity space. It assumes that each traffic agent optimizes its velocity based on the navigation goal, while being constrained by kinematic constraints (e.g. non-holonomic motion of car) and geometric constraints (collision avoidance with nearby agents). For a given agent A, let K_A represent the set of velocities that satisfy kinematic constraints and G_A^τ represent the set of velocities that satisfy geometric constraints for at least τ time. Then GAMMA selects for A a new velocity from their intersection:

$$v_A^{\mathrm{new}} = \operatorname*{arg\,min}_{v \in \mathrm{G}_A^\tau \cap \mathrm{K}_A} \left\| v - v_A^{\mathrm{pref}} \right\|, \tag{1}$$

where v_A^{pref} is A' preferred velocity computed from its goal. When computing K_A and G_A^τ, GAMMA also takes into account responsibility and attention of the agent, to generate more human-like motions. We refer readers to [28] for more details of the construction of K_A and G_A^τ. Geometrically, K_A is a convex velocity set and G_A^τ is the intersection of velocity-space half planes. See Fig. 3*a* for an example of K_A and G_A^τ.

GAMMA handles heterogeneous traffic agents with different kinematics and geometry in a unified velocity-space framework. It has been demonstrated with real-world data sets [28] to generate realistic behaviors for traffic agents.

However, as GAMMA does not make explicit use of road contexts, it often fails to generate realistic simulation for complex urban roads. GAMMA agents can aggressively head towards their goals and be trapped by the complex road structure. In the real-world, road contexts can effectively guide and constrain traffic agents' behaviors: vehicles tend

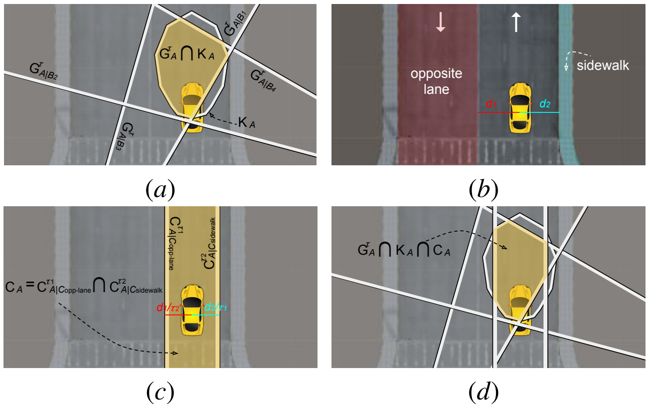


Fig. 3: Crowd behaviour modelling with Context-GAMMA: (*a*) the feasible velocity space (yellow) of an agent A as the intersection of kinematic constraints K_A and geometry constraints G_A^τ. (*b*) A car constrained by the opposite lane and the sidewalk. (*c*) The corresponding contextual constraints $\mathrm{C}^{\tau_1}_{A|C_{\text{opp-lane}}}$ and $\mathrm{C}^{\tau_2}_{A|C_{\text{sidewalk}}}$. (*d*) The augmented feasible velocity space.

to follow particular lanes when the road is clear and avoid to drive along the wrong direction. Moreover, as traffic agents are heterogeneous, they are affected by different sets of static obstacles: pedestrians consider sidewalks as open spaces, but vehicles consider them as obstacles.

Our new model, Context-GAMMA, provides a general way to embed road contexts as objectives and constraints in velocity space. Context-GAMMA extracts the preferred velocity v^{pref} of traffic agents from the lane and sidewalk networks. To diversify agent behaviors, it randomly selects a lane from all feasible lanes ahead of the agent, and points v^{pref} to a look-ahead waypoint along the selected lane.

Context-GAMMA can also model traffic rules by casting contextual constraints, e.g., no wrong-direction driving, into half planes in agents' velocity space, forcing them to select velocities complying with the road rule. Denote the intersection of all the contextual half planes for an agent A as C_A. Context-GAMMA optimizes the agent velocity in the augmented feasible velocity space $\mathrm{K}_A \cap \mathrm{G}_A^\tau \cap \mathrm{C}_A$. Note that $\mathrm{K}_A \cap \mathrm{G}_A^\tau \cap \mathrm{C}_A$ is also convex by construction and the objective function Eqn. (1) is quadratic. Therefore, the optimization problem can be efficiently solved using linear programming.

Fig. 3*b* and Fig. 3*c* show an example of contextual constraints. To prevent car A in Fig. 3*b* from driving to the opposite lane within a time window τ_1, the lateral speed of the car should be constrained under d_1/τ_1, where d_1 is the distance from the car to the opposite lane. This constraint forms a half-plane in the velocity space, $\mathrm{C}^{\tau_1}_{A|C_{\text{opp-lane}}}$, defined by a separation line parallel to the opposite lane with an offset of d_1/τ_1 from the origin. Any velocity in $\mathrm{C}^{\tau_1}_{A|C_{\text{opp-lane}}}$ would be feasible. Similarly, collision avoidance with the sidewalk can result in another half-plane $\mathrm{C}^{\tau_2}_{A|C_{\text{sidewalk}}}$ in the velocity space with an offset of d_2 from the origin. The intersection of the two half-planes forms A's contextual constraint, C_A (Fig. 3*c*), which is further imposed on $\mathrm{K}_A \cap \mathrm{G}_A$ to form the feasible space of A (Fig. 3*d*).

C. *Interfaces*

The Python API of SUMMIT extends that of CARLA, exposing to external algorithms not only sensor data and agent states, but also road contexts like lane networks, sidewalk networks, and map occupancy grids. Algorithms can also send vehicle control back to the simulation including steering, acceleration, braking, reversing, etc.. We further provide ROS bridging for state, sensor, and occupancy information to communicate with common robotics systems. SUMMIT thus enables a wide range of applications such as perception, sensor-based control, model-based reasoning, and end-to-end learning.

IV. Context-aware POMDP Planning

Driving in an unregulated dense traffic is extremely challenging. The task is safety critical. No accident or crash should be raised by the robot. The robot has to be smart enough to make efficient progress, instead of being "frozen" and stuck in the highly dynamic crowd. Algorithms have to model a large-scale, highly dynamic, and interactive scene and plan for robot given the large number of participants that matters in decision-making. We suggest that algorithms should leverage the road contexts to help long-term planning. In the following section, we present a Context-POMDP planner that conditions the POMDP problem proposed in [26], [31] on road contexts to achieve real-time long-term planning.

Context-POMDP conditions the hidden states of agents on road contexts and uses these hidden states to constrain online planning. It consists of two components: a belief tracker that infers a joint belief over exo-agents' hidden states, and an online POMDP solver that takes the current belief and computes the optimal driving action. We will discuss these two components in the following sections.

A. *Belief Tracking*

The belief tracker maintains a joint belief over two dimensions of hidden states:

- The *intention* of the traffic agent: Let $U_i, i \in I_{exo}$ be the set of path candidates for the ith traffic agent extracted from the road contexts such as the lane network and the sidewalk network. This agent may take any of the path candidates in U_i as its actual intention.
- The *type* of a traffic agent: An agent can be either *distracted*, thus not interacting with the ego-vehicle, or be *attentive*, thus cooperatively avoid collision with the ego-vehicle.

The belief tracker is implemented as a factored histogram filter [32]. It maintains for each exo-agent a set of possible hidden state value pairs and the corresponding probability inferred from the interaction history. At each time step, it uses a motion model (Section IV-B.3) to compute the likelihood of transitions and observations, and updates the posterior belief using the Bayes rule.

B. *Context-POMDP*

The Context-POMDP planner solves a context-conditioned POMDP model to compute actions for the exo-vehicle.

1) State and Observation Modelling: A state in Context-POMDP includes both discrete-domain variables and continuous-domain variables:

- State of the ego-vehicle, $s_c = (x, y, \phi, \mu)$, including the position (x, y), heading direction ϕ, and the intended driving path μ.
- Observable states of exo-agents, $\{s_i = (x, y, \vec{v})\}_{i \in I_{exo}}$, including the position (x, y) and the current velocity $\vec{v}$. I_{exo} defines the set of indices of exo-agents.
- Hidden states of exo-agents, $\{\theta_i = (t_i, \mu_i)\}_{i \in I_{exo}}$, including the type and the (sampled) intended path of the ith traffic agent.

We assume that the ego-vehicle can observe its own state and discretized values of the observable states of exo-agents. The hidden states of exo-agents can only be inferred and modelled with beliefs.

