MIXSIM: A Hierarchical Framework for Mixed Reality Traffic Simulation

Simon Suo[1,2*] Kelvin Wong[1,2*] Justin Xu[1,3] James Tu[1,2]
Alexander Cui[1,2] Sergio Casas[1,2] Raquel Urtasun[1,2]
[1]Waabi [2]University of Toronto [3]University of Waterloo
{ssuo,kwong,jjxu,jtu,acui,scasas,urtasun}@waabi.ai

Abstract

The prevailing way to test a self-driving vehicle (SDV) in simulation involves non-reactive open-loop replay of real world scenarios. However, in order to safely deploy SDVs to the real world, we need to evaluate them in closed-loop. Towards this goal, we propose to leverage the wealth of interesting scenarios captured in the real world and make them reactive and controllable to enable closed-loop SDV evaluation in what-if situations. In particular, we present MIXSIM, *a hierarchical framework for mixed reality traffic simulation.* MIXSIM *explicitly models agent goals as routes along the road network and learns a reactive route-conditional policy. By inferring each agent's route from the original scenario,* MIXSIM *can reactively re-simulate the scenario and enable testing different autonomy systems under the same conditions. Furthermore, by varying each agent's route, we can expand the scope of testing to what-if situations with realistic variations in agent behaviors or even safety critical interactions. Our experiments show that* MIXSIM *can serve as a realistic, reactive, and controllable digital twin of real world scenarios. For more information, please visit the project website:* https://waabi.ai/research/mixsim/

1. Introduction

During your commute to work, you encounter an erratic driver. As you both approach a merge, the erratic driver suddenly accelerates and nearly collides with you. Your heart skips a beat and you feel uneasy. While you've avoided an accident, you can't help but wonder: What if the erratic driver was a bit more aggressive? Would you have needed to swerve? Would other drivers be able to react to you?

Having the ability to test a self-driving vehicle (SDV) in these *what-if* situations would be transformational for safely deploying SDVs to the real world. However, the current approach for testing SDVs in simulation lacks this ability. In particular, the self-driving industry largely relies on non-reactive open-loop replay of real world scenarios; traffic agents do not react to the SDV and so the SDV cannot observe the consequences of its actions. This limits the realism and interpretability of the tests. Clearly, naively replaying real world scenarios in simulation is not enough!

*Indicates equal contribution.

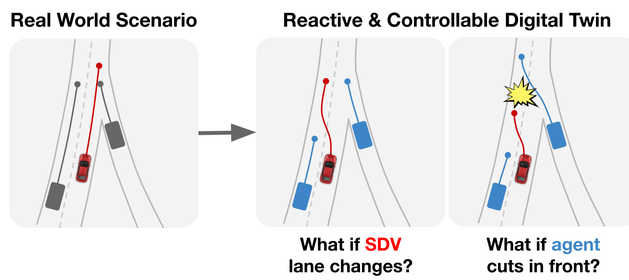

Figure 1. In *mixed reality traffic simulation*, given a real world scenario, we aim to build a reactive and controllable digital twin of how its traffic agents behave. This enables us to re-simulate the original scenario and answer *what-if* questions like: What if the SDV lane changes? What if the agent cuts in front of the SDV?

In this paper, we propose to leverage the wealth of interesting scenarios captured in the real world and make them reactive and controllable to enable closed-loop SDV evaluation in what-if situations. We call this task mixed reality traffic simulation. Specifically, given a recorded scenario, we aim to build a reactive and controllable digital twin of how its traffic agents behave (see Fig. 1). The digital twin should preserve the high-level behaviors and interactions of the original scenario; *e.g.*, a driver cutting in front of the SDV. But it should also react to changes in the environment in a realistic manner. This allows us to reactively re-simulate the original scenario to test different autonomy systems in the same conditions. It also expands the scope of testing to include what-if situations with realistic variations of agent behaviors or even safety critical interactions.

To incorporate reactivity into replaying recorded scenarios, the self-driving industry commonly relies on heuristic models [18, 19, 21, 38] and trajectory optimization-based methods [14, 15]. Complementary to this, [39] use black box optimization over agent trajectories to find safety critical variations of the original scenario. The common pitfall of these methods is that the resulting scenarios do not exhibit human-like driving behaviors. The large sim-to-real gap precludes us from drawing meaningful conclusions of how the SDV will behave in the real world. On the other hand, recent works [2, 17, 36] capture human-like driving by learning generative models of traffic scenarios from real data but can only generate free-flow traffic given the initial scenario context, without preserving the behaviors and interactions in the original scenario. This motivates the need for a simulation framework that learns human-like driving while being controllable at the behavior level.

Towards this goal, we propose MIXSIM, a hierarchical framework for mixed reality traffic simulation. In our approach, we explicitly disentangle an agent's high-level goal (*e.g*., taking an off-ramp) from its reactive low-level maneuvers (*e.g*., braking to avoid collisions). We use routes along the road network as the goal representation and learn a reactive route-conditional policy to recover human-like driving behavior. This enables high-level controllability via specifying the agent's route, while the low-level policy ensures realistic interaction in closed-loop simulation. MIXSIM re-simulates a recorded scenario by first inferring each agent's route from its original trajectory and then unrolling the route-conditional policy. Furthermore, it enables synthesizing what-if situations by conditioning on routes sampled from a learned route-generating policy or routes optimized to induce safety critical interactions.

To understand the suitability of MIXSIM for mixed reality traffic simulation, we conduct an analysis of its sim2real domain gap on both urban and highway traffic scenarios. Our experiments show that MIXSIM exhibits greater realism, reactivity, and controllability than the competing baselines. Notably, MIXSIM achieves the lowest reconstruction error when re-simulating a given scenario and the lowest collision rate when reacting to diverse SDV behaviors. We also demonstrate MIXSIM's ability to simulate useful what-if scenarios for autonomy development. Specifically, MIXSIM can sample diverse yet realistic what-if situations that cover the space of what could have happened and generate safety critical scenarios that are far more realistic than existing methods. Altogether, MIXSIM unlocks closed-loop SDV evaluation in what-if scenarios of varying severity. This represents an exciting first step towards a new paradigm for offline autonomy evaluation.

