# LANGTRAJ: Diffusion Model and Dataset for Language-Conditioned Trajectory Simulation

Wei-Jer Chang[1] Wei Zhan[1] Masayoshi Tomizuka[1] Manmohan Chandraker[2,3]
Francesco Pittaluga[2]

[1] UC Berkeley [2] NEC Labs America [3] UC San Diego

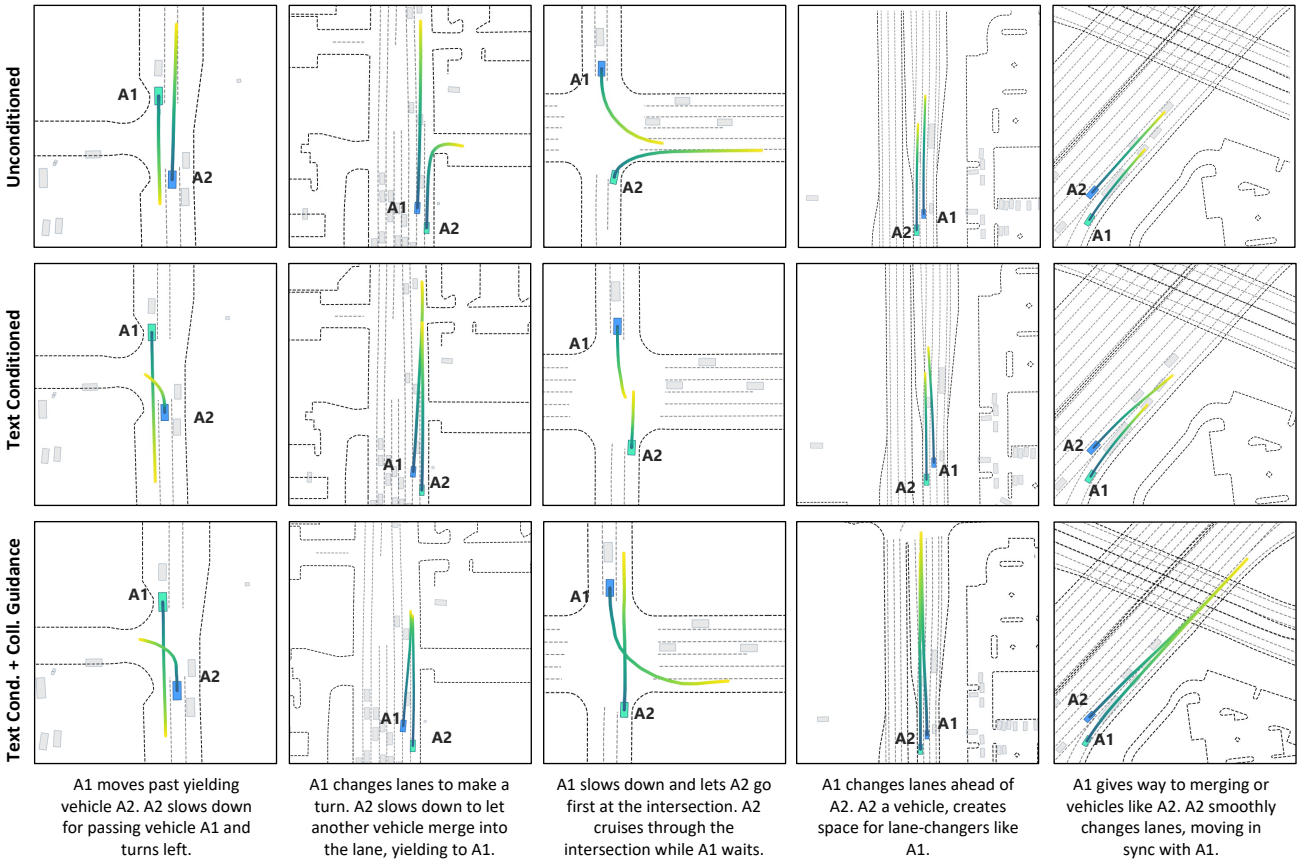


Figure 1. **Unconditioned (top), Text-Conditioned (mid), and Safety-Critical + Text-Conditioned (bot) Simulations by LANGTRAJ.** Adversarial collision guidance is applied in conjunction with text conditioning to generate the safety-critical scenarios shown in the bottom row, where the dark blue car serves as the adversarial agent. Language annotations are from the test set of INTERDRIVE.

## Abstract

*Evaluating autonomous vehicles with controllability allows for scalable testing in counterfactual or structured settings, improving both efficiency and safety. We introduce* LANGTRAJ*, a language-conditioned scene-diffusion model that simulates the joint behavior of all agents in traffic scenarios. By conditioning on natural language inputs,* LANGTRAJ *enables flexible and intuitive control over interactive behaviors, generating nuanced and realistic scenarios. Unlike prior approaches that rely on domain-specific guidance functions,* LANGTRAJ *incorporates language conditioning during training for more intuitive traffic simulation control. In addition, we propose a novel closed-loop*

*training strategy for diffusion models to enhance realism in closed-loop simulation. To support language-conditioned simulation, we develop a scalable pipeline for annotating agent-agent interactions and single-agent behaviors, which we use to develop* INTERDRIVE*, a large-scale dataset offering diverse and interactive labels for training language-conditioned diffusion models. Validated on the Waymo Open Motion Dataset,* LANGTRAJ *demonstrates strong performance in both realism, language controllability, and language-conditioned safety-critical simulation, establishing a new paradigm for flexible and scalable autonomous vehicle testing. Project website: https://langtraj.github.io/.*

## 1. Introduction

Traffic simulation is essential for the safe and scalable development of autonomous vehicles (AVs). By simulating interactions among multiple traffic participants, it enables AVs to handle a wide range of realistic scenarios. This approach is critical for three main reasons: 1) it accelerates development by enabling large-scale, repeatable testing in controlled environments, 2) it provides structured testing to validate vehicle behavior across diverse conditions, and 3) it allows AVs to train for real-world complexities, improving safety and reliability. A key aspect of simulation is *controllability*: the ability to model the interactive behavior of other road users, which AVs must learn to navigate and respond to safely and effectively.

Traditionally, structured testing of AVs has relied on manually designed scenarios to simulate failure cases or counterfactual situations, such as collisions or typical interactive behaviors. While effective for targeted testing, this approach is inherently limited in scalability. Recent advances in diffusion-based generative models demonstrate strong capabilities in simulating complex distributions with flexible controllability [2, 35], though this often depends on domain-specific heuristic guidance functions constructed based on human knowledge *post-training*. Our key insight is to **leverage language conditioning during training** to directly learn semantics from the data distribution, enabling users—including non-experts—to generate counterfactual scenarios with ease. By conditioning on natural language, we can significantly expand the range of possible scenarios, creating a more capable model as the data scales and allowing for diverse driving behaviors. Note that direct conditioning on language is orthogonal to guidance functions, allowing us to combine the strengths of both human knowledge and data-driven insights for more diverse simulations.

We present LANGTRAJ, a language-controlled diffusion model that generates realistic and controllable trajectories for AV simulation by conditioning on natural language prompts. This approach enables LANGTRAJ to respond to diverse instructions such as "yield to the right" or "merge left," capturing complex interactive behaviors for counterfactual testing across varied driving scenarios.

Achieving this functionality requires overcoming two key challenges: developing a model that can effectively condition on diverse user inputs during closed-loop simulation and acquiring high-quality data with natural language labels. To address these challenges, we first introduce a scene-diