

Long-term Traffic Simulation with Interleaved Autoregressive Motion and Scenario Generation

Xiuyu Yang* Shuhan Tan* Philipp Krähenbühl
 UT Austin

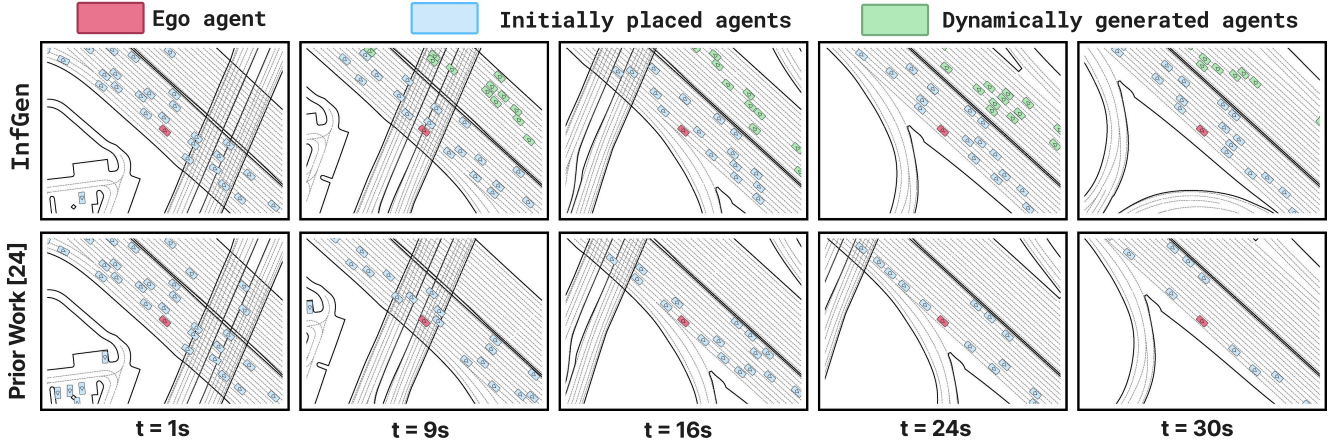


Figure 1. Long-term traffic simulation with InfGen and prior SOTA [31]. InfGen keeps scene layout realistic while [31] becomes empty.

Abstract

An ideal traffic simulator replicates the realistic long-term point-to-point trip that a self-driving system experiences during deployment. Prior models and benchmarks focus on closed-loop motion simulation for initial agents in a scene. This is problematic for long-term simulation. Agents enter and exit the scene as the ego vehicle enters new regions. We propose *InfGen*, a unified next-token prediction model that performs interleaved closed-loop motion simulation and scene generation. *InfGen* automatically switches between closed-loop motion simulation and scene generation mode. It enables stable long-term rollout simulation. *InfGen* performs at the state-of-the-art in short-term (9s) traffic simulation, and significantly outperforms all other methods in long-term (30s) simulation. The code and model of *InfGen* will be released at <https://orangesodahub.github.io/InfGen>.

1. Introduction

Traffic simulation is a cornerstone of the extensive and safe development of self-driving systems. The ultimate goal of traffic simulation is to create realistic trip-level driving experiences that faithfully reflect real-world self-driving con-

ditions [1, 7, 9, 13, 30]. A simulator should provide a realistic model of the environment, the ego-vehicle, and all other traffic agents throughout the trip. Existing simulators easily handle an expansive static environment [1, 9, 30] and intricate ego-vehicle dynamics [7, 13]. However, they often lack a stable long-term simulation of non-ego traffic agents.

In this paper, we introduce *InfGen*, a long-term traffic simulator: Given a short (1 second) driving-log, *InfGen* simulates realistic traffic flow around the ego agent for up to 30 seconds. This long-term setting leads to new challenges. The ego agent may move outside its initial simulation area, leading to logged agents moving out of sight and becoming irrelevant. Furthermore, when the ego agents drive into new map area not covered in the log, these street areas have no agents. Gradually, a scenario becomes sparser and eventually empty (Fig. 1 bottom row). This is clearly unrealistic. *InfGen* models this by combining closed-loop motion simulation with scene generation to remove exiting agents and spawn new agents according to the spatial scene layout.

InfGen (Fig. 2) is a unified autoregressive transformer with interleaved token prediction. It handles *temporal* motion simulation and *spatial* scene generation in a unified model. We design a set of tokenizers to convert task-specific behaviors of motion simulation and scenario generation into discrete tokens. We then add mode-control tokens to mark

*Equal contribution. Work done while Xiuyu interned at UT Austin.

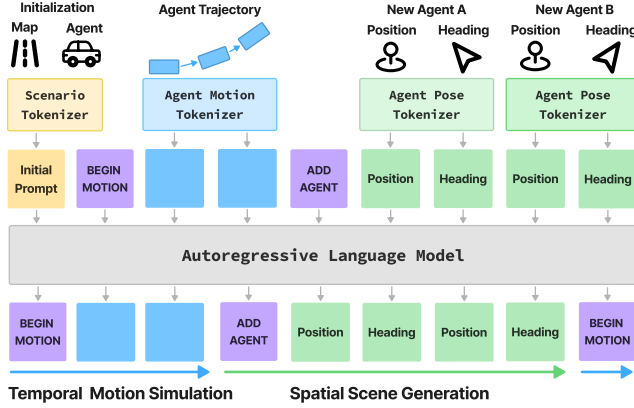


Figure 2. Overview of InfGen interleaved next-token-prediction process. Colors mark different token modalities.

the task switch between the two tasks, indicating what the current task is and when to switch. This design allows us to convert each real log into a single ordered sequence of tokens containing interleaved data of both tasks. We directly train InfGen with the next token prediction objective end-to-end on real data. Noticeably, thanks to the next token prediction formulation, we can train InfGen on short-term driving logs and produce stable long-term rollouts, up to $6\times$ longer than the training horizon (Fig. 1 top). We show a detailed pipeline of InfGen in Fig. 3.

We show in Section 5 that InfGen significantly outperforms prior SOTA models [31, 41] in 30-second long-term traffic simulation for in terms of both motion and scenario realism, showing its strong capacity for stable long-horizon rollout. We contrast different models’ rollout with visualizations (Fig. 5). Furthermore, InfGen achieves strong performance even on the standard short-term Sim Agent setting [14], showing its strong adaptivity to different rollout horizons. In addition, we provide a comprehensive set of analysis on the long-term traffic simulation task to shed light on research towards trip-level driving simulators.

