2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)
October 23-27, 2022, Kyoto, Japan

InterSim: Interactive Traffic Simulation via Explicit Relation Modeling

Qiao Sun[1], Xin Huang[2], Brian C. Williams[2], and Hang Zhao[1*]

Abstract— **Interactive traffic simulation is crucial to autonomous driving systems by enabling testing for planners in a more scalable and safe way compared to real-world road testing. Existing approaches learn an agent model from large-scale driving data to simulate realistic traffic scenarios, yet it remains an open question to produce consistent and diverse multi-agent interactive behaviors in crowded scenes. In this work, we present InterSim, an interactive traffic simulator for testing autonomous driving planners. Given a test plan trajectory from the ego agent, InterSim reasons about the interaction relations between the agents in the scene and generates realistic trajectories for each environment agent that are consistent with the relations. We train and validate our model on a large-scale interactive driving dataset. Experiment results show that InterSim achieves better simulation realism and reactivity in two simulation tasks compared to a state-of-the-art learning-based traffic simulator.**

I. Introduction

With the recent development of autonomous driving technologies, traffic simulation has played an important role in enabling testing the planner system on a large scale. Compared to real-world road testing, simulation offers a more time and resource efficient alternative by reconstructing rare but important traffic scenarios. More importantly, it allows simulating risky scenarios that are usually difficult to obtain in real-world driving.

Traditional simulators often rely on static log replay that simulates the agent behavior based on what happened in the collected data. It fails to produce reactive behavior of environment agents when the ego plan diverges from the original log and thus becomes less useful in interactive scenarios. On the other hand, heuristic-based models, such as the intelligent driver model (IDM) [1]–[3], produce more reactive behaviors in response to diverging ego plans, but they are limited to following predefined trajectories and have difficulties in producing diverse scenarios.

Recent advances in machine learning have enabled realistic and diverse agent simulation by training agent models from realistic driving data. They demonstrated great potential in supporting closed loop planner evaluation. For instance, [4] proposes to generate realistic driving episodes by leveraging a probabilistic prediction model given traffic observations and environmental context; [5] infers future agent states as both discrete intent and continuous controls conditioned on past observations over all agents. On the other hand, they focus on simulating trajectories of individual agents without reasoning about their future interactions, which could lead to colliding trajectories in dense traffic. To overcome this challenge, [6] adds a task loss to penalize collisions and [7] proposes a feasibility check on the generated trajectories to filter out colliding trajectories. Instead of requiring a hand-crafted loss or an ad-hoc filter, [8] offers simulation consistency by rolling outs joint trajectories over all the agents in a scene through an implicit latent variable learned by a conditional variational autoencoder; however, such generative models rely on probabilistic sampling and suffer from producing rare or dangerous scenarios, which are crucial to testing autonomous driving planners.

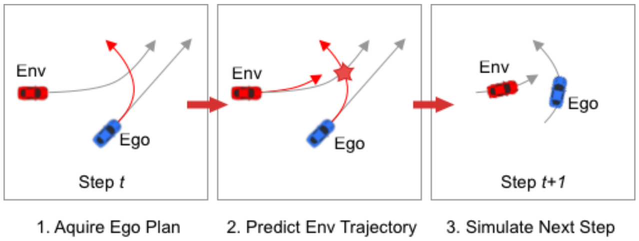

Fig. 1: A motivating example of InterSim. Given an updated ego plan in red, InterSim first identifies the relevant agent whose future trajectories may collide with the ego plan, and infers the interacting relations between the agents, such as the environment agent is yielding to the ego agent. It then predicts consistent and reactive trajectories for the relevant agent given the inferred relations. The simulator simulates the next step of all agents based on the predictions and waits for the new plan.

In this work, we propose InterSim, an interactive traffic simulation model that reasons about explicit interaction relations when rolling out future trajectories for all agents in a scene. As shown in Fig. 1, given an updated ego plan in red, InterSim first identifies the relevant agents whose future trajectory may collide with the new ego plan. Next, it infers the interaction relations between the ego agent and the relevant environment agents, and predicts reactive trajectories for the relevant agents based on the relation. Finally, it follows the predictions to simulate one step ahead for the environment agents and repeat the same procedure until the end of the simulation episode.

Compared to existing learning-based models, InterSim offers a few advantages by reasoning about the explicit interactive relations among the relevant traffic agents. First, the relations guide the trajectory simulator to produce consistent trajectories of multiple agents in complex interactive scenarios and offer better interpretability. Second, one can use it to manipulate an interactive scenario by specifying the

[1]IIIS, Tsinghua University alan.qiao.sun@gmail.com
[2]CSAIL, Massachusetts Institute of Technology
[*]Corresponding at hangzhao@mail.tsinghua.edu.cn

978-1-6654-7927-1/22/$31.00 ©2022 IEEE

interaction relation between agents, which is a non-trivial task for generative models that may require a large number of samples. Third, it affords better efficiency by identifying only the relevant agents influenced by the ego plans and modifying their future trajectories in simulation as opposed to all agents.

Our contribution is as follows:

- We propose a learning-based simulation model, InterSim, that rolls out realistic and consistent future trajectories of multiple traffic agents based on explicit interaction relations.
- We leverage a relation predictor to infer interaction relations for better interpretability and simulation efficiency, and show how our simulator can be used to manipulate different interaction situations by specifying the relations.
- We train and evaluate our model on the Waymo Open Motion Dataset, a publicly available real-world driving benchmark, and demonstrate its advantage compared to a state-of-the-art baseline in two simulation tasks.

II. Related Work

In this section, we discuss relevant literature in three aspects: traffic simulation, behavior prediction, and interaction modeling.

A. Traffic Simulation

Traffic simulation is an important task for intelligent transportation systems, allowing for training and evaluating driving models in a more scalable and safe way. Existing traffic simulators render high-fidelity driving environments in the context of racing [9] and urban driving [10], [11]. However, they often simulate agent behaviors through heuristic-based models that fail to cover diverse scenarios or interactions.

Recently, learning-based models have demonstrated great success in simulating realistic and reactive agent behaviors by learning driving patterns from real-world driving data. For instance, [4] trains a deep neural network through a rasterized representation derived from driving logs to simulate future agent trajectories; [5] infers future agent intent and control inputs to model stochastic traffic dynamics. While such methods consider the past trajectories of all the agents at once, they assume independence of future trajectory rollouts that may lead to inconsistent or colliding trajectories between simulated agents in interactive scenarios.

In order to improve simulation consistency over multiple interacting agents, [6] leverages a collision loss and [7] proposes a rule-based fall-back layer to discourage or avoid collisions. While such works often require hand-crafted losses or post-processing filters, [8] proposes a multi-agent behavior model that simulates joint agent behaviors directly through an implicit latent variable that governs the agent interactions. Compared to existing models, we propose a relation-aware simulator that simulates diverse and realistic interactive behaviors in a more straightforward and efficient way by explicitly modeling interacting relations.

B. Behavior Prediction

Behavior prediction offers a natural solution to simulate agent behaviors through the predicted trajectories given the environmental context. Recent models prove great success in improving prediction accuracy, by learning agent dynamics and environmental context represented either as a vector representation [12], [13] or a rasterized image [14], [15].

Due to uncertainty in human intent, the future trajectories are multi-modal. To handle the multi-modality and improve prediction coverage, a family of models are proposed to first predict high-level intent, such as goal targets [16]–[18], lanes to follow [19], [20], maneuvers [21]–[23], and linguistic descriptions [24], before predicting low-level trajectories that are conditioned on the intent.

In this work, we take advantage of the goal-conditioned models in the behavior prediction literature to simulate realistic agent trajectories given the environmental context and agent intent.

C. Interaction Modeling

Modeling interaction is an important task in motion prediction and simulation when reasoning about multi-agent behaviors. While many existing approaches [25]–[28] rely on implicit latent variables to model interactions, we focus on modeling and predicting explicit interaction relations in this work for better interpretability. These explicit relations allow us to produce and manipulate different types of interactive scenarios.

In this work, we follow [29]–[31] that define agent relations based on the pass and yield relationship and predict the relationship as a classification problem through a separate learning model. The predicted relations are useful in guiding the motion predictor to generate consistent trajectories among multiple agents, as shown by [31].

When there exist potential conflicts between a novel ego plan and the simulated trajectories of environment agents given the predicted relations, we adopt conflict resolution techniques that are widely used in planning [32], search [33], and ordering [34].

III. Problem Formulation

We formulate the problem of learning realistic interactive behaviors for traffic simulation following [8]. Given map states M and the observed states S of N traffic agents in a scene, the goal is to roll out the future states of all agents Y up to a finite horizon T.

Due to the computational complexity and memory constraint in simulating joint behaviors over all traffic agents in the scene, our model focuses on simulating agent behaviors that are *relevant* to the ego plan, as the irrelevant agent behaviors are often ignored by the ego planner. For an irrelevant agent whose future trajectory stays the same given a new ego plan at the next step, our simulator can simply roll out its future trajectory from the data.

One key consideration in our problem is to faithfully follow the agent's origin intent as much as possible. We define such intent based on the goal location collected from

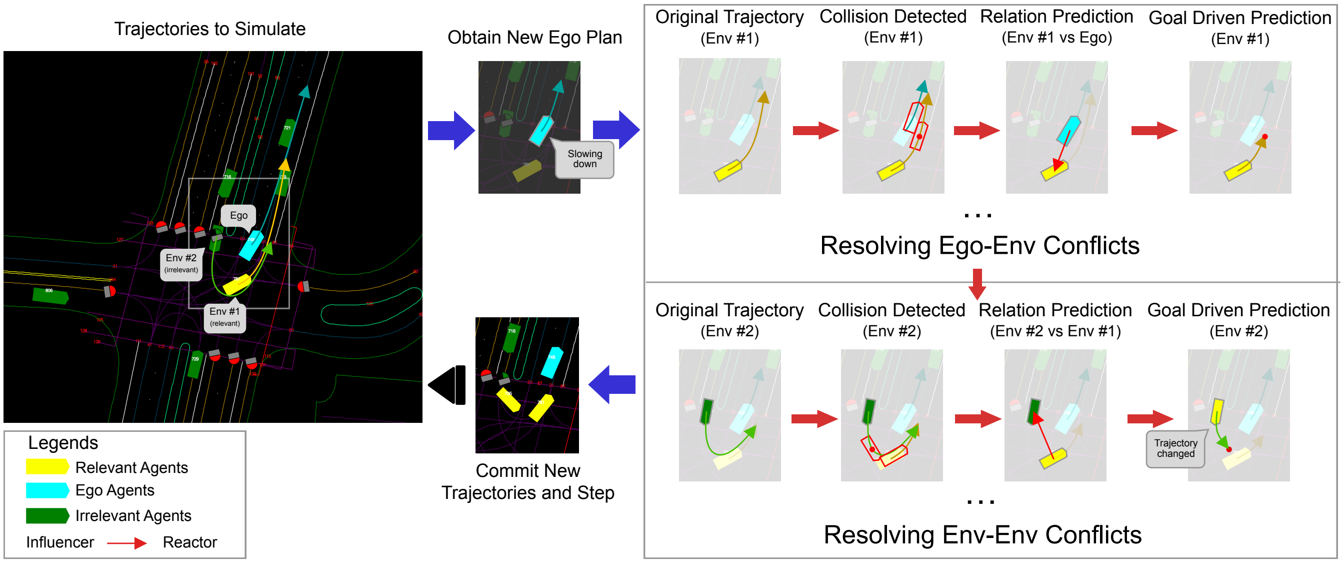
Trajectories to Simulate
Ego
Env #2 (irrelevant)
Env #1 (relevant)
Obtain New Ego Plan
Slowing down
Commit New Trajectories and Step
Legends
Relevant Agents
Ego Agents
Irrelevant Agents
Influencer
Reactor
Original Trajectory (Env #1)
Collision Detected (Env #1)
Relation Prediction (Env #1 vs Ego)
Goal Driven Prediction (Env #1)
...
Resolving Ego-Env Conflicts
Original Trajectory (Env #2)
Collision Detected (Env #2)
Relation Prediction (Env #2 vs Env #1)
Goal Driven Prediction (Env #2)
Trajectory changed
...
Resolving Env-Env Conflicts

Fig. 2: Illustration of InterSim. In this example, given a new plan for the ego agent (in cyan) to slow down, the simulator updates its simulated trajectories for the environment agents as follows. First, it checks for potential collisions with all environment agents and labels colliding ones as the relevant agents in yellow[1]. For each relevant agent, such as Env #1, it predicts the interaction relation and updates its trajectory based on the relation using a goal driven trajectory predictor. Second, it resolves collisions between the newly updated trajectories of the environment agent(s) and the remaining agents (i.e. Env #2) iteratively until all collisions are resolved. In the end, InterSim successfully generates scene consistent trajectories for Env #1 and Env #2 to react and slow down, and commit these trajectories to simulate for the next step.

ground truth future trajectories in the data. Although this consideration may suffer from less diverse scenarios, we observe that most drivers do not change their long-term goals when interacting with others (e.g., an agent turning left in the data is unlikely to change its intent to go straight or turn right) and we can achieve sufficient diversity by simulating multiple options to reach the long-term goals, as we show in the experiments.

IV. Approach

In this section, we describe our approach InterSim that simulates interactive responses of environment agents given an ego plan to be tested. We first present a high-level overview with an illustrated example followed by detailed explanations on conflict detection, relation-aware conflict resolution, goal driven trajectory prediction, conflict resolution between environment agents, and one-step simulation.

A. Overview

As illustrated in Fig. 2, InterSim takes the input, including the past agent states and the environmental context, into a rasterized representation and a vectorized representation. Given a new ego plan (in cyan), InterSim identifies the potential conflicts defined as the collisions between the plan and the simulated trajectory of other agents, as shown on the first row in Fig. 2. For the conflicting agents, InterSim predicts their interaction relations with the ego agent and updates their future trajectories or the ego plan conditioned on the predicted relations and their intent. In cases where the updated trajectories cause new conflicts with the other environment agents, InterSim further updates the colliding trajectories of these agents using the same relation prediction and trajectory prediction procedure until all conflicts are resolved, as shown on the second row in Fig. 2. Following this tree-like structure to resolve collisions, InterSim is guaranteed to produce no dead-end cases. After updating the trajectories of all the relevant environment agents, InterSim advances the simulation by one step and repeats this process until the end of the simulation episode.

B. Conflict Detection

At each simulation step, the ego planner supplies a planned trajectory that may result in conflicts with the future trajectories of the environment agents from log replay or past simulations. The conflict is defined as the collision between agents based on their shapes represented as bounding boxes, as customary in planning [35]. If the bounding boxes between two agents overlap at any time in the future, a conflict is detected and requires the simulator to update the colliding trajectories for better consistency and realism. We present an example of conflict highlighted by the red boxes on the top row (third to the right plot) in Fig. 2, which indicate the collision between the ego agent and environment agent #1.

C. Relation-Aware Conflict Resolution

To resolve a collision conflict, InterSim first identifies the *relevant* agents that have colliding trajectories with the ego

[1] The simulator has already labeled some relevant agents in yellow from the previous simulation step. In the current step, it will re-evaluate the relevance for all agents.

plan and revises their trajectories or the ego plan based on the predicted relations between the ego and relevant agents.

In order to model the interaction relation, we define the interactive agents as the influencer and the reactor, as customary in [30], [31], and predict the relation as a classification problem through a deep neural network. The network includes a context encoder that encodes the input using two encoder heads. The first encoder head follows VectorNet [12] by taking observed agent trajectories and map states as a set of polylines. For each polyline, the encoder first runs an MLP to encode the feature of each vector within the polyline and a graph neural network followed by a max-pooling layer to extract the polyline feature from accumulated vector features. The final vectorized feature is obtained by running cross attentions between the agent polyline features and the map polyline features. Due to the success of rasterized representations in trajectory prediction [15], we leverage a second encoder head by rasterizing the input states into an image with multiple channels that include the position of the agents at each past time step with the map information and obtaining the encoded rasterized feature through a pre-trained VGG16 [36] model. The output of the context encoder is a concatenation of the vectorized feature and the rasterized feature from both encoder heads. Finally, the network predicts the probability over interaction relations using a two-layer MLP. The relation predictor is trained using a cross entropy loss between the predicted probabilities and the ground truth interaction label. The label is obtained at training time based on which agent gets to the cross point first given their ground truth future trajectories in the data, as in [31].

After training a relation predictor from realistic driving data, we can resolve the collision conflict based on the definition that the reactor is always yielding to the influencer. Thus, we keep the trajectory of the influencer unchanged and modify the goal of the reactor to avoid colliding with the influencer trajectory. There exist a few options to set the goal for the reactor, depending on the trade-off between progress (i.e. how far the reactor travels) and simulation consistency (i.e. how likely the agents are colliding). In this work, we update the goal point of the reactor to the colliding point (or the crossing point) between the trajectories as shown in the top right plot in Fig. 2 to favor simulation consistency, such that the agents are unlikely to collide, while ensuring the progress to some extent.

D. Goal Driven Trajectory Prediction

We train a goal-driven trajectory predictor as a deep neural network to help simulate realistic reactor trajectories given its updated goal at the cross point. This predictor is also used to roll out the agent trajectories from novel states unseen in the dataset, or update the ego plan if the ego agent is predicted as the reactor. This network shares the same context encoders as the relation predictor, except that it adds an additional channel in the rasterized representation to include the goal point. The decoder is an MLP-based predictor head that regresses the full trajectory conditioned on the goal. At training time, we use the teacher forcing technique and provide the ground truth goal point from the data. We train the model to minimize the L2 loss between the regressed trajectory and the ground truth future trajectory from data.

E. Conflict Resolution between Environment Agents

After modifying the trajectories to resolve the conflict between the ego and its relevant agents, it is possible that new collisions are created by the modified trajectories. Thus, we follow the same conflict detection and resolution procedure iteratively for agents whose future trajectories are colliding with the updated trajectories until all conflicts are resolved. In the example illustrated in Fig. 2, after InterSim updates the trajectory of environment agent #1 to resolve its collision with the new ego plan (see the top row), it detects a new collision between agent #2 and the updated trajectory of agent #1. In this case, it continues to resolve this conflict by predicting the relations between these two agents and updating the future trajectory of agent #2 as the reactor (see the second row). Following this procedure, our simulator will keep searching for the collisions given newly updated trajectories until all conflicts are resolved. In the end, it produces scene consistent trajectories over all agents.

F. Simulation

Given the predicted trajectories of all agents in a scene, InterSim commits these trajectories to simulate the agent behaviors at the next step. If the plan stays the same at the next step, InterSim will follow the remaining predicted trajectories at each future step, until the end of the simulation horizon. On the other hand, if a new plan is given, it will follow the procedures described in Sec. IV-B to Sec. IV-E to update the environment agent trajectories when necessary.

V. Experimental Results

In this section, we introduce the dataset and the model, and present a series of quantitative and qualitative experiments to demonstrate the effectiveness of our approach compared to the baseline simulators.

A. Dataset and Model Details

We train our simulator on the Waymo Open Motion Dataset (WOMD) [38], a large-scale driving dataset collected from realistic traffic scenarios. More specifically, we focus on the interactive dataset in WOMD that is mined to involve close interactions. At test time, we evaluate our simulator against the validation set, which includes $42,318$ challenging scenarios. In each scenario, the simulator takes as the input one second of the observed trajectory and the environmental context, and simulates the agent behaviors over the next 8 seconds. We follow [8] and choose a simulation frequency at 2Hz.

We implement our relation predictor and goal driven trajectory predictor by following [31], a state-of-the-art interactive trajectory predictor on the WOMD interactive prediction benchmark. Both predictors leverage the same context encoder headers that encode the observed trajectories

Method	Relevant Ratio	ADE ↓	FDE ↓	Front Collision Rate ↓	Side Collision Rate ↓	Rear Collision Rate ↓	Progress ↑
SimNet [4]	21%	8.74	16.99	0.89%	2.80%	4.97%	49.35
InterSim-M0	26%	7.26	6.19	0.57%	1.99%	9.28%	**54.11**
InterSim-M1	23%	8.31	6.58	**0.10%**	**0.41%**	**1.00%**	39.83
InterSim	23%	**7.11**	**6.17**	0.19%	0.71%	1.91%	42.85

TABLE I: Task 1 simulation performance (c.f. Sec. V-C for more details on the metrics) of InterSim and the baselines over 40k interactive scenarios, during which the ego agent is controlled by a state-of-the-art marginal predictor [37]. Compared to the baselines, InterSim achieves the best simulation realism at the cost of small simulation reactivity by modeling explicit interaction relations during simulation.