2. Related Work

Agent behaviors for offline autonomy evaluation: A key challenge in offline autonomy evaluation is how to test the SDV across the broad spectrum of scenarios it might encounter in the real world. A common approach in industry is to use hand-designed scenarios [1,11,35] that validate the SDV under precise interactions. However, this approach is costly, requiring meticulous efforts to ensure realism. Although trajectory optimization-based algorithms [14, 15] can be used to reduce this burden, the overall workflow remains limited in its scalability. Therefore, to complement this approach, engineers also replay scenarios captured from the real world [18]. This is typically done in a non-reactive open-loop manner—agents do not react to the SDV and the SDV cannot execute its actions—which limits the realism and interpretability of the tests. Simple methods to incorporate reactivity have been explored (*e.g*., using heuristic models to modulate speed along the original path [18]) but they fall short in modeling human-like behaviors. In this paper, we also draw on the wealth of interesting scenarios collected from the real world. But unlike existing methods, we propose to build a reactive and controllable digital twin of these scenarios, allowing us to re-simulate and explore what-if variations. As a result, we can test the SDV in closed-loop simulation and obtain a more accurate assessment of its performance in the real world.

Traffic simulation: Traffic simulation has a wide range of applications. In transportation engineering, traffic simulators [26] use heuristic car-following models [19, 21, 38] that capture high-level traffic characteristics accurately (*e.g*. flow, density, *etc*.) but not street-level details. As a result, these models exhibit too large of a domain gap to be used for offline autonomy evaluation. To close this gap, recent work [2, 17, 22, 36] learn traffic models from real data. SimNet [2] learns a deterministic control policy using one-step behavior cloning. To avoid compounding errors [33], [17, 22, 36] learn policies in closed-loop by unrolling them through differentiable simulation. These models share a common focus on generating realistic (and diverse) simulations from an initial scene. But they lack the controllability necessary to re-simulate and manipulate the idiosyncratic behaviors observed in a given reference scenario. The focus of our work is to augment their controllability to enable mixed reality traffic simulation.

Hierarchical model of driving behavior: Building on the intuition that human driving is intentional, recent work have increasingly embraced hierarchical models that disentangle high-level intentions from low-level controls. One line of work uses latent variable models [5,32,36,37] to discover goals, driving styles, and agent-to-agent interactions without explicit supervision. These models accurately capture the multi-modal nature of driving but the resulting latent spaces are generally uninterpretable and difficult to manipulate [25]. An alternative approach uses explicit supervision to learn an interpretable hierarchy. IntentNet [6] uses high-level intentions (*e.g*., turn left), [7,30] use prototypical trajectories, and [8, 13, 20, 41, 42, 46] use goal waypoints at a fixed horizon. We also use explicit supervision to learn an interpretable representation of high-level intentions. In particular, we represent intentions as routes since they are a main source of multi-modality in driving, time-invariant, and easy to manipulate. Our representation most closely resembles the route representations used in [3, 9, 17, 44]. But whereas prior work focus on improving the sample diversity, we focus on using routes to infer and re-simulate high-level behaviors from real world scenarios.

Adversarial agent behaviors: A related line of work focuses on simulating adversarial agent behaviors that induce realistic but safety-critical interactions with the SDV [10]. A popular approach is to directly perturb the agents' trajectories [4, 16, 39, 45]. These methods generate severe but implausible scenarios where the agents are not reactive, disregard traffic rules, and lack human-like interactions (*e.g.*, the agent chases and rear-ends the SDV). To improve realism, recent works incorporate a learned traffic model to find safety-critical scenarios, either perturbing the model's weights [28, 29] or latent variables [31]. Like these methods, MIXSIM uses a learned traffic prior for finding realistic yet safety-critical scenarios. But instead of model weights or latent variables, MIXSIM uses routes, which is a more compact and interpretable representation of behaviors.

3. Mixed Reality Traffic Simulation

MIXSIM is a hierarchical framework for *mixed reality traffic simulation* that enables re-simulating a real world scenario and exploring what-if variations. In the following, we describe our hierarchical model of traffic scenarios, outline how we learn this model from data, and describe three canonical ways to use it for mixed reality traffic simulation.

3.1. A Hierarchical Model of Traffic Scenarios

We model the generative process of traffic scenarios as a multi-agent discrete-time system. A traffic scenario $\mathcal{S}$ consists of high definition (HD) map $\mathbf{m}$, agent states $\mathbf{s}_{0:T}$, and agent actions $\mathbf{a}_{0:T}$. We denote $\mathbf{s}_t$ (resp., $\mathbf{a}_t$) the joint state (resp., action) of all agents at time t. We parameterize the i-th agent's state $\mathbf{s}_{i,t}$ by position, heading, 2D bounding box, and velocity over the past H time-steps and its action $\mathbf{a}_{i,t}$ by acceleration and steering angle. We model agent dynamics $f(\mathbf{s}_{t+1}|\mathbf{a}_t, \mathbf{s}_t, m)$ with the kinematic bicycle model [23].