2. Related Work

Closed-loop Motion Simulation. In traffic simulators, motion simulation aims to model realistic multi-agent interactions that mimic real-world data. Early works like Wayformer [15] focus on open-loop imitation learning from real logs [20, 36, 38, 40, 43]. Other works like ProSim [25] and CAT-K [41] instead focus on modeling closed-loop interaction between agents [2, 22, 39]. Ctrl-Sim [18] learns reactive agents by applying offline reinforcement learning to diverse traffic scenarios. More recent works like SMART [31] discover the effectiveness of modeling this task as an autoregressive next-token-prediction problem [11, 16, 21, 42, 46]. ProSim [25] enables multimodal prompts to control behavior semantics of any agent. Most

recently, GIGAFLOW [7] shows strong agent performance emerges from large-scale self-play in simulation. For this direction, nuPlan [1] and WOSAC [14] provide data and benchmark for fair comparisons. All these works focus on simulating motions of agents existed in the history, leading to unrealistic scene layouts under long-term rollout. InfGen solves this issue with interleaved scene generation, maintaining realistic scene layout across the rollout horizon.

Traffic Scenario Generation. This line of work focuses on generation realistic and interesting traffic scenarios. Early works like SceneGen [23] and TrafficGen [10] generates agent initial poses on an empty map. Another popular direction is to generate near-collision scenarios with adversarial optimization [4, 17, 29, 33, 35]. More recent works like LCTGen [24] enables better customizations of the generated scenarios in forms of text [24, 44], scenario queries [8] or cost functions [45]. SLEDGE [6] combine generative models with rule-based traffic simulation to synthesize dynamic driving scenarios. These works either focus on static scene layout initialization, or only generate short-term open-loop scenarios. In contrast, InfGen conducts dynamic scenario layout generation during closed-loop rollout, enabling stable long-term simulation. Concurrent works like SceneDiffuser++ [27] and ScenarioDreamer [19] introduces a vectorized latent diffusion approach to generate realistic and diverse driving simulation environments.

Interleaved Next-Token Prediction. Recent advances in vision-language models have sparked works that unify generation and understanding tasks with interleaved mixed-modal token sequences. For example, Chameleon [3] proposes to train LLMs on interleaved text and image tokens in any arbitrary sequences. This line of works show strong performance on traditional multimodal tasks, but also long-form mixed modal generation that interleaves between image and text generation [3, 12, 28, 32, 37]. InfGen follows the same philosophy but focus on a different pair of modalities: temporal agent motion and spatial agent layout.

3. Problem Formulation

The goal of traditional traffic simulation [14] is to predict future agent trajectories given historical observations (with time span T_H) and a static map. Specifically, at timestep t_0 , we are given the static map $\mathcal{M}$ and the history states of all the agents $\mathcal{A}_{0:t_0} = \{a_{0:t_0}^1, a_{0:t_0}^2, \dots, a_{0:t_0}^N\}$, where each agent a^i has history states up to timestep t_0 . The standard task is to predict future agent states over a fixed horizon T , formulated as estimating the conditional distribution: $p(\mathcal{A}_{t_0+1:T} \mid \mathcal{M}, \mathcal{A}_{0:t_0})$. Prior work [25, 31] factorize the simulation on the time axis and turn it to an autoregres-

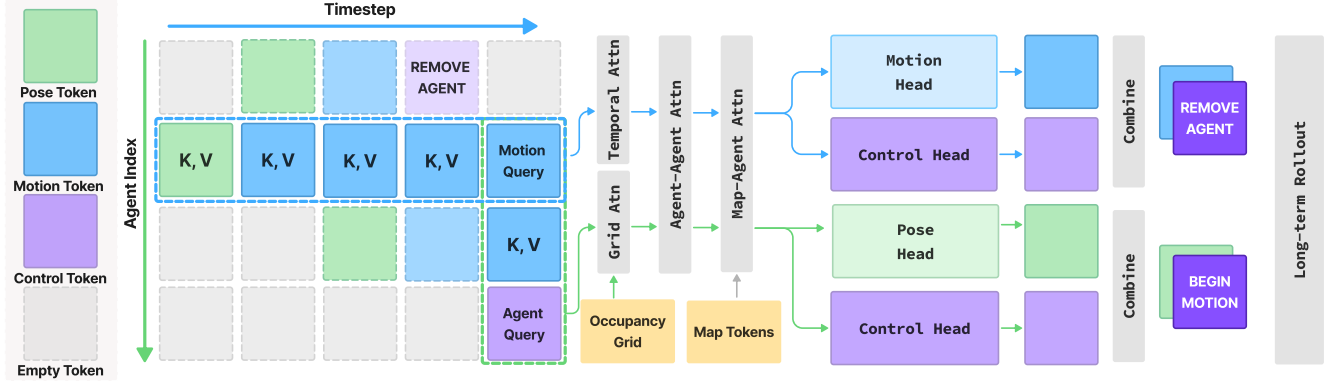


Figure 3. Pipeline of InfGen interleaved motion simulation (blue flow) and scene generation (green flow). For either task, we first pass its query feature through blocks of attention layers and feed it to a task-specific head and a control head. We then sample from both heads to obtain a motion token or pose token, as well as a control token, which determines which task to execute next.

sive prediction task:

$$p(\mathcal{A}_{t_0+1:T} \mid \mathcal{M}, \mathcal{A}_{0:t_0}) = \prod_{t=t_0}^{T-1} p(\mathcal{A}_{t+1} \mid \mathcal{M}, \mathcal{A}_{0:t}). \quad (1)$$

However, this formulation assumes a fixed set of agents throughout the prediction horizon, which does not hold true for long-term simulations with the horizon $T' (\gg T)$. In realistic scenarios (e.g., [14]), agents dynamically enter and leave the observable region around the ego vehicle. To address this issue, we model two interleaved processes at each timestep: 1) motion simulation: predicts future motions for existing agents; 2) scene generation: dynamically inserting new agents and removing agents exiting the scenario. At each timestep t , we first performs motion simulation: $\mathcal{A}_{t+1} \sim p_{\text{motion}}(\mathcal{A}_{t+1} \mid \mathcal{M}, \mathcal{A}'_{0:t})$, followed by scene generation: $\mathcal{A}'_{t+1} \sim p_{\text{scene}}(\mathcal{A}'_{t+1} \mid \mathcal{M}, \mathcal{A}_{t+1})$. Here, $\mathcal{A}_{t+1}$ contains the predicted motions of existing agents, while $\mathcal{A}'_{t+1}$ represents the updated set of agents after adding new agents and removing exiting agents. Then, we formulate the *long-term traffic simulation* task as:

$$p(\mathcal{A}'_{t+1:T'} \mid \mathcal{M}, \mathcal{A}_{0:t_0}) = \prod_{t=t_0}^{T'-1} p_{\text{scene}}(\mathcal{A}'_{t+1} \mid \mathcal{M}, \mathcal{A}_{t+1}) \times p_{\text{motion}}(\mathcal{A}_{t+1} \mid \mathcal{M}, \mathcal{A}'_{0:t}). \quad (2)$$

4. InfGen

InfGen is a unified autoregressive model for long-term traffic simulation. Regarding the inputs, it first tokenizes all scene context information ($\mathcal{M}$ and $\mathcal{A}_{0:t_0}$) into sequences of discrete tokens, see Sec. 4.1. It then uses an autoregressive model for interleaved next-token prediction, see Sec. 4.2. And Secs. 4.3 and 4.4 provide the details on the model architecture and training process of InfGen.

4.1. Tokenization

We aim to convert all agent motions, layouts and the map in a real log into a sequence of discrete tokens. We tokenize each modality differently.

Map Tokenizer. We adapt the map tokenizer from [31]: we uniformly segment all the road elements into a set of fixed-length road vectors. Each vector contains the corresponding features, including start/end points, directions and road type. We collect all road vectors to the map token set $\mathcal{V}_{\text{map}}$.

Agent Motion Tokenizer. We follow the motion tokenizer from prior works [31, 41].¹ Specifically, we segment the continuous trajectories of all agents in the dataset with a fixed time span of 0.5 seconds. Then we use k-disks algorithm [31] to cluster these trajectories into the set of motion vocabulary $\mathcal{V}_{\text{motion}}$. Finally, at every 0.5-second interval, we convert the continuous trajectory into a discrete token with the index of its nearest neighbor in the motion vocabulary.

Agent Pose Tokenizer. When a new agent is inserted into the scenario, its initial pose (position and heading) is given. We tokenize the pose of each inserted agent as a pair of discretized *position* and *heading* tokens. For position, we construct a grid with a radius of R , centered on the ego agent's location with x-axis aligned with ego agent's heading, resulting in position token set $\mathcal{V}_{\text{pos}}$. To get the position token, we obtain the index of the grid closest to the agent based on L2 distance. For heading, we divide the 360° range at the interval of $\Delta\theta$, resulting in heading token set $\mathcal{V}_{\text{head}}$. To get the heading token, we similarly obtain the index of the heading interval closest to the agent heading. For simplicity, we refer the pair of position and heading tokens as a *pose token*.

¹Please refer to [31] for the details of motion tokenizer and k-disks approach.

Mode Control Tokenizer. We model long-term traffic simulation as interleaved motion simulation and scenario generation. We design the control tokens $\mathcal{V}_{\text{control}}$, consists of 4 special tokens, to mark mode transition between tasks: 1) <BEGIN MOTION>: the next token is an agent motion token to simulate current existing agents; 2) <ADD AGENT>: the next token is an agent pose token to insert a new agent; 3) <KEEP AGENT>: the current agent is kept in the scenario; 4) <REMOVE AGENT>: the current agent will be removed in the next timestep. In the next section we will show how to use $\mathcal{V}_{\text{control}}$ to control the interleaved simulation process.

Tokenizing a real log into a discrete token sequence allows us to convert the complex mixture-task simulation process into a simple interleaved next token prediction task.

4.2. Interleaved Next Token Prediction

Dynamic Agent Matrix. Traffic simulation can be represented by an agent matrix shown in Figure 3. The horizontal axis represent *temporal* lifecycle of each agent: being inserted, active moving and finally exit the scenario. The length of the temporal axis equals to the rollout horizon. On the other hand, the vertical axis represent *spatial* agent layout at each timestep, where the width represent the number of active agents at each step. When a new agent is inserted, a new row is created and append to the matrix. Conversely, when a current agent gets removed, its row gets deleted from the matrix. For long-term scenarios, because agents are frequently being inserted and removed from the scenario, the number of rows of the matrix are also constantly changing. Hence, we term it the *dynamic* agent matrix.

As shown in Figure 3, we represent the long-term traffic simulation task as extending the dynamic agent matrix on different axis. Motion simulation (the upper blue flow) extends the temporal axis by adding new columns with predicted motion tokens. In contrast, scenario generation (the lower green flow) extends the spatial axis by adding new rows with pose tokens of new agents, and removing current rows of exiting agents. The control tokens determine how to interleave these two processes.

Temporal Motion Simulation. We show this process in the blue flow of Figure 3. For the i_{th} active agent at timestep t , we use its current motion token m_i^t as the query $q_{m_i}^t$ to obtain its motion feature $f_{m_i}^t$.² Specifically, we input q_m to a Temporal Attention layer to attend to the key and value of all its own past motion tokens within t_w timesteps (within the same row as the query token):

$$q_{m_i}^t = \text{MHSA}^t(q_{m_i}^t, \{k_{m_i}^{t-\tau}\}_{\tau=1}^{t_w}, \{v_{m_i}^{t-\tau}\}_{\tau=1}^{t_w}). \quad (3)$$

The output is then sent to an Agent-Agent Attention layer to

attend to all the other N^t active agents within a valid range $r^{a \leftrightarrow a}$ at the same timestep t :

$$q_{m_i}'' = \text{MHCA}^{a \leftrightarrow a}(q_{m_i}', \{k_{m_j}\}_{j=1}^{N^t}, \{v_{m_j}\}_{j=1}^{N^t}). \quad (4)$$

Finally, the query goes through a Map-Agent Attention layer to attend to the N_r precomputed map tokens within a valid range $r^{m \leftrightarrow a}$:

$$f_{m_i} = \text{MHCA}^{m \leftrightarrow a}(q_{m_i}'', \{k_{m_j}\}_{j=1}^{N_r}, \{v_{m_j}\}_{j=1}^{N_r}). \quad (5)$$

The motion head and control head separately take f_m and output the probabilities over the motion and control tokens, from which we sample a motion token and a control token for each active agent. In this subtask, we enforce the control token to be sampled from <KEEP AGENT> and <REMOVE AGENT>. If control token is <KEEP AGENT>, we add the sampled motion tokens to the next column of each agent. Otherwise, if the control token is <REMOVE AGENT>, we add this control token to the next column and discard the motion token. The above process is conducted for all the current active agents *in parallel* in training and inference. After this, we switch to the scene generation step.

Spatial Scene Generation. We show this process in the green flow of Figure 3. After each motion simulation step, we use a learnable agent query a_0 to obtain the scene generation feature f_{a_0} . Same as the motion query, the agent query is also sent through three attention layers to collect the context information. The latter two layers are the same as the motion query, while the Temporal Attention layer is replaced by a Grid Attention layer. This layer allows the agent query to attend to the occupancy grid tokens g of total size $N_g = N_p = |\mathcal{V}_{\text{pos}}|$ derived from the position tokens:

$$q_{a_0}' = \text{MHCA}^g(q_{a_0}, \Gamma(\{k_{g_j}\}_{j=1}^{N_g}), \Gamma(\{v_{g_j}\}_{j=1}^{N_g})), \quad (6)$$

where $\Gamma(\cdot)$ presents the transformation preceding the attention calculation for efficiency. We then pass q_{a_0}' through the other two layers (Eq. 4 and Eq. 5) to produce f_{a_0} . Differently, q_{a_0}' will have various visible range for the active agents $r^{q \leftarrow a}$ at current timestep and for the map tokens $r^{q \leftarrow m}$, please refer to Appendix B for more details.

Then the pose head and control head take f_{a_0} and output distributions for pose and control tokens respectively, from which we sample a pose token and a control token for each generation step (Please refer to Appendix B for more details). We enforce the control token to be sampled from <ADD AGENT> and <BEGIN MOTION> for this subtask. Controlled by <ADD AGENT>, we append a new row to the agent matrix and assign the sampled pose token to the current timestep. Then, conditioned on the all active agents, including the newly inserted one, we repeat the above step to autoregressively insert another new agents. This process is terminated when the sampled control token is <BEGIN

²We omit the t and i in this section when possible for simplicity.

MOTION>. In this case, we end the scene generation process and move to motion simulation of the next timestep. Finally, we remove any row that has a <REMOVE AGENT> token at the current timestep from the agent matrix.

4.3. Model Architecture

Token Embedding. Our model takes tokens of different modalities. To model them with a single token sequence, we take different MLP layers to embed different kinds of tokens into the same latent dimension D before entering the model. Please refer to Appendix B for more details.

Modeling Layer. Our transformer model is composed of L blocks of attention layers. As mentioned in Section 4.2, each block contains 4 attention layers: Temporal Attention, Agent-Agent Attention, Map-Agent Attention and Grid Attention. Here the first layer are implemented with multi-head self attention layer (MHSA), while the other layers are multi-head cross attention layer (MHCA). Furthermore, we apply the position-aware attention from prior works [25, 31] to explicitly model the relative positions between tokens. Please refer to the Appendix for the details.

Occupancy Grid Encoder. We obtain occupancy grid features f_g of the current scenario with the agent position tokens and map tokens via $\Gamma(\cdot)$. Specifically, given the vocabulary size N_p of position tokens $\mathcal{V}_{pos}$, we directly assign each token with occupation indication $\{0, 1\}$, leading to an agents occupancy grid $g^{1 \times N_p} \in \{0, 1\}$. We utilize the occupancy maps of agents in decoding pose token process, to make the agent query efficiently infer the spatial distributions of agents. We use an MLP Layer to convert $g^{1 \times N_p}$ to its features $f_g^{N_p \times D}$ before feed it into Grid Attention layers.

4.4. Training

Ground-truth Sequence. As described in Section 3, real-world traffic scenario log [14] naturally contains data of agent motion, insertion and removal behaviors. To train our model with the interleaved NTP problem explained in Section 4.2, we convert each ground-truth log into an ordered sequence of token labels. To this end, we enforce a specific ordering to chain tokens from different modalities. Specifically, at each timestep we arrange each type of tokens with a fixed order: 1) motion tokens from all the current agents; 2) control tokens <REMOVE AGENT> and <KEEP AGENT> for the current agents; 3) pose tokens and <ADD AGENT> for any inserted agents; 4) <BEGIN MOTION> that mark the transition to the next timestep. For the same type of tokens we order them following to the agent’s distance to the ego agent from near to far. With this rule we obtain a sequence of ground-truth (GT) token labels from each real log to train our model. Please refer to Appendix C for more details about the GT tokens.

Table 1. Short-term traffic simulation in WOSAC [14] ($\uparrow$).

Method	Composite	Kinematic	Interactive	Map
TrafficBots [40]	0.6976	0.3994	0.7103	0.8342
GUMP [11]	0.7404	0.4773	0.7872	0.8339
SMART-7M [31]	0.7521	0.4799	0.8048	0.8573
CatK [41]	0.7603	0.4611	0.8103	0.8732
InfGen	0.7514	0.4754	0.7936	0.8502

Learning Objective. As shown in Figure 2, given the GT input tokens as input, our model predicts the distributions of different kinds of tokens at the corresponding position. We then train our model with a set of standard NTP objective for each type of tokens. For example, the loss function for the motion token is:

$$\mathcal{L}_{\text{motion}} = - \sum_{t=1}^{T-1} \log p_{\theta}(m_i^{t+1} | c^{1:t}), \quad (7)$$

where T is the total number of timesteps, m_i^{t+1} is the GT motion token in the next timestep, $p_{\theta}(m_i^{t+1} | c^{1:t})$ is the model-predicted probability of the GT token. Here, $c^{1:t}$ is the ensemble of all history context the model attends to, including history motions of the same agent, positions of other agents in the current timestep, and the map. We formulate the loss function in the same way for agent pose tokens $\mathcal{L}_{\text{pose}}$ and mode control tokens $\mathcal{L}_{\text{control}}$. We also have supervision of shapes and types for those new agents. Our total training loss can be written as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{motion}} + \mathcal{L}_{\text{pose}} + \lambda_4 \mathcal{L}_{\text{control}} + \mathcal{L}_{\text{attr}}, \quad (8)$$

where $\mathcal{L}_{\text{pose}} = \lambda_2 \mathcal{L}_{\text{pos}} + \lambda_3 \mathcal{L}_{\text{head}}$ and $\mathcal{L}_{\text{attr}} = \lambda_5 \mathcal{L}_{\text{shape}} + \lambda_6 \mathcal{L}_{\text{type}}$. We directly end-to-end train InfGen with $\mathcal{L}$ on the fully tokenized dataset. During training InfGen not only learns how to conduct the two tasks respectively, but also learns to automatically and seamlessly switch between them. Please refer to Appendix D for more training explanations.