Method	Rel. Rat.	Front ↓	Side ↓	Rear ↓	Prog. ↑
SimNet [4]	9.33%	0.60%	4.55%	33.90%	46.55
InterSim-M0	11.06%	0.47%	4.90%	35.00%	**50.69**
InterSim-M1	10.38%	**0.24%**	1.98%	**6.94%**	26.67
InterSim	10.49%	0.30%	**1.95%**	7.48%	28.47

TABLE II: Task 2 simulation performance of InterSim and the baselines over 40k interactive scenarios, during which the ego agent follows a slowing down action, requiring the simulator to generate reactive behaviors to avoid collisions. Results in this more challenging task show that an explicit collision resolution process helps decrease the collision rate for all types of collisions by a large margin.

and environment states into a vectorized representation and a rasterized representation. More specifically, we implement a standard VectorNet encoder [12] following the open-source implementations by [37]. To obtain the rasterized feature, we generate a 224×224 image from the past trajectories and the environmental context, in which each pixel represents an area of $1\text{m} \times 1\text{m}$, and run a pre-trained VGG16 [36] model as the raster encoder. The relation prediction head is a two-layer MLP. The first layer has a hidden size of 128, followed by a layer normalization layer and a ReLU activation layer. The second layer outputs the distribution over the binary interaction relations. The goal driven prediction head follows a similar two-layer MLP, consisting of a hidden layer with a size of 128, a layer normalization layer, a ReLU activation layer, and an output layer that outputs the 2-D positions over the simulation horizon. We refer to [31], [37] for additional implementation details.

Both models are trained on the training set from WOMD with a batch size of 64 for 30 epochs. We use an Adam optimizer and a learning rate scheduler that decays the learning rate by 30% every 5 epochs, with an initial value of 1e-3.

We further introduce a few baselines to verify the effectiveness of our approach, which include 1) *SimNet* [4] that simulates agent trajectories using a ResNet-50 model without reasoning about their future interactions; 2) *InterSim-M0* is a variant of our model that simulates agent trajectories but does not resolve the collision conflicts; 3) *InterSim-M1* that is a variant of our model that resolves the collision conflict in a conservative manner by simulating both agents to the cross point without considering their relations, which shares the same spirit as [7]. For each variant, we use the same backbone as our model for a fair comparison.

B. Simulation Tasks

We propose two tasks to test the effectiveness of our simulator. In the first task, we test its capability to generate realistic and consistent trajectories given a proper planner. In the second task, we test its stability against a planner that generates novel out-of-distribution plans with respect to the WOMD dataset. In both tasks, we randomly select the ego agent to be one of the interactive agents labeled in the dataset to test our simulator in highly interactive scenarios.

1) Task 1: Simulation with Respect to Proper Plans: To test the overall performance of InterSim, we control the ego agent using a state-of-the-art marginal trajectory predictor [37], which was the winner of the WOMD marginal prediction challenge. This allows us to simulate realistic ego behaviors that closely follow the data distribution but deviate slightly from the ego future trajectory collected in the data. As a result, it is necessary for the simulator to adjust future trajectories of the environment agents to react to the deviated plan.

2) Task 2: Simulation with Respect to Novel Plans: Learning-based models often suffer from instability given novel inputs, such as the ego trajectories that are out of the distribution of the training dataset. Therefore, one should verify the stability of the simulation given novel ego plans. In this task, we control the ego vehicle to commit a slowing down action with a reasonable deceleration in each scene to test the robustness of our simulator and the baselines, which are required to produce reactive behaviors for the environment agents, especially those behind the ego agent. The deceleration is set to $-1.5m/s^2$. We upper bound the speed of the ego plan by its original speed from the dataset, in case the ego vehicle is going to slow down with a larger deceleration. As the slowing down action is easy to implement and reproduce, we find it straightforward to test and compare the performance of interactive simulators.

C. Quantitative Results

We evaluate the quantitative performance of our model and the baselines against a few standard metrics introduced in [4], [8], [39], including:

- **Relevant ratio** as the number of relevant agents whose future trajectories are modified by the simulator over the number of all environment agents. This measures the density of agents that needs to be simulated in WOMD.
- **Average displacement error (ADE)** and **final displacement error (FDE)** as the distance between the