Formulation: Given a reference scenario, our goal is to build a reactive and controllable digital twin that allows us to re-simulate the scenario and explore what-if variations. The digital twin should preserve the high-level behaviors and interactions of the original scenario (*e.g.*, taking an off-ramp) but not the specific trajectories themselves (*e.g.*, braking to avoid a collision). This motivates explicitly modeling each agent's unobserved goal $\mathbf{g}_i$ in the generative process of traffic scenarios. Concretely, we start with,

$$\mathbf{g}_i \sim h_i(\cdot|\mathbf{s}_0, \mathbf{m}) \tag{1}$$

where h_i is a prior over the i-th agent's goal given the initial state $\mathbf{s}_0$. A scenario unrolls over time according to independent goal-directed agent policies π_i and dynamics f,

$$\mathbf{a}_{i,t} \sim \pi_i(\cdot|\mathbf{s}_t, \mathbf{m}, \mathbf{g}_i) \tag{2}$$

$$\mathbf{s}_{t+1} \sim f(\cdot|\mathbf{s}_t, \mathbf{m}, \mathbf{a}_t) \tag{3}$$

By varying each agent's goal $\mathbf{g}_i$, we can simulate various mixed reality traffic scenarios. Under this model, re-simulating a reference scenario amounts to inferring each agent's goals from its observed behaviors and then unrolling the conditional distribution described in Eqs. (2) to (3). Furthermore, by sampling from the prior over goal $h_i(\mathbf{g}_i|\mathbf{s}_0, \mathbf{m})$, we can automatically generate realistic variations of the reference scenario. Finally, by searching over agent goals with black box optimization, we can automatically discover safety critical yet realistic variations. This allows us to expand the set of possible re-simulations from exploring *what has happened* to *what could have happened*.

Representing goals as routes: We represent each agent's high-level unobserved goals by a route along the road network. Specifically, we represent the HD map as a lane graph $G = (V, E)$ [8, 24, 42]. Each node $u \in V$ is a lane segment and an edge $(u, v) \in E$ indicates that v is a successor, predecessor, left, or right neighbour of u. A route is a directed path of lane segments $u_0, \dots, u_L$ in G. This gives us a compact yet interpretable representation of an agent's goal that captures time-invariant semantics (*e.g.*, whether to go straight or turn) without constraining the agent to a specific sequence of actions. The key challenge is to learn a reactive route-conditional policy with human-like behaviors.

3.2. Learning a Reactive Route-Conditional Policy

Given the current state $\mathbf{s}_t$, the HD map $\mathbf{m}$, and the agents' routes $\mathbf{g}$, the route-conditional policy predicts acceleration and steering for all agents. We want to learn a policy $\pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{m}, \mathbf{g}; \theta)$ with parameters θ that exhibits realistic, reactive, and intentional behaviors. In the following, we outline how we parameterize and learn such a policy. See the supplementary materials for details.

Architecture: Our architecture consists of three building blocks: (1) a set of context encoders; (2) an interaction module; and (3) a route-conditional decoder. Given the current state $\mathbf{s}_t$ and the HD map $\mathbf{m}$, our context encoders use a 1D convolutional network (CNN) followed by a gated recurrent

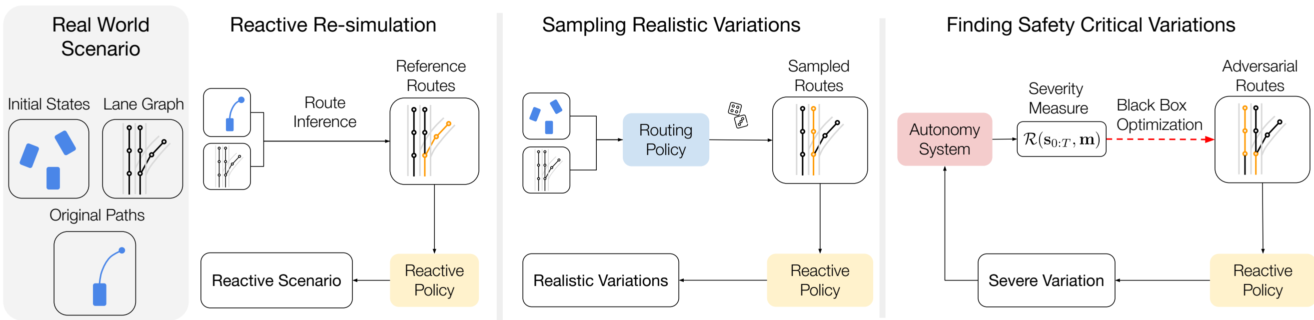

Figure 2. MIXSIM is a hierarchical framework for mixed reality traffic simulation. By varying each agent's route given to the learned reactive route-conditional policy, we can realistically re-simulate a real world scenario in what-if situations: (1) infer reference route from trajectory for **reactive re-simulation**; (2) sample routes from a learned routing policy for **realistic variations**; and (3) find adversarial routes that maximizes a severity measure of the autonomy system under test with black box optimization for **safety critical variations**.

unit (GRU) to encode each agent's past trajectory $\mathbf{s}_{i,t}$ and a graph neural network (GNN) to encode the lane graph representation of $\mathbf{m}$. Next, we use a HeteroGNN [8]—a SoTA GNN architecture for modeling spatial graphs—to fuse the resulting agent and lane graph features, allowing our policy to capture agent-to-agent, agent-to-map, and map-to-map interactions that are critical for simulating realistic reactive behaviors. Finally, for each agent, our decoder aggregates lane graph features along the agent's route, combines them with the agent's features, and use an MLP to predict continuous values representing the acceleration and steering.

Training: To improve robustness to covariate shift [33], related works in traffic simulation train their models using closed-loop training [17, 36]. Likewise, since our dynamics function is fully differentiable, we use gradient-based optimization to learn the policy parameters θ in closed-loop simulation. Specifically, given a dataset $D = \{(\mathbf{s}_{0:T}, \mathbf{m})\}$ of real world scenarios, we optimize an imitation objective,

$$\theta^\star = \arg\min_\theta \mathbb{E}_{(\mathbf{s}_{0:T}, \mathbf{m})\sim D}\left[\frac{1}{T}\sum_{t=1}^{T} \mathrm{d}(\mathbf{s}_t, \tilde{\mathbf{s}}_t)\right] \tag{4}$$

where $\mathrm{d}(\mathbf{s}_t, \tilde{\mathbf{s}}_t)$ is the Huber loss between the positions of each agent in the ground truth $\mathbf{s}_t$ and the simulation $\tilde{\mathbf{s}}_t$. The simulation states $\tilde{\mathbf{s}}_{1:T}$ are generated by iteratively unrolling the policy $\pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{m}, \mathbf{g}; \theta)$ through Eqs. (2) to (3) starting from the initial state $\mathbf{s}_0$ and given ground truth routes reconstructed from each agent's observed behavior in $\mathbf{s}_{0:T}$. We describe our algorithm for route reconstruction in Sec. 3.3.

3.3. Simulating Mixed Reality Traffic Scenarios

We describe three canonical ways we can use our model to simulate mixed reality traffic scenarios. By varying the routes given to the route-conditional policy, we can use our model for reactive re-simulation, sampling realistic variations, and finding safety critical scenarios (see Fig. 2). Moreover, we can combine all three ways to simulate nuanced what-if scenarios in the form of: "what-if agent A cuts in front of the SDV, while agents behind the SDV react realistically, and other agents follow alternative routes."

Reactive re-simulation: To build a digital twin of a scenario, our first step is to infer the routes underlying each agent's original trajectory. To do this, we adapt a hidden Markov model (HMM) for map matching [27] to our setting. Specifically, we model the joint distribution over an agent's states $\mathbf{s}_{i,0:T}$ and underlying route $\mathbf{g}_i$ with an HMM, where $\mathbf{s}_{i,0:T}$ are the observations and the lane segments in $\mathbf{g}_i$ are the hidden variables. We can then estimate an agent's maximum a posteriori route using the Viterbi algorithm,

$$\mathbf{g}_i^\star = \arg\max_{\mathbf{g}_i} p^{\text{hmm}}(\mathbf{g}_i|\mathbf{s}_{i,0:T}) \tag{5}$$

We repeat this process for all agents in the scenario. From here, reactive re-simulation simply amounts to unrolling the route-conditional policy $\pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{m}, \mathbf{g}^\star; \theta)$ through Eqs. (2) to (3) in closed-loop simulation.

Sampling realistic variations: Beyond reactive re-simulation, we can learn a routing policy $h(\mathbf{g}|\mathbf{s}_0, \mathbf{m}; \phi)$ from which we can sample realistic routes to simulate plausible variations of a real world scenario. Our routing policy models the generative process of an agent's route as an autoregressive traversal over the lane graph $G = (V, E)$,

$$h_i(\mathbf{g}_i|u_0, \mathbf{s}_0, \mathbf{m}; \phi) = \prod_{j=0}^{L} h_i^{\text{edge}}(u_{j+1}|u_j, \mathbf{s}_0, \mathbf{m}; \phi) \tag{6}$$

where $\mathbf{g}_i$ is the route $u_0, \ldots, u_L$. Concretely, we parameterize h^{edge} with a similar architecture as that of the route-conditional policy, differing only in their respective decoders. For each agent i and edge (u, v), we concatenate agent features with lane graph features of u and v and use a

Method	Ref. Path	Reconstruction FDE	ATE	CTE	Distribution JSD Speed	Accel.	Lead Dist.	Nearest Dist.	Common Sense Collision	Off Road
Replay									0.80	1.58
Heuristic		14.48	12.65	3.84	0.07	0.36	0.08	0.06	1.49	2.24
BC		8.25	7.34	2.14	0.05	0.43	0.05	0.07	1.51	4.22
IL		5.84	5.32	1.29	**0.04**	0.33	**0.03**	**0.03**	1.21	1.81
Heuristic-Path	✓	12.46	11.88	1.39	0.10	0.33	0.06	0.06	2.02	6.55
BC-Path	✓	8.54	8.14	1.14	0.07	0.40	0.06	0.08	1.34	5.50
IL-Path	✓	6.47	6.09	1.02	0.06	**0.32**	0.04	0.06	1.41	4.25
MIXSIM	✓	**5.04**	**4.86**	**0.67**	**0.04**	0.33	**0.03**	**0.03**	**1.07**	**1.29**

Table 1. Sim2real domain gap results on AV2: *Ref. Path* means method condition on $\mathbf{s}_{0:T}$, otherwise only $\mathbf{s}_0$.

MLP to predict a logit. Then, we decode the transition probability from u_j by taking the softmax over outgoing edges $\{(u, v) \in E : u = u_j\}$. The policy is trained to maximize the likelihood of ground truth routes.

To sample a scenario, we start by associating each agent to its closest lane graph node $u_0 \in V$ at $t = 0$. Then, we iteratively sample edges from $h_i^{\text{edge}}(u_{j+1}|u_j, \mathbf{s}_0, \mathbf{m}; \phi)$ until termination, yielding a route sample,

$$\mathbf{g}_i^\star \sim h_i(\mathbf{g}_i|\mathbf{s}_0, \mathbf{m}; \phi) \tag{7}$$

As before, we unroll our reactive policy to simulate a realistic variation of the original scenario.

Finding safety crticial variations: A critical aspect of autonomy evaluation is stress testing its behavior in safety critical scenarios. MIXSIM enables efficient search over routes to discover realistic yet safety critical interactions with the SDV. To do this, we follow prior work [31, 39] to generate agent routes under an adversarial framework. Given a real world scenario, we determine the set of candidate agents that are capable of interacting with the SDV and sample one of them to have an adversarial route. Then, the goal is to obtain route $\mathbf{g}_i^\star$ which maximizes a severity measure $\mathcal{R}$ to induce a safety critical scenario,

$$\mathbf{g}_i^\star = \arg\max_{\mathbf{g}_i} \mathcal{R}(\mathbf{s}_{0:T}, \mathbf{m}) \tag{8}$$

We choose $\mathcal{R}$ as the SDV's minimum distance to collision. Here $\mathbf{s}_{0:T}$ are the simulation states generated by unrolling our reactive policy and the SDV through Eqs. (2) to (3). Notably, we also re-simulate all other agents to enable realistic closed-loop interactions unlike [39] which replays unrealistic non-reactive behavior. Since $\mathcal{R}(\mathbf{s}_{0:T}, \mathbf{m})$ is a complex function of the autonomy model in a dynamic simulation environment, we treat it as a black-box function and use Bayesian Optimization to solve the maximization problem.

4. Experiments

In this section, we evaluate MIXSIM's suitability for mixed reality traffic simulation. We begin by briefly outlining our experiment setup. Then, we present our main results, demonstrating MIXSIM's ability to build a digital twin of real world scenarios and simulate what-if variations, from realistic variations to safety-critical ones. See the supplementary materials for additional details and results.

4.1. Experiment setup

Datasets: We perform experiments on two datasets that cover urban and highway driving scenarios. Our first dataset Argoverse 2 (AV2) [40] consists of urban scenarios which we split into training and evaluation sets of 50,000 and 2500. These scenarios were mined for multi-agent interactions and complex road topologies. Each simulation is given 5s of context and simulates for 6s. We simulate focal and score vehicles only, which we call the *interactive agents*; other agents' trajectories are replayed due to noisy or incomplete annotations.

Our second dataset HIGHWAY consists of simulated highway scenarios which we split into training and evaluation sets of 800 and 200. The main challenge in these scenarios is modeling its high-speed multi-agent interactions. Each simulation is given 3s of context and simulates for 10s. All vehicle agents are interactive.

Baselines: Our first baseline (**Heuristic**) simulates normative driving behaviors (*e.g.*, collision avoidance, traffic rule compliance) by using IDM [38] for longitudinal control and MOBIL [19] to select target lanes. Our next two baselines represent the state-of-the-art for learning-based traffic simulation. **BC** is trained using one-step behavior cloning [2] and **IL** is trained using closed-loop policy unrolling [17, 36]. Both BC and IL share an identical architecture with MIXSIM, differing only in their decoders. In particular, BC and IL replace MIXSIM's route-conditional decoder with an MLP that directly regresses acceleration and steering angle. Doing so allows us to ablate the efficacy of our route representation.

We also extend these unconditional baselines, which only condition on the initial state, to our setting where they

Method	Ref. Path	Reconstruction FDE	ATE	CTE	Distribution JSD Speed	Accel.	Lead Dist.	Nearest Dist.	Common Sense Collision	Off Road
Replay									0.00	0.00
Heuristic		17.30	17.10	1.08	0.17	0.13	0.06	**0.03**	**0.14**	**0.00**
BC		26.10	17.00	15.60	0.11	0.32	0.18	0.08	11.40	61.60
IL		11.10	10.90	1.05	0.10	0.11	0.05	**0.03**	1.49	**0.00**
Heuristic-Path	✓	17.20	17.20	0.11	0.17	0.14	0.05	**0.03**	2.21	**0.00**
BC-Path	✓	16.80	16.80	0.14	0.10	0.31	0.05	0.04	2.41	0.02
IL-Path	✓	10.90	10.90	**0.10**	0.10	0.11	**0.04**	**0.03**	1.44	**0.00**
MixSim	✓	**10.30**	**10.30**	0.25	**0.09**	**0.10**	**0.04**	**0.03**	0.86	**0.00**

Table 2. Sim2real domain gap results on Highway: *Ref. Path* means method condition on $\mathbf{s}_{0:T}$, otherwise only $\mathbf{s}_0$.

have access to the full reference scenario (**Heuristic-Path**, **BC-Path**, **IL-Path**). In particular, these variants fit smooth driving paths to the ground truth (GT) paths in the reference scenario and use their base models for longitudinal control along these driving paths. After fully traversing their paths, they fallback to their base models for steering as well. Compared to naively interpolating along the GT paths, this approach is more robust to noise in the GT paths and produces trajectories with more realistic kinematics.

Metrics: We propose a suite of metrics that collectively measure realism, reactivity, and controllability—hallmarks of a good digital twin for mixed reality traffic simulation.

- *Reconstruction:* Our first set of metrics measure the ability to reconstruct a real scenario in simulation. Given paired real and simulated scenarios, we compute the final displacement error (**FDE**) [5], which is the L2 distance between an agent's final positions in a pair of scenarios, averaged across agents and scenarios. We also consider the along-track error (**ATE**) and cross-track error (**CTE**) of an agent's final position projected onto its ground truth trajectory. In high-speed scenarios, ATE (resp., CTE) correlates with speed (resp., lateral) variability.
- *Distributional realism:* Our second set of metrics measure the similarity between the distributions of scenario features induced by real scenarios and simulation scenarios. We compute the Jensen-Shannon divergence (**JSD**) [17] between histograms of agent kinematics, agent-to-agent interactions, and agent-to-map interactions. This allows us to evaluate the realism of what-if scenarios that cannot be paired with a real counterpart.
- *Common sense:* Our third set of metrics quantifies realism with respect to our prior knowledge about what constitutes realistic traffic, namely that agents *do not collide* and *do not drive off-road*. Following [36], we measure the percentage of agents who collide with an interactive agent (**Collision**) and the percentage of interactive agents who drive off-road (**Off Road**), averaged over all scenarios.
- *Controllability:* We evaluate MixSim's controllability via cyclic consistency [43]. Specifically, we compute route consistency (**RC**), which measures the path-to-path distance between an agent's desired route and the route reconstructed from its executed trajectory, averaged over agents and scenarios. This metric is based on the intuition that, for an controllable agent, the difference between its desired route and executed route should be small.