2) Action Modelling: The action space of the ego-vehicle consists of its steering angle and acceleration. Given the well-known exponential complexity of POMDP planning [33], Context-POMDP decouples the action space of the ego-vehicle to keep the branching factor of the planning problem within a tractable range. Concretely, we restrict the POMDP to compute the acceleration along the intended path, while the steering angle is generated using a pure-pursuit algorithm [34]. The action space contains three possible accelerations for each time step: $\{ACC, MAINTAIN, DEC\}$. The acceleration value for ACC and DEC is $3\,\mathrm{m/s^2}$ and $-3\,\mathrm{m/s^2}$, respectively. The maximum speed of the ego-vehicle is $6\,\mathrm{m/s}$.

3) Transition Modelling : Context-POMDP predicts traffic agents' motion using the following set of models. Distracted traffic agents are assumed to track their intended path with the current speed. Attentive traffic agents also tend to follow the sampled path, but use PORCA [26], an interactive collision avoidance model that is similar to GAMMA but considerably simpler, to generate the actual local motion. The motion of all agents, including the ego-vehicle, are constrained by their kinematics, e.g., pedestrians are simulated using holonomic motion and car-like vehicles are simulated using bicycle models. To model stochastic transitions of exo-agents, their motion are perturbed by Gaussian noises on the displacement.

4) Reward Modelling: The reward function in Context-POMDP takes in to account safety, efficiency, and smoothness of driving. It assigns large penalties when the ego-vehicle collides with any exo-agent, uses a motion cost to penalize driving at low speed, and finally, penalizes frequent deceleration. Details of the reward function can be found in [26].

5) Solving the Context-POMDP: The POMDP model is solved using a parallel online planner, HyP-DESPOT. We refer readers to [35] for details on the planner.

V. Results

We want to answer the following questions in the experiments:

- Can SUMMIT simulate realistic traffic on complex maps?
- How does SUMMIT scale with the density of traffic?
- What are the benefits brought by SUMMIT over rule-based models commonly used in simulators?
- Can Context-POMDP drive a vehicle safely and efficiently in dense unregulated urban traffic?

We provide both qualitative and quantitative results to answer these questions.

A. Simulation on Benchmark Scenarios

We designed three real-world benchmark scenarios to evaluate the performance of SUMMIT and the Context-POMDP planner.

1) Singapore-Highway (Fig. 1a): A highway in Singapore with multiple lanes. Traffic agents try to drive as fast as possible and thus conduct overtaking frequently.

2) Magic-Roundabout (Fig. 1b): A roundabout at Swindon, England with very complex layout. Traffic agents meet at the main roundabout and the accompanying intersections, having to coordinate with each other.

3) Meskel-Intersection (Fig. 1c): A complex intersection at the Meskel square, Addis Abeba. Traffic agents come from different directions and encounter at the intersection, all of them driving aggressively.

SUMMIT automatically constructs the scenarios from the open-access map data. It then automatically generates and simulates unregulated dense traffic on these maps. All scenarios contain 120 heterogeneous traffic agents driving or walking in the region of interest, each conducting aggressive and unregulated behaviors. Once an agent moves out of the region, we replace it with new agents inside the region to maintain high density of the traffic. Fig. 1(d-f) shows qualitative simulation results on the benchmark scenarios. Comparison with the real scenarios shows that the simulated traffic closely represent the reality. More simulation results can be found in the accompanying video or via `www.dropbox.com/s/c9a0lt2pl01wrpl`.

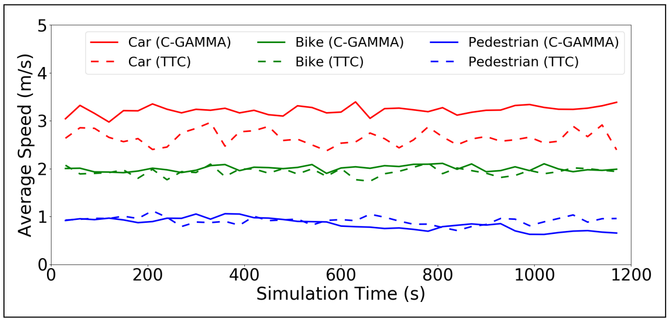
B. Comparison with Rule-based behavior Model

Context-GAMMA provides sophisticated interactive behaviors for dense traffic. To provide a quantitative view of this capability, we compare Context-GAMMA with a reactive model that moves agents along lane center-curves and uses time-to-collision (TTC) [36] to calculate the vehicle's speed. Performances of the two models are measured using the average speed of traffic agents and a congestion factor defined as the percentage of agents being jammed in the crowd. These jammed agents are removed from the crowd after being stationary for $5\,\mathrm{s}$.

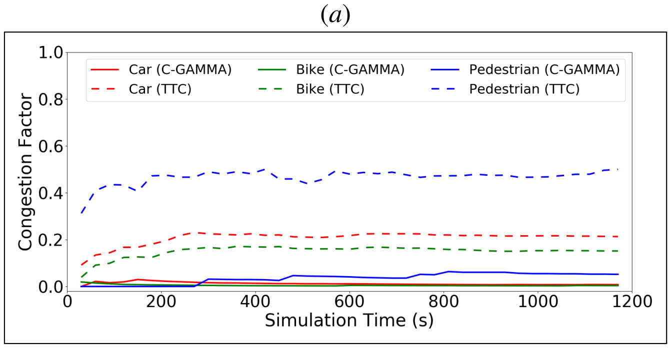
Table II shows the comparison on the benchmark scenarios. Agents controlled by Context-GAMMA generally drive much faster than the TTC-agents in all benchmark scenarios. The crowd controlled by Context-GAMMA also causes significantly lower traffic congestion throughout 20 minutes of simulation. This is because Context-GAMMA explicitly models cooperation between agents and provides

TABLE II: Comparison between crowd behaviour models on real-world scenarios (Fig. 1): Context-GAMMA vs. Time-To-Collision.

		Avg. Agent Speed (m/s)	Congestion Factor
Highway	Context-GAMMA	2.56	0.01
	TTC	2.16	0.11
Roundabout	Context-GAMMA	2.42	0.02
	TTC	2.17	0.27
Intersection	Context-GAMMA	2.67	0.01
	TTC	2.30	0.21



(*a*)



(*b*)

Fig. 4: Performance profile of Context-GAMMA and TTC on the Meskel-Intersection: (a) average speed of traffic agents; (b) congestion factor of the traffic.

TABLE III: Time performance and scalability of the simulation for different number of agents on a laptop with an i7-9750H CPU.

Number of Agents	150	200	250	300	350	400
Frequency (Hz)	22.3	21.5	21.2	19.7	15.5	11.1
Update Time (ms)	44.8	46.5	47.2	50.7	64.6	89.9

optimal collision avoidance motion using both steering and acceleration.

Fig. 4 shows a detailed profile of the agent speeds and the congestion factors for car-like agents, bicycle-like agents and pedestrians against the simulation time. Context-GAMMA consistently leads to higher agent speed for all agent types. The congestion factor of the TTC-controlled traffic grows quickly with the simulation time, indicating that agents fail to coordinate with each other. In contrast, the congestion factor brought by Context-GAMMA remains significantly lower for all types of agents.

TABLE IV: Comparison on the driving performance of planning algorithms: Context-POMDP vs. Roll-out.

		Collision Rate	Avg. Speed (m/s)	Deceleration Frequency
Highway	Context-POMDP	0.00089	2.05	0.087
	Roll-out	0.004	0.036	0.031
Roundabout	Context-POMDP	0.00037	1.66	0.105
	Roll-out	0.00027	0.14	0.059
Intersection	Context-POMDP	0.0	2.5	0.069
	Roll-out	0.0	0.12	0.066

C. Time Performance and Scalability

Context-GAMMA scales well to large crowds of traffic agents. As shown in Table III, the simulation runs at a high rates when modelling up to 400 agents, and the growth of computation time is almost linear until the map saturates with agents.

D. Performance of the Context-POMDP Planner

Driving in an unregulated dense crowd is a challenging, large-scale planning problem. As most traffic agents are aggressive and highly dynamic, long-term planning is required for driving safely, efficiently and smoothly. We validate this by comparing the driving performance of Context-POMDP with a Roll-out algorithm that rolls out a reactive controller and measures the future accumulative reward to choose an optimal action. The reactive controller accelerates the ego-vehicle when exo-agents in front are far way ($> 4\,\mathrm{m}$ away), maintains half-speed when they are in caution range ($2 \sim 4\,\mathrm{m}$ away), and decelerates when they are close-by ($< 2\,\mathrm{m}$ away). Table IV provides measurements of the collision rate per meter, the average vehicle speed, and the frequency of deceleration when driving the ego-vehicle using SUMMIT and Roll-out. Compared to Roll-out that can barely move, Context-POMDP can drive the vehicle through highly dynamic crowds with significantly higher speed while maintaining low collision rates and deceleration frequency. We thus conclude that sophisticated long-term planning is important for driving in unregulated traffic, and Context-POMDP establishes a reference for future crowd-driving algorithms.