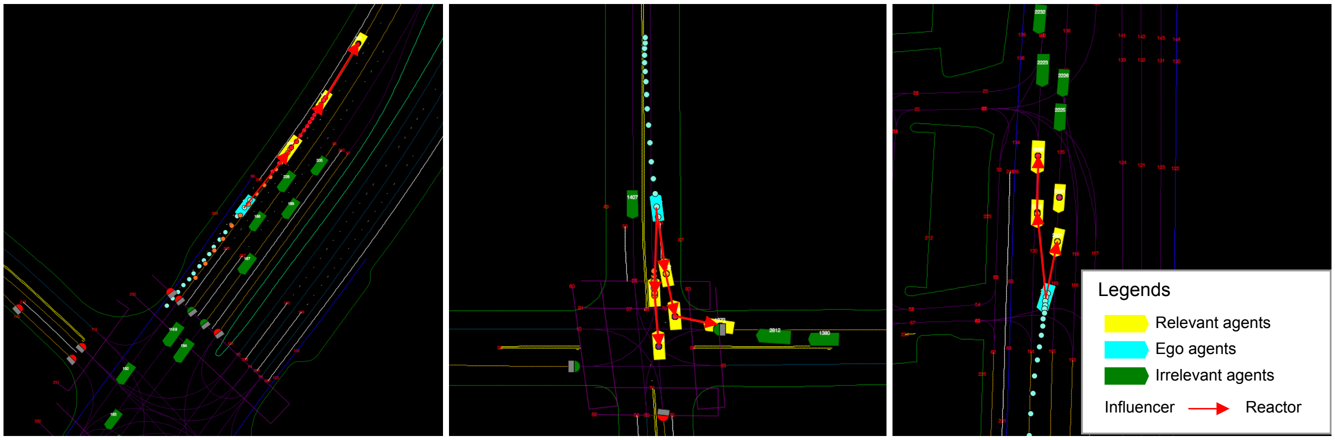

Fig. 3: Examples of InterSim successfully identifying relevant agents in challenging interactive scenarios and simulating realistic interactive behaviors. In the first scenario from the left, our model identifies all the platooning agents in yellow that are relevant to the ego agent plan. In the second and third scenarios, our model identifies the agents with complex interactions during turning and merging. The planned ego trajectories and the simulated trajectories for each relevant environment agent are drawn as dots in different colors.

simulated trajectories and the ground truth future trajectories from data. These two metrics measure the simulation realism [4] and the scenario reconstruction ability [8].

- **Collision rate** as the number of the colliding agent pairs divided by the number of simulated agents. We follow [4] and divide the collisions into front collisions, side collisions, and rear collisions. Such metric measures the simulation reactivity [4] and the interaction reasoning capability [8].
- **Progress** as the total traveled distance of all simulated agents divided by the number of simulated agents. This measures the level of goal achievement [39].

The results are reported in Table I and Table II for Task 1 and Task 2, respectively. We summarize the key observations as follows.

1) Simulation Efficiency: In both simulation tasks, we observe a small portion (i.e. at most 26% for Task 1 and at most 11.06% for Task 2) of relevant agents that are influenced by the ego plan out of all environment agents. This demonstrates that a simulator only needs to modify a small portion of agents on the road to test a planner's performance to achieve better run-time efficiency. A joint trajectory simulator such as [8], on the other hand, might not be the most efficient method especially under time constraints or memory limits.

2) Simulation Realism: In Task 1, we measure the simulation realism as the displacement errors, i.e. ADE and FDE, between the simulated trajectories and the ground truth future trajectories in the data, and observe that our simulator achieves the lowest errors, especially compared to SimNet that leverages a simple ResNet-50 prediction backbone. When compared to the other two invariants using the same prediction backbone, InterSim achieves the best performance by reasoning about the underlying relations between the agents.

3) Simulation Reactivity: We measure the simulation reactivity in terms of collision rates in both tasks. In Task 1 (see Table I), while both InterSim-M1 and InterSim achieve better reactivity by reasoning about future interactions, InterSim-M1 achieves the lowest collision rates as it adopts a conservative approach to resolve conflicts by changing the goals of both interactive agents to the cross point. On the other hand, our approach InterSim resolves the conflicts by only modifying the reactor agent and achieves a better balance between simulation realism and reactivity.

We observe a similar pattern in Task 2 (see Table II), in which we test the simulator with a slowing down ego plan. Both SimNet and InterSim-M0 struggle to resolve new conflicts and cause rear collisions in one-third of the scenarios by simulating the rear agent to collide with the slowing down ego agent. Despite having the lowest values in two types of collisions, InterSim-M1 handles conflicts by yielding both agents without really reasoning about their interactions. On the other hand, our proposed approach InterSim makes more progress at similar collision rate levels by resolving conflicts via explicit interaction modeling.

4) Simulation Progress: We measure the level of goal achievement for all the agents through the progress metric. A good planner should control the ego agent to navigate safely without blocking other traffic. In both tasks, we observe that our approach achieves better progress compared to InterSim-M1 that resolves the interaction conflicts conservatively. In addition, we notice that the simulators tend to achieve lower progress scores in Task 2, due to the slowing down ego behavior.

D. *Qualitative Examples*

1) Realistic Behavior Simulation: In Fig. 3, we present three representative interactive scenarios from Task 1 to showcase InterSim. Our model successfully identifies all relevant agents, as highlighted in yellow, in challenging scenarios involving platooning, turning, and merging, and