4.2. Building a Digital Twin

In this set of experiments, we compare MixSim and the baselines in their capacity to serve as a digital twin of real world scenarios. Our results show that MixSim is more realistic, reactive, and controllable than competing baselines.

Realism analysis: We begin by evaluating how well each method can reconstruct a given scenario when re-simulating all agents in the scenario, including the SDV. Given a reference scenario, MixSim first infers the routes underlying each agent's behaviors and then re-simulates the scenario conditional on these routes. We compare against both the unconditional baselines and their path-following variants.

Tab. 1 and Tab. 2 summarize our results in AV2 and Highway. In AV2, MixSim outperforms the baselines across all metrics, demonstrating its ability to faithfully re-simulate challenging urban scenarios. In Highway, where it is essential to accurately model speed profiles over long horizons, methods that learn using closed-loop training (*i.e.*, IL and MixSim) achieve the lowest reconstruction error and the best distributional realism in terms of agent-to-agent interactions. Unsurprisingly, the path-following baselines achieve the lowest cross-track error. We note, however, that they do not achieve zero cross-track error since: (1) they fallback to their base models for steering control after fully traversing their GT paths; and (2) they do not follow the agents' GT paths exactly but rather their smoothed versions. Moreover, MixSim achieves the lowest collision rate across both datasets, even in comparison to the path-following baselines. This highlights a key limitation of

Method	Replay SDV CR	Replay SDV JSD	Braking SDV CR	Braking SDV JSD	Aggr. SDV CR	Aggr. SDV JSD
Replay	-	-	2.92	-	2.01	-
Heuristic-Path	2.03	0.10	2.14	0.10	2.23	0.09
BC-Path	2.78	0.14	4.70	0.14	3.99	0.12
IL-Path	1.41	**0.07**	1.44	**0.08**	1.88	**0.07**
MIXSIM	**1.17**	**0.07**	**1.28**	**0.08**	**1.42**	**0.07**

Table 3. MIXSIM reacts realistically to a range of SDV policies, achieving lowest collision rate (CR) on HIGHWAY.

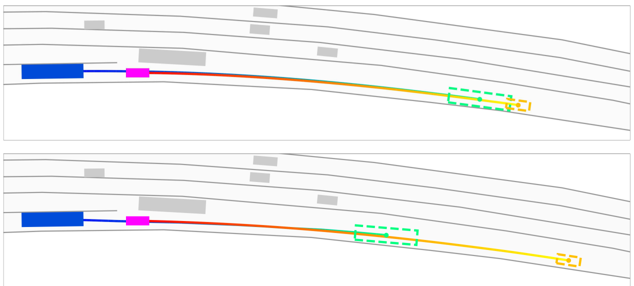

Figure 3. **Top**: The non-reactive Replay agent rear-ends the SDV. **Bottom**: The MIXSIM agent reacts realistically to avoid collision.

rigidly following the agents' original paths—it makes realistic reactions like nudging and swerving impossible.

Reactivity analysis: To answer what-if questions, our digital twin must also react realistically to changes in the environment, notably when the SDV is controlled by a new autonomy stack under evaluation. In this experiment, we compare MIXSIM and the baselines on their ability to simulate realistic reactive behaviors. We re-simulate each HIGHWAY scenario where the simulation model controls all agents but the SDV, which is controlled by one of three policies: (1) static replay of its original trajectory; (2) braking then lane following; and (3) a state-of-the-art motion planner [34] configured to be aggressive. Notably, these SDV policies are designed such that the resulting scenarios are *solvable*; *i.e.*, agents can reasonably react to avoid collision.

From Tab. 3, we observe that MIXSIM achieves the best generalization to different SDV behaviors, achieving the lowest collision rate among all methods while continuing to exhibit realistic behaviors, as measured by JSD. In Fig. 3, we qualitatively illustrate how MIXSIM simulates realistic reactive behaviors when the SDV deviates from its original trajectory. In contrast, Replay simulates an unrealistic collision, limiting its usefulness for autonomy evaluation.

Controllability analysis: A key capability of mixed reality traffic simulation is editing the behavior of certain agents to evaluate how the autonomy system will respond.

Method	GT Route RC	GT Route JSD	Sampled Route RC	Sampled Route JSD	Heuristic Route RC	Heuristic Route JSD
NO-ROUTE	0.51	0.08	1.39	0.08	0.91	0.08
MIXSIM	**0.03**	**0.06**	**0.07**	**0.07**	**0.15**	**0.08**

Table 4. MIXSIM generalizes to novel routes from different distributions in HIGHWAY scenarios

To achieve this, the simulation must be *controllable* while remaining realistic. We evaluate the controllability of MIXSIM by varying the input routes: (1) inferred from the reference scenario; (2) sampled from a learned routing policy; and (3) sampled from a heuristic routing policy. Notably, the last configuration allows us to evaluate how well our model generalizes to novel, out-of-distribution routes.

Tab. 4 shows that MIXSIM generalizes well to novel routes, with minimal degradation in controllability and realism as we move from GT routes seen during training to novel routes. In contrast, the unconditional model implicitly follows the GT route but cannot generalize to novel ones.

4.3. Simulating What-if Variations

In this section, we demonstrate how mixed reality traffic simulation can be used to simulate variations of the original scenario that expand the scope of testing beyond reactive re-simulation. Specifically, we evaluate MIXSIM's ability to sample realistic variations and find safety critical ones. Our results show that MIXSIM is a promising first step towards a new paradigm for offline autonomy evaluation.

Sampling realistic variations: We evaluate MIXSIM's ability to sample diverse but plausible what-if variations of *what could have happened*. For each scenario in HIGHWAY, we sample $K = 6$ variations with each method. For Heuristic, we sample IDM and MOBIL parameters at the start of the scenario; for BC and IL, we use Monte Carlo dropout [12] to sample kinematic controls at each step of the simulation; and for MIXSIM, we sample routes at the start of the scenario. We evaluate diversity with *final displacement diversity* (**FDD**), which measures the maximum difference in final displacement among variations. We also report the minimum scenario FDE (**minSFDE**) [36], where the error is computed for only the best matching sample.

From Tab. 5, we can observe that MIXSIM achieves higher diversity than BC and IL, reflecting its ability to better model multi-modality through routes rather than controls. Moreover, although MIXSIM is less diverse than Heuristic, it is significantly more realistic. See Fig. 4 for AV2 variations sampled from MIXSIM.

Finding safety critical variations: We evaluate MIXSIM's ability to find realistic safety-critical varia-

Method	FDD ↑	minSFDE ↓	JSD
Heuristic	**14.90**	16.00	0.10
BC	3.44	25.20	0.20
IL	8.77	10.50	0.10
MIXSIM	9.83	**10.40**	**0.08**

Table 5. MIXSIM generates more diverse and realistic variations of HIGHWAY scenarios compared to baselines.

	Attack		Distribution JSD		
Method	ASR ↑	Iter.	Accel.	L. Accel.	Curvature
BICYCLE	**43.7**	26.4	0.57	0.83	0.82
BICYCLE-F	**43.7**	21.9	0.57	0.83	0.82
MIXSIM	26.5	**18.6**	**0.31**	**0.53**	**0.49**

Table 6. MIXSIM finds more realistic safety-critical scenarios with fewer optimization iterations compared to baselines.

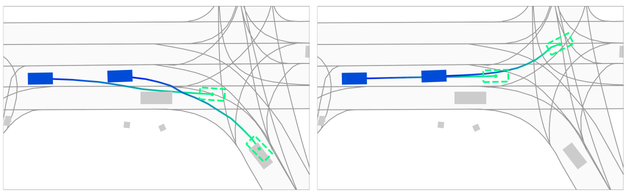

Figure 4. AV2 scenario variations sampled from MIXSIM

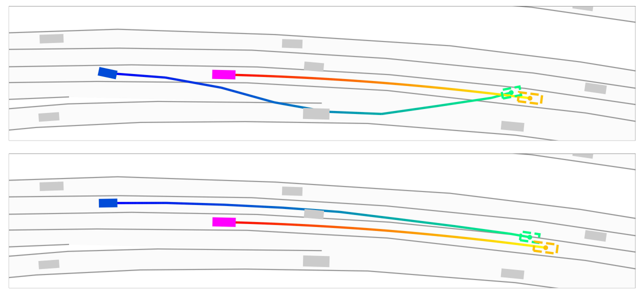

Figure 5. **Top**: ADVSIM [39] finds unrealistic scenario with erratic agent behavior, **Bottom**: MIXSIM finds a more realistic safety-critical scenario.

tions of real world scenarios. Across HIGHWAY, we identify 400 candidate agents with which we conduct an attack on the SDV. During the attack, non-adversarial agents are re-simulated using MIXSIM given GT routes. We use a budget of 75 iterations for bayesian optimization. Optimization terminates upon finding a successful attack where the SDV collides into another actor. Then, we evaluate: (1) attack success rate (**ASR**); (2) average number of optimization iterations to attack; and (3) distributional realism of the adversarial agents. When measuring average number of optimization iterations, we ignore attacks that succeed within five iterations to filter out trivial attacks. For distributional realism, we highlight the kinematic and agent-to-map realism via acceleration, lateral acceleration, and lane-relative curvature. Metrics on agent-to-agent interactions are less relevant, since it conflicts with our adversarial objective of creating collisions.

We consider two baselines inspired by [39]. In BICYCLE, we parameterize each agent's behavior via a bicycle model trajectory with limits on steering and acceleration. In BICYCLE-F, we additionally constrain the search space to a feasible set of trajectories (*e.g.*, no off-road, no collision with non-SDV agents) that are generated as a preprocessing step [39]. In contrast, MIXSIM constrains the search space via the route parameterization of agent behavior, thus naturally enforcing realism learned by our reactive policy.

From Tab. 6, we observe that BICYCLE, the least constrained search space, requires the highest optimization budget and exhibits the worst realism. By constraining the search space to a feasible set of trajectories, BICYCLE-F improves optimization efficiency but does not improve realism. Compared to the baselines, MIXSIM is the most efficient and generates adversarial scenarios that are far more realistic. The difference is apparent in Fig. 5. Our experiments highlight a trade-off in attack strength *vs*. realism and search efficiency. MIXSIM encodes stronger realism constraints via a learned policy, favoring realistic safety-critical scenarios that are more relevant to real-world driving and ultimately more *useful* for autonomy development.

5. Conclusion

In this paper, we presented MIXSIM, a hierarchical framework for *mixed reality traffic simulation* to enable closed-loop SDV evaluation in what-if situations. MIXSIM explicitly models agent goals as routes along the road network and learns a reactive route-conditional policy. MIXSIM can reactively re-simulate real world scenarios to explore *what has happened*. Moreover, MIXSIM can expand the scope of testing to *what could have happened*, with realistic variations in agent behaviors or even safety critical interactions. This represents a promising first step towards a new paradigm for offline autonomy evaluation.

We hope MIXSIM establishes a holistic view on the problem of mixed reality traffic simulation and sets the stage for important extensions. Some directions include: (1) improving nominal realism by incorporating traffic rules for policy learning; (2) improving realism under adversarial optimization by training with out-of-distribution routes in closed-loop simulation; and (3) improving controllability over longitudinal behaviors; *e.g.*, driving fast *vs*. slow.

Acknowledgements: We thank Chris Zhang for insightful discussions and the Waabi team for invaluable support.