5. Experiments

In this section, we validate our work from two aspects: 1) *How does InfGen compare to the SOTA baselines on conventional short-term traffic simulation task in standard benchmarks?* 2) *How does InfGen perform on our mainly introduced long-term traffic simulation task?*

Dataset. We train and validate InfGen on Waymo Open Motion Dataset (WOMD) [9], with $\sim 480K$ scenarios for training and $\sim 44K$ scenarios for validation. Each scenario consists of a $T = 9.1$ s recorded rollout, with the first $T_H = 1.1$ s historical frames and the subsequent 8 s future frames.

Training. To train InfGen, we use a total batch size of 8 on 8 NVIDIA A5000 GPUs with the AdamW optimizer

Table 2. Long-term traffic simulation evaluation ($\uparrow$) on WOMB validation split measured by the metrics introduced in Sec. 5.2.

Method	Composite	Kinematic	Interactive	Map-based	overall	Placement-based			
						N_+	N_-	D_+	D_-
SMART-7M [31]	0.6519	0.5839	0.7542	0.8102	0.4324	0.5713	0.4964	0.3371	0.3248
CatK [41]	0.6584	0.5850	0.7584	0.8186	0.4424	0.5842	0.5233	0.3371	0.3248
InfGen	0.6606	0.5966	0.7619	0.8087	0.4542	0.6273	0.5635	0.3169	0.3092

and cosine annealing learning rate scheduler. The initial learning rate is 0.0005. For the loss function in Eq. 8, we set $\lambda_1 = \lambda_3 = 1$, $\lambda_2 = \lambda_4 = 10$, $\lambda_5 = 0.2$, $\lambda_6 = 5$.

5.1. WOMB Sim Agent Challenge

We first evaluate InfGen on the standard *short-term* traffic simulation task in WOSAC [14]. For this setting, we enforce InfGen to skip all the scene generation steps, and predict the rollout for only 8s. We then evaluate the rollouts under the official WOSAC metrics and compare the results with the top-performing methods in Table. 1. The results show InfGen performs very competitive even under the short-term setting without any task-specific tuning, achieving similar performance to SOTA [41] and outperforms strong models [11, 40].

5.2. Long-term Traffic Simulation Setup

Rollout Setup. To be compatible with the map size of WOMB, we set the total rollout duration of our long-term traffic simulation experiments as $T' = 31.1$ s with FPS = 10. The first 1.1 s corresponds to the historical segment, from which we simulate 300 future steps.

WOSAC Metric Adaption. The standard WOSAC metric [14] evaluates short-term traffic simulation by comparing simulated rollouts directly with ground-truth (GT) logs over an 8-second horizon. Specifically, it computes 9 metrics that assess aspects such as kinematic realism, interaction realism, and map adherence, assuming a one-to-one correspondence between simulated and logged agents.

However, in our long-term simulation setting, the rollout duration extends significantly beyond the standard 8-second window, reaching up to 30 seconds. This creates two critical challenges: (1) there is no direct one-to-one correspondence between simulated agents and logged agents over the entire long-term rollout, as agents dynamically enter and exit the scene; (2) the evaluation window (8 seconds in WOSAC) is shorter than our simulation horizon (30 seconds). Therefore, we need to adapt the original WOSAC metric to suit our long-term simulation setting.

Specifically, we adapt the WOSAC evaluation as follows. Given a long-term simulated rollout $\sigma' = (\mathcal{M}, \mathcal{A}'_{0:T'})$, we extract short segments using a sliding window approach. Specifically, we slide a window of

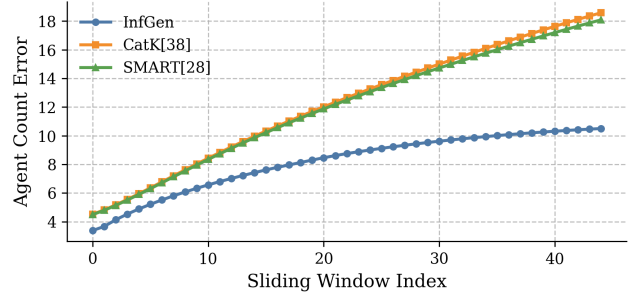


Figure 4. Agent Count Error (ACE) curves of InfGen against baselines over 30s long-term simulation rollouts.

length $T_w = T - T_H$ (matching the standard 8 s evaluation window) at a fixed interval Δt throughout the entire simulated rollout. Each sliding window generates a short-term segment $\mathcal{A}'_{t:t+T_w}$ that matches the length of standard evaluation segments. Finally, we get $\mathcal{A}'_{0:T'} = \{\mathcal{A}'_{\Delta s(i-1):\Delta s(i-1)+T_w}\}_{i=1}^P$ with the number of segments equal to P , and $\Delta s(P-1) + T_w = T'$.

Since the number of simulated agents N' in these segments may differ from the logged agents N , we cannot directly apply the original WOSAC evaluation. Instead, we compute agent-level Negative Log-Likelihood (NLL) scores for all simulated agents by evaluating their behaviors against a global distribution learned from the entire validation dataset (~ 48 K scenarios). Concretely, we first estimate empirical distributions for agent motions, interactions, and placements from the entire validation dataset, and then measure how well our simulated agent behaviors conform to these reference distributions. This modified approach ensures a fair and consistent evaluation of long-term simulation realism with varying numbers of agents and rollout durations.