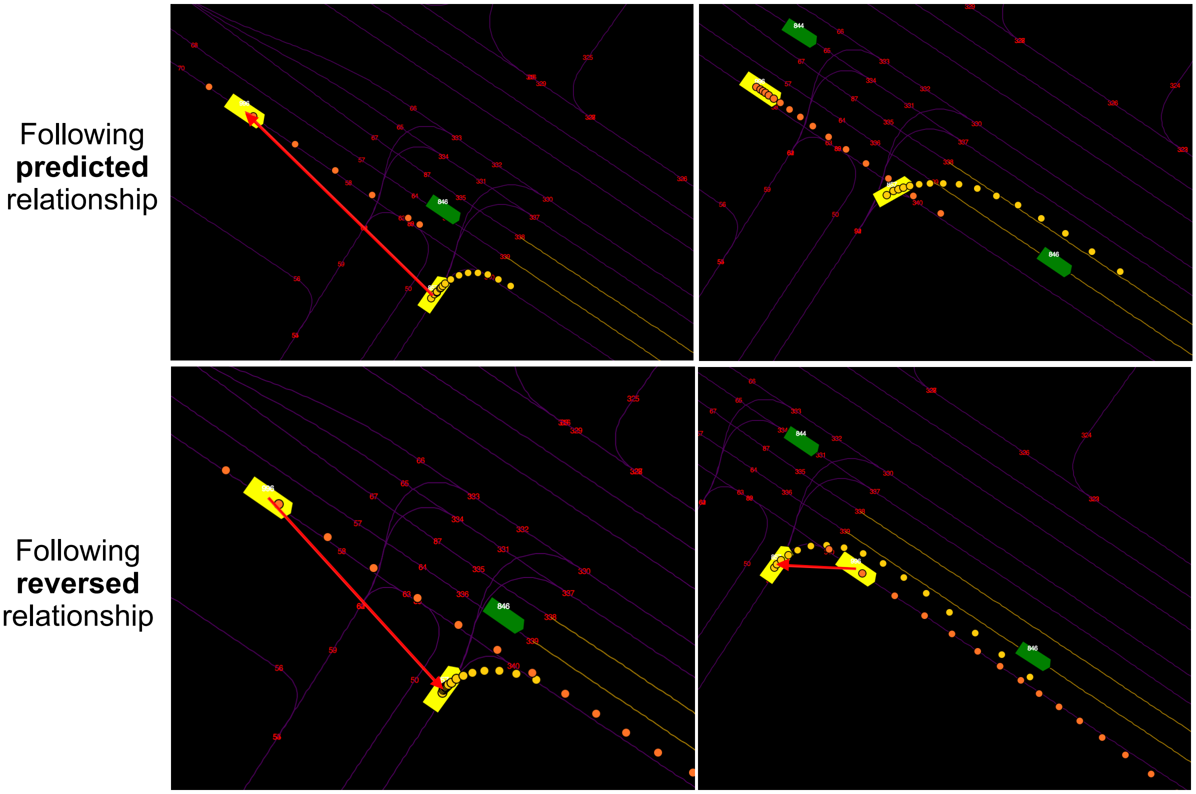
Following **predicted** relationship
Following **reversed** relationship
846
844

Fig. 4: An example of InterSim manipulating the interaction relationship between two agents to simulate different interaction modalities. Top row: The model follows the predicted relation and simulates the forward agent to yield to the turning agent who has already initiated the turn. Bottom row: The model follows a reversed relation to simulate a different scenario.

simulates their future trajectories to be consistent with the interaction relations and the environmental context.

2) Interaction Manipulation with Perturbed Relations: In Fig. 4, we present an example in which InterSim can be used to manipulate the interaction relations between agents to simulate different interactive behaviors. The simulator in this example is used to simulate the trajectories of a forward agent and a turning agent. On the top row, our relation predictor predicts that the forward agent is yielding to the turning agent who already initializes the turn. Given the predicted relation, InterSim simulates a realistic interacting scenario for both agents. In contrast, we can reverse the interaction relation and simulate a different scenario, in which the turning agent yields to the forward agent, as shown on the bottom row. This demonstrates the advantage of our approach that manipulates relations and simulates diverse and rare events, compared to generative models, i.e. [8], that rely on probabilistic sampling approaches and may require a large number of samples to cover rare events.

E. Collisions Analysis

While our model is designed to resolve all collision conflicts in simulation, we observe that the collision rates in the experimental results are not eliminated to zero. We present a couple of reasons behind such failure cases.

First, the goal driven trajectory prediction model is not always stable and may fail given novel inputs, such as strange observations from other agents. Such novel inputs often cause the trajectory to overshoot and lead to a rear collision. Second, due to noise in perception data, including the bounding box labels and the agent positions, there exist a few scenarios that already include collisions at the beginning of the simulation.

VI. Conclusions

In conclusion, we present an interactive traffic simulator InterSim that simulates realistic and reactive agent trajectories with respect to the ego plan. Our simulator takes advantage of a relation predictor to reason about the underlying relations between interactive agents when simulating their future trajectories. In the experiments, we test our simulator and the baselines with two planners, a reasonable planner and a novel planner, to validate the advantage of our approach. Quantitative results show that InterSim achieves a good balance between simulation realism and reactivity, especially in a challenging task in which the ego plan is forced to slow down and requires the simulator to respond. We also present a few representative examples to demonstrate the effectiveness of InterSim in dense interactive scenarios and its ability to manipulate agent interactions to produce diverse and rare events.

References

[1] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[2] J. Liu, W. Zeng, R. Urtasun, and E. Yumer, “Deep structured reactive planning,” in *International Conference on Robotics and Automation (ICRA).* IEEE, 2021, pp. 4897–4904.

[3] X. Huang, G. Rosman, A. Jasour, S. G. McGill, J. J. Leonard, and B. C. Williams, “TIP: Task-informed motion prediction for intelligent vehicles,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2022.

[4] L. Bergamini, Y. Ye, O. Scheel, L. Chen, C. Hu, L. Del Pero, B. Osiński, H. Grimmett, and P. Ondruska, “SimNet: Learning reactive self-driving simulations from real-world observations,” in *International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5119–5125.

[5] K. Sun, S. Chaves, P. Martin, and V. Kumar, “RTGNN: A novel approach to model stochastic traffic dynamics,” in *International Conference on Robotics and Automation (ICRA)*, 2022.

[6] J. Zhou, R. Wang, X. Liu, Y. Jiang, S. Jiang, J. Tao, J. Miao, and S. Song, “Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1450–1457.