References

[1] Matthias Althoff, Markus Koschi, and Stefanie Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *IV*, 2017. 2

[2] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Blazej Osinski, Hugo Grimmett, and Peter Ondruska. Simnet: Learning reactive self-driving simulations from real-world observations. In *ICRA*, 2021. 2, 5

[3] Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, Justin Fu, Austin Abrams, Punit Shah, Evan Racah, Benjamin Frenkel, Shimon Whiteson, and Dragomir Anguelov. Hierarchical model-based imitation learning for planning in autonomous driving. In *IROS*, 2022. 3

[4] Yulong Cao, Danfei Xu, Xinshuo Weng, Zhuoqing Mao, Anima Anandkumar, Chaowei Xiao, and Marco Pavone. Robust trajectory prediction against adversarial attacks. In *CoRL*, 2022. 3

[5] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *ECCV*, 2020. 2, 6

[6] Sergio Casas, Wenjie Luo, and Raquel Urtasun. IntentNet: Learning to predict intention from raw sensor data. In *CoRL*, 2018. 2

[7] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, 2019. 2

[8] Alexander Cui, Sergio Casas, Kelvin Wong, Simon Suo, and Raquel Urtasun. GoRela: Go relative for viewpoint-invariant motion forecasting. In *ICRA*, 2023. 3, 4

[9] Nachiket Deo, Eric Wolff, and Oscar Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *CoRL*, 2021. 3

[10] Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical driving scenario generation - A methodological perspective. *CoRR*, 2022. 3

[11] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. CARLA: an open urban driving simulator. In *CoRL*, 2017. 2

[12] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 7

[13] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. GOHOME: graph-oriented heatmap output for future motion estimation. In *ICRA*, 2022. 3

[14] Yi Han, Jiaping Ren, Shuning Wang, Wenxin Sun, Ruigang Yang, and Xiaogang Jin. TraEDITS: Diversity and irregularity-aware traffic trajectory editing. *RA-L*, 2022. 2

[15] Yi Han, He Wang, and Xiaogang Jin. Spatio-temporal keyframe control of traffic simulation using coarse-to-fine optimization. *CoRR*, 2022. 2

[16] Niklas Hanselmann, Katrin Renz, Kashyap Chitta, Apratim Bhattacharyya, and Andreas Geiger. KING: generating safety-critical driving scenarios for robust imitation via kinematics gradients. In *ECCV*, 2022. 3

[17] Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. In *ICRA*, 2022. 2, 3, 4, 5, 6

[18] Applied Intuition. Using re-simulation to verify an av stack against disengagements, 2021. https://blog.applied.co/blog-post/closed-loop-log-replay. 1, 2

[19] Arne Kesting. MOBIL: General lane-changing model for car-following models. 2007. 1, 2, 5

[20] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction for autonomous driving. *CoRR*, 2020. 3

[21] Karsten Kreutz and Julian Eggert. Analysis of the generalized intelligent driver model (GIDM) for uncontrolled intersections. In *ITSC*, 2021. 1, 2

[22] Alex Kuefler, Jeremy Morton, Tim Allan Wheeler, and Mykel J. Kochenderfer. Imitating driver behavior with generative adversarial networks. In *IV*, 2017. 2

[23] Steven M. LaValle. *Planning Algorithms*. 2006. 3

[24] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 3

[25] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019. 2

[26] Pablo Álvarez López, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie WieBner. Microscopic traffic simulation using SUMO. In *ITSC*, 2018. 2

[27] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *SIGSPATIAL*, 2009. 4

[28] Justin Norden, Matthew O'Kelly, and Aman Sinha. Efficient black-box assessment of autonomous vehicle safety. *CoRR*, 2019. 3

[29] Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, Russ Tedrake, and John C. Duchi. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *NeurIPS*, 2018. 3

[30] Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, Oscar Beijbom, and Eric M. Wolff. CoverNet: Multimodal behavior prediction using trajectory sets. In *CVPR*, 2020. 2

[31] Davis Rempe, Jonah Philion, Leonidas J. Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *CVPR*, 2022. 3, 5

[32] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. PRECOG: Prediction conditioned on goals in visual multi-agent settings. In *ICCV*, 2019. 2

[33] Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011. 2, 4

[34] Abbas Sadat, Mengye Ren, Andrei Pokrovsky, Yen-Chen Lin, Ersin Yumer, and Raquel Urtasun. Jointly learnable behavior and trajectory planning for self-driving vehicles. In *IROS*, 2019. 7

[35] John M. Scanlon, Kristofer D. Kusano, Tom Daniel, Christopher James Alderson, Alexander Ogle, and Trent Victor. Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. 2021. 2

[36] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. TrafficSim: Learning to simulate realistic multi-agent behaviors. In *CVPR*, 2021. 2, 4, 5, 6, 7

[37] Charlie Tang and Russ Salakhutdinov. Multiple futures prediction. In *NeurIPS*, 2019. 2

[38] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 2000. 1, 2, 5

[39] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. AdvSim: Generating safety-critical scenarios for self-driving vehicles. In *CVPR*, 2021. 2, 3, 5, 8

[40] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 2021. 5

[41] Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. BITS: Bi-level imitation for traffic simulation. In *ICRA*, 2023. 3

[42] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. LaneRCNN: Distributed representations for graph-centric motion forecasting. In *IROS*, 2021. 3

[43] Eric Zhan, Albert Tseng, Yisong Yue, Adith Swaminathan, and Matthew J. Hausknecht. Learning calibratable policies using programmatic style-consistency. In *ICML*, 2020. 6

[44] Lingyao Zhang, Po-Hsun Su, Jerrick Hoang, Galen Clark Haynes, and Micol Marchetti-Bowick. Map-adaptive goal-based trajectory prediction. In *CoRL*, 2021. 3

[45] Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z. Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. In *CVPR*, 2022. 3

[46] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. TNT: target-driven trajectory prediction. In *CoRL*, 2020. 3