Placement-based Metrics. The standard WOSAC metrics evaluate simulation realism through multiple components: *kinematic-based*, *interaction-based*, and *map-based*. These metrics are then aggregated into an overall realism composite metric using predefined weights. However, these metrics assume a fixed set of agents throughout the evaluation horizon and thus fail to capture the realism of agent insertion and removal events that are essential in long-term traffic simulations. We then propose the *placement-based* component, which comprises 4 types of statistics: the num-

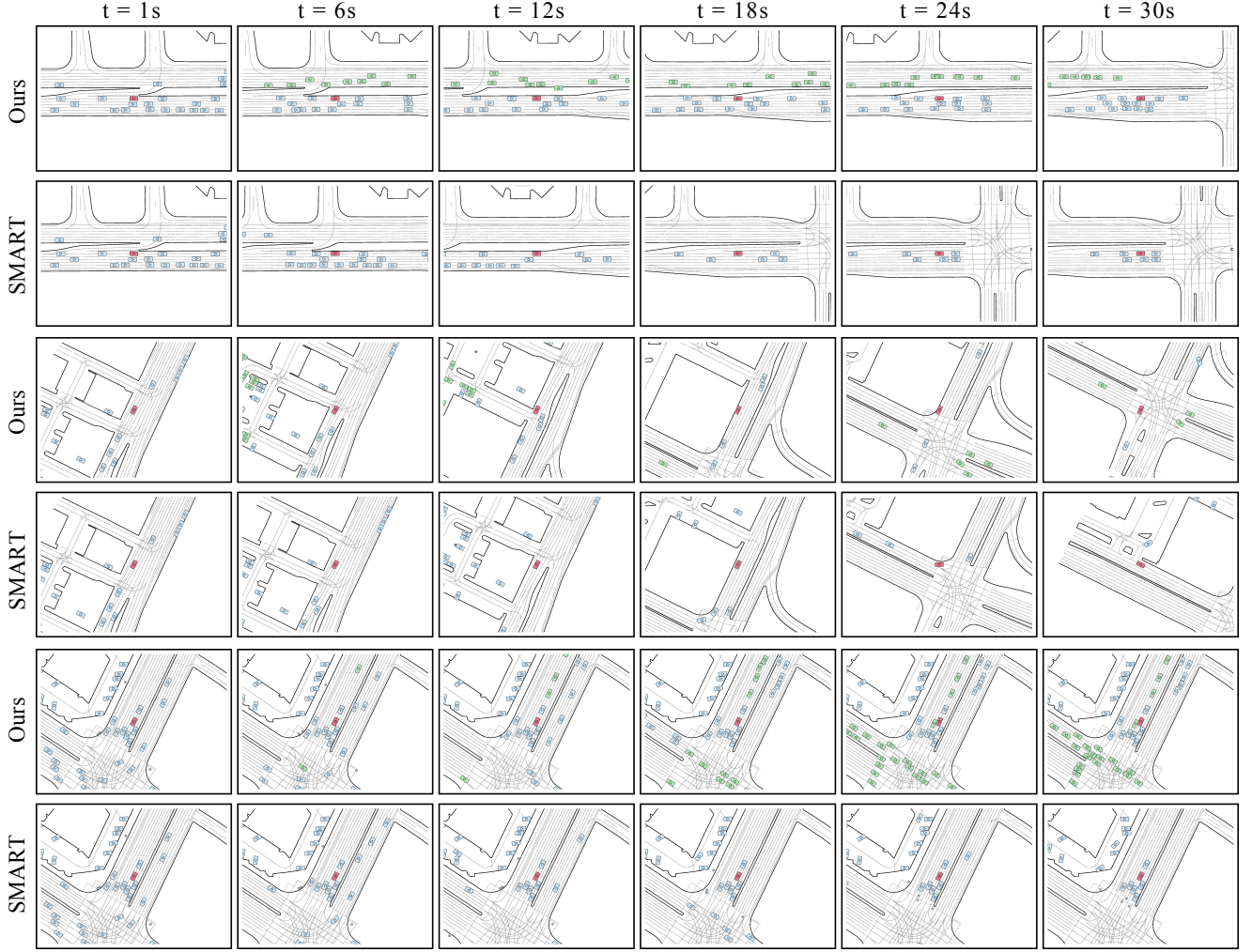


Figure 5. **Qualitative results of long-term closed-loop rollouts for 5 scenarios.** We compare rollouts of InfGen and SMART [31] here. ■ are the initially placed agents, ■ are the new agents inserted by InfGen, and ■ are the ego agents. Please refer to Supplementary Materials for more results.

ber of placement N_+ , the number of removal N_- , the distance of placement D_+ and the distance of removal D_- , where the distances here are relative to the ego agent. We aim to use the placement-based metrics to assess the realism of InfGen in modeling the entry and exit of agents during the long-term rollout. Similarly, we calculate the NLLs of the placement-based statistics under the logged distributions in the same way as the other components.

Agent Count Error Metrics. In addition to WOSAC metrics extensions, we also introduce a new Agent Count Error (ACE) metric to evaluate scene realism during long-horizon rollouts. For each sliding window over the 30 s rollout, we compute the mean absolute difference in agent count between simulation and the ground-truth distribution of the validation set. We summarize the results with two scalar metrics (lower is better): *Mean ACE* (overall error) and *ACE*

Slope (error growth rate via linear regression).

With the above setup we are now ready to evaluate different methods for long-term traffic simulation.

5.3. Long-term Traffic Simulation Evaluation

Baselines. Since existing works do not focus on long-term rollout, to fairly compare, our baselines are derived by improving upon the SOTA simulation methods SMART [31] and CatK [41]. SMART leverages vectorized map and agent trajectory data, predicting motion sequences through a decoder-only transformer architecture. CatK further introduces a closed-loop supervised fine-tuning technique and achieve better WOSAC simulation performance.

First, we extend their rollout duration to $T' = 31.1s$ and then calculate the kinematic, interactive and map-based metrics. For the placement-based metrics, we design a

Table 3. Long-term motion prediction evaluation ($\uparrow$).