[7] M. Vitelli, Y. Chang, Y. Ye, A. Ferreira, M. Wołczyk, B. Osiński, M. Niendorf, H. Grimmett, Q. Huang, A. Jain *et al.*, “SafetyNet: Safe planning for real-world self-driving vehicles using machine-learned policies,” in *2022 International Conference on Robotics and Automation (ICRA).* IEEE, 2022, pp. 897–904.

[8] S. Suo, S. Regalado, S. Casas, and R. Urtasun, “TrafficSim: Learning to simulate realistic multi-agent behaviors,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 10 400–10 409.

[9] 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, p. 2, 2000.

[10] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An open urban driving simulator,” in *Conference on Robot Learning (CoRL).* PMLR, 2017, pp. 1–16.

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

[12] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, “VectorNet: Encoding hd maps and agent dynamics from vectorized representation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 525–11 533.

[13] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, “Learning lane graph representations for motion forecasting,” in *Proceedings of the European Conference on Computer Vision (ECCV).* Springer, 2020, pp. 541–556.

[14] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, “Multimodal trajectory predictions for autonomous driving using deep convolutional networks,” in *International Conference on Robotics and Automation (ICRA).* IEEE, 2019, pp. 2090–2096.

[15] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, “Home: Heatmap output for future motion estimation,” in *IEEE International Intelligent Transportation Systems Conference (ITSC).* IEEE, 2021, pp. 500–507.

[16] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, “Precog: Prediction conditioned on goals in visual multi-agent settings,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2821–2830.

[17] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid *et al.*, “TNT: Target-driven trajectory prediction,” in *Conference on Robot Learning (CoRL)*, 2020.

[18] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, “It is not the journey but the destination: Endpoint conditioned trajectory prediction,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020.

[19] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, “Learning to predict vehicle trajectories with model-based planning,” in *Conference on Robot Learning (CoRL)*, 2021.

[20] B. Kim, S. H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi, “LaPred: Lane-aware prediction of multi-modal future trajectories of dynamic agents,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] N. Deo and M. M. Trivedi, “Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms,” in *IEEE Intelligent Vehicles Symposium (IV).* IEEE, 2018, pp. 1179–1184.

[22] X. Huang, S. G. McGill, J. A. DeCastro, L. Fletcher, J. J. Leonard, B. C. Williams, and G. Rosman, “DiversityGAN: Diversity-aware vehicle motion prediction via latent semantic sampling,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5089–5096, 2020.

[23] X. Huang, G. Rosman, I. Gilitschenski, A. Jasour, S. G. McGill, J. J. Leonard, and B. C. Williams, “HYPER: Learned hybrid trajectory prediction via factored inference and adaptive sampling,” in *International Conference on Robotics and Automation (ICRA)*, 2022.

[24] Y.-L. Kuo, X. Huang, A. Barbu, S. G. McGill, B. Katz, J. J. Leonard, and G. Rosman, “Trajectory prediction with linguistic representations,” in *International Conference on Robotics and Automation (ICRA)*, 2022.

[25] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, “Implicit latent variable model for scene-consistent motion forecasting,” in *Proceedings of the European Conference on Computer Vision (ECCV).* Springer, 2020.

[26] Y. Chen, C. Liu, B. E. Shi, and M. Liu, “HGCN-GJS: Hierarchical graph convolutional network with groupwise joint sampling for trajectory prediction,” *arXiv preprint arXiv:2009.07140*, 2020.

[27] S. Casas, C. Gulino, R. Liao, and R. Urtasun, “SpAGNN: Spatially-aware graph neural networks for relational behavior forecasting from sensor data,” in *International Conference on Robotics and Automation (ICRA).* IEEE, 2020, pp. 9491–9497.

[28] K. Nitin, Z. Hao, T. Dweep, Z. Ming, and L. Yan, “Multi-agent trajectory prediction with fuzzy query attention,” in *Advances in Neural Information Processing Systems*, 2020.

[29] D. Lee, Y. Gu, J. Hoang, and M. Marchetti-Bowick, “Joint interaction and trajectory prediction for autonomous driving using graph neural networks,” *arXiv preprint arXiv:1912.07882*, 2019.

[30] S. Kumar, Y. Gu, J. Hoang, G. C. Haynes, and M. Marchetti-Bowick, “Interaction-based trajectory prediction over a hybrid traffic graph,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2021, pp. 5530–5535.

[31] Q. Sun, X. Huang, J. Gu, B. Williams, and H. Zhao, “M2I: From factored marginal trajectory prediction to interactive prediction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[32] M. R. Adler, A. B. Davis, R. Weihmayer, and R. W. Worrest, “Conflict-resolution strategies for nonhierarchical distributed agents,” in *Distributed artificial intelligence.* Elsevier, 1989, pp. 139–161.

[33] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik, “Chaff: Engineering an efficient sat solver,” in *Proceedings of the 38th annual Design Automation Conference*, 2001, pp. 530–535.

[34] J. Chen, C. Fang, D. Wang, A. Wang, and B. Williams, “Efficiently exploring ordering problems through conflict-directed search,” in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, 2019, pp. 97–105.

[35] X. Huang, M. Feng, A. Jasour, G. Rosman, and B. Williams, “Risk conditioned neural motion planning,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2021, pp. 9057–9063.

[36] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[37] J. Gu, C. Sun, and H. Zhao, “DenseTNT: End-to-end trajectory prediction from dense goal sets,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 303–15 312.

[38] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, “Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9710–9719.

[39] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, “NuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles,” in *CVPR ADP3 workshop*, 2021.