VI. CONCLUSION

We presented SUMMIT, a simulator for generating high-fidelity interactive data for developing, training, and testing crowd-driving algorithms. The simulator uses online maps to automatically construct unregulated dense traffic at any location of the world. By integrating topological road contexts with an optimization-based crowd behavior model, SUMMIT can generate complex and realistic crowds that closely represent unregulated traffic in the real-world. We also provided context-POMDP as a reference planning algorithm for future developments. We envision that SUMMIT will support a wide range of applications such as perception, control, planning, and learning for driving in unregulated dense urban traffic.

REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[2] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[3] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[5] C. L. Azevedo, N. M. Deshmukh, B. Marimuthu, S. Oh, K. Marczuk, H. Soh, K. Basak, T. Toledo, L.-S. Peh, and M. E. Ben-Akiva, "Simmobility short-term: An integrated microscopic mobility simulator," *Transportation Research Record*, vol. 2622, no. 1, pp. 13–23, 2017.

[6] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, IEEE, 2018.

[7] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net*, vol. 4, no. 6, 2000.

[8] "Apollo simulation," *http://apollo.auto/platform/simulation.html*, 2018.

[9] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 902–919, 2018.

[10] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2213–2222, 2017.

[11] A. Best, S. Narang, L. Pasqualin, D. Barber, and D. Manocha, "Autonovi-sim: Autonomous vehicle simulation platform with weather, sensing, and traffic control," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1048–1056, 2018.

[12] Q. Chao, X. Jin, H.-W. Huang, S. Foong, L.-F. Yu, and S.-K. Yeung, "Force-based heterogeneous traffic simulation for autonomous vehicle testing," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8298–8304, IEEE, 2019.

[13] H. Zhao, A. Cui, S. A. Cullen, B. Paden, M. Laskey, and K. Goldberg, "Fluids: A first-order local urban intersection driving simulator," in *CASE.*, IEEE, 2018.

[14] M. Naumann, F. Poggenhans, M. Lauer, and C. Stiller, "Coincar-sim: An open-source simulation framework for cooperatively interacting automobiles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–6, IEEE, 2018.

[15] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, p. 4282, 1995.

[16] R. Löhner, "On the modeling of pedestrian motion," *Applied Mathematical Modelling*, vol. 34, pp. 366–382, 2010.

[17] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2013.

[18] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 261–268, IEEE, 2009.

[19] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robotics Research*, vol. 17, pp. 760–772, 1998.

[20] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. on Robotics & Automation*, 2008.

[21] J. Van Den Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Proc. Int. Symp. on Robotics Research*, 2009.

[22] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, pp. 696–706, 2011.

[23] D. Wilkie, J. Van Den Berg, and D. Manocha, "Generalized velocity obstacles," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pp. 5573–5578, IEEE, 2009.

[24] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*, pp. 203–216, Springer, 2013.

[25] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots," in *Proc. IEEE Int. Conf. on Robotics & Automation*, 2012.

[26] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "Porca: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3418–3425, 2018.

[27] E. Cheung, A. Bera, and D. Manocha, "Efficient and safe vehicle navigation based on driver behavior classification," in *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pp. 1024–1031, 2018.

[28] Y. Luo, P. Cai, D. Hsu, and W. S. Lee, "GAMMA: A general agent motion prediction model for autonomous driving," *arXiv preprint arXiv:1906.01566*, 2019.

[29] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ." `https://www.openstreetmap.org`, 2017.

[30] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *arXiv preprint arXiv:1808.00177*, 2018.

[31] P. Cai, Y. Luo, A. Saxena, D. Hsu, and W. S. Lee, "Lets-drive: Driving in a crowd by learning from tree search," 2019.

[32] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[33] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99 – 134, 1998.

[34] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," tech. rep., Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[35] P. Cai, Y. Luo, D. Hsu, and W. S. Lee, "HyP-DESPOT: A hybrid parallel algorithm for online planning under uncertainty," in *Proc. Robotics: Science & Systems*, 2018.

[36] M. Meghjani, Y. Luo, Q. H. Ho, P. Cai, S. Verma, D. Rus, and D. Hsu, "Context and intention aware planning for urban driving," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2019.