Method	Composite	Kinematic	Interactive	Map
SMART-7M [31]	0.7428	0.5413	0.7626	0.8349
CatK [41]	0.7316	0.5216	0.7347	0.8495
InfGen	0.7432	0.5495	0.7685	0.8213

heuristic approach: we obtain the entered and exited agents by partitioning the distance between agents and the ego agent by the radius R , which corresponds to the *distance placement* and *distance removal* of placement-based measurements, introduced in Sec. 5.2. During the rollout process, we have the distances from each agent to the ego agent $\{\mathcal{D}_{0:T'}^i\}_{i=1}^N$. At step t , the agents that first run within the range R , whose $\mathcal{D}_t \leq R$ given $\mathcal{D}_{0:t-1} > R$, are considered as *entered* agents, while those agents with $\mathcal{D}_t > R$ and $\mathcal{D}_{0:t-1} \leq R$, are considered as *exited* agents. Throughout the experiment, we adjust R to achieve the highest placement-based score for baselines on validation set. In this case, we assign additional validity values to each agent at each timestep, thus only those valid agents will be included in baseline inferences and evaluation metrics. We follow the default settings in their papers.

Quantitative Results. Under the settings and proposed evaluation metrics in Sec. 5.2, Table. 2 shows the results of long-term traffic simulation under extended WOSAC metrics. As can be seen, InfGen achieves better realism performance than baselines. For placement-based metrics, our method significantly outperforms the baselines, demonstrating that our approach effectively models the spatial sequences of agents. The lower scores on map-based metrics may be attributed to the insertion of agents in regions outside driving lanes or near road boundaries, leading to less realistic motion trajectories.

Figure 4 shows the curve of comparison results under our introduced ACE metrics, where our method significantly outperforms prior methods: InfGen achieves Mean ACE of 8.1 while baselines (CatK [41], SMART [31]) have scores of 12.2 and 12.0, which reflects the improvements of our dynamically scenario generation; InfGen has ACE Slope: 0.15, however, baselines accumulate such errors with slopes of 0.32 and 0.31. This significantly confirms the much better long-term stability observed in Figure 5.

Qualitative Analysis. We further show InfGen through the visualizations of the long-term rollouts. As depicted in Figure. 5, the results highlight two core properties of InfGen: *long-term* and *closed-loop*. The first scenario depicts a bidirectional driving road, where InfGen successfully simulates oncoming traffic. In the third scenario, when the rollout reaches $t = 18s$, most of active agents run outside the scenario, resulting in an empty region, as reflected by prior works. Under the control of InfGen,

new agents enter the scenario, allowing continued agent-agent and agent-map interactions with high realism. While in baselines, those regions around the ego agent remains empty. Moreover, from the forth scenario, we observe our model can place new agents not only on driving roads but also in parking lots or other open areas.

Motion-only Analysis. In Section 3 we formulate the long-term traffic simulation task as a product of motion simulation and scene generation (Equation 2). Here we investigate the performance of different models in long-term traffic simulation when we only consider motion simulation. Specifically, we disable agent insertion and removal for all the methods, and simply let them rollout for 30 seconds. We then evaluate the rollouts with the adapted WOSAC metrics. As shown in Table 3, all the compared methods have very similar performance. This indicate that simply extending rollout horizon will not reveal important aspects of long-term traffic simulation, like the empty scenarios we see for SMART rollouts in Figure 5.

More Experiments and Analysis. Additional experiments and results are provided in Sec. ?? of the Supplementary Materials, including further analysis on long-term traffic simulation, as well as ablation studies on our various tokens. Please refer to it for more details.

6. Conclusion

In this work we propose InfGen, a unified next-token prediction model for long-term traffic simulation. InfGen learns to automatically switch between temporal motion simulation and spatial scene generation. Our experiments show InfGen significantly outperforms prior methods in long-term simulation, while keeping strong performance on standard short-term simulation. We believe InfGen is a steady step towards realistic trip-level traffic simulation.

Limitations and Future Works. The main limitation of this paper is that we did not evaluate InfGen under a real trip-level rollout duration (> 5 minutes). Our main constraint is the map areas available in WOSAC scenarios are too small: even in our 30s rollout the ego agent often drive to areas where no map is available. We plan to further explore with suitable data. Another limitation is that InfGen is trained purely with supervised learning on logged real-world data, which may lead to failures due to overfitting and the model capturing spurious causal relationships. Given the interactive nature of this task and its autoregressive generation process, in future work we plan to explore interactive reinforcement learning to further encourage realistic agent interactions and scenario generation through environment feedback [5, 26].

References

- [1] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric M Wolff, Alex H Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. 1, 2
- [2] Sergio Casas, Cole Gulino, Shuai Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *European Conference on Computer Vision (ECCV)*, 2020. 2
- [3] Chameleon Team. Chameleon: Mixed-Modal Early-Fusion Foundation Models. *arXiv preprint arXiv:2405.09818*, 2024. 2
- [4] Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, Wei Zhan, and Manmohan Chandraker. Safe-sim: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries. *arXiv preprint arXiv:2401.00391*, 2024. 2
- [5] Keyu Chen, Wenchao Sun, Hao Cheng, and Sifa Zheng. Rift: Closed-loop rl fine-tuning for realistic and controllable traffic simulation, 2025. 8
- [6] Kashyap Chitta, Daniel Dauner, and Andreas Geiger. Sledge: Synthesizing driving environments with generative models and rule-based traffic. In *European Conference on Computer Vision*, pages 57–74. Springer, 2024. 2
- [7] Marco Cusumano-Towner, David Hafner, Alex Hertzberg, Brody Huval, Aleksei Petrenko, Eugene Vinitsky, Erik Wijmans, Taylor Killian, Stuart Bowers, Ozan Sener, Philipp Krähenbühl, and Vladlen Koltun. Robust autonomy emerges from self-play. *arXiv preprint arXiv:2502.03349*, 2025. 1, 2
- [8] Wenhao Ding, Yulong Cao, Ding Zhao, Chaowei Xiao, and Marco Pavone. Realgen: Retrieval augmented generation for controllable traffic scenarios. *arXiv preprint arXiv:2312.13303*, 2023. 2
- [9] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. 1, 5
- [10] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3567–3575, 2023. 2
- [11] Yihan Hu, Siqi Chai, Zhening Yang, Jingyu Qian, Kun Li, Wenxin Shao, Haichao Zhang, Wei Xu, and Qiang Liu. Solving motion planning tasks with a scalable generative model. In *European Conference on Computer Vision*, pages 386–404. Springer, 2024. 2, 5, 6
- [12] Siqi Kou, Jiachun Jin, Chang Liu, Ye Ma, Jian Jia, Quan Chen, Peng Jiang, and Zhijie Deng. Orthus: Autoregressive Interleaved Image-Text Generation with Modality-Specific Heads. *arXiv preprint arXiv:2412.00127*, 2024. 2
- [13] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1
- [14] Nico Montali, John Lambert, Paul Mouglin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, et al. The waymo open sim agents challenge. *Advances in Neural Information Processing Systems*, 36:59151–59171, 2023. 2, 3, 5, 6, 13, 17
- [15] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratharth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 2
- [16] Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajenglish: Traffic modeling as next-token prediction. *arXiv preprint arXiv:2312.04535*, 2023. 2
- [17] Davis Rempe, Jonah Philion, Leonidas J Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17284–17294, 2022. 2
- [18] Luke Rowe, Roger Girgis, Anthony Gosselin, Bruno Carrez, Florian Golemo, Felix Heide, Liam Paull, and Christopher Pal. Ctrl-sim: Reactive and controllable driving agents with offline reinforcement learning. *arXiv preprint arXiv:2403.19918*, 2024. 2
- [19] Luke Rowe, Roger Girgis, Anthony Gosselin, Liam Paull, Christopher Pal, and Felix Heide. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. *arXiv preprint arXiv:2503.22496*, 2025. 2
- [20] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision (ECCV)*, 2020. 2
- [21] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023. 2
- [22] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10400–10409, 2021. 2
- [23] Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Scenegen: Learning to generate realistic traffic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [24] Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation. In *7th Annual Conference on Robot Learning (CoRL)*, 2023. 2
- [25] Shuhan Tan, Boris Ivanovic, Yuxiao Chen, Boyi Li, Xinshuo Weng, Yulong Cao, Philipp Krähenbühl, and Marco Pavone. Promptable closed-loop traffic simulation. In *8th Annual Conference on Robot Learning (CoRL)*, 2024. 2, 5, 16

- [26] Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025. 8
- [27] Shuhan Tan, John Lambert, Hong Jeon, Sakshum Kulshrestha, Yijing Bai, Jing Luo, Dragomir Anguelov, Mingxing Tan, and Chiyu Max Jiang. Scenediffuser++: City-scale traffic simulation via a generative world model. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 1570–1580, 2025. 2
- [28] Changyao Tian, Xizhou Zhu, Yuwen Xiong, Weiyun Wang, Zhe Chen, Wenhai Wang, Yuntao Chen, Lewei Lu, Tong Lu, Jie Zhou, Hongsheng Li, Yu Qiao, and Jifeng Dai. Mm-interleaved: Interleaved image-text generative modeling via multi-modal feature synchronizer. *arXiv preprint arXiv:2401.10208*, 2024. 2
- [29] Jingkan 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9909–9918, 2021. 2
- [30] Cathy Wu, Abdelrahman Kreidieh, Karthik Parvate, Eugene Vinitzky, and Alexandre M Bayen. Flow: A modular learning framework for mixed autonomy traffic. In *IEEE Transactions on Robotics*, pages 1677–1689, 2021. 1
- [31] Wei Wu, Xiaoxin Feng, Ziyang Gao, and Yuheng KAN. Smart: Scalable multi-agent real-time motion generation via next-token prediction. In *Advances in Neural Information Processing Systems*, pages 114048–114071. Curran Associates, Inc., 2024. 1, 2, 3, 5, 6, 7, 8, 11, 13, 15
- [32] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 2
- [33] Yuting Xie, Xianda Guo, Cong Wang, Kunhua Liu, and Long Chen. Advdiffuser: Generating adversarial safety-critical driving scenarios via guided diffusion. *arXiv preprint arXiv:2410.08453*, 2024. 2
- [34] Zixun Xie, Sicheng Zuo, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, Jie Zhou, Jiwen Lu, and Shanghang Zhang. Gpd-1: Generative pre-training for driving, 2024. 17
- [35] Chejian Xu, Ding Zhao, Alberto Sangiovanni-Vincentelli, and Bo Li. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. *AdvML-Frontiers 2023*, 2023. 2
- [36] Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Bits: Bi-level imitation for traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, page 2929–2936. IEEE, 2023. 2
- [37] Zhiyang Xu, Minqian Liu, Ying Shen, Joy Rimchala, Jiaxin Zhang, Qifan Wang, Yu Cheng, and Lifu Huang. Modality-specialized synergizers for interleaved vision-language generalists. In *International Conference on Learning Representations (ICLR)*, 2025. 2
- [38] Xiuyu Yang, Zhuangyan Zhang, Haikuo Du, Sui Yang, Fengping Sun, Yanbo Liu, Ling Pei, Wenchao Xu, Weiqi Sun, and Zhengyu Li. Rmmdet: Road-side multitype and multi-group sensor detection system for autonomous driving. *arXiv preprint arXiv:2303.05203*, 2023. 2
- [39] Chris Zhang, James Tu, Lunjun Zhang, Kelvin Wong, Simon Suo, and Raquel Urtasun. Learning realistic traffic agents in closed-loop. In *7th Annual Conference on Robot Learning*, 2023. 2
- [40] Zhejun Zhang, Christos Sakaridis, and Luc Van Gool. Trafficbots v1. 5: Traffic simulation via conditional vaes and transformers with relative pose encoding. *arXiv preprint arXiv:2406.10898*, 2024. 2, 5, 6
- [41] Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3, 5, 6, 7, 8, 15, 16
- [42] Jianbo Zhao, Jiaheng Zhuang, Qibin Zhou, Taiyu Ban, Ziyao Xu, Hangning Zhou, Junhe Wang, Guoan Wang, Zhiheng Li, and Bin Li. Kigras: Kinematic-driven generative model for realistic agent simulation. *arXiv preprint arXiv:2407.12940*, 2024. 2
- [43] Tianyang Zhao, Yuke Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2
- [44] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. In *Conference on Robot Learning*, pages 144–177. PMLR, 2023. 2
- [45] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3560–3566. IEEE, 2023. 2
- [46] Zikang Zhou, Haibo Hu, Xinhong Chen, Jianping Wang, Nan Guan, Kui Wu, Yung-Hui Li, Yu-Kai Huang, and Chun Jason Xue. Behaviorgpt: Smart agent simulation for autonomous driving with next-patch prediction. *arXiv preprint arXiv:2405.17372*, 2024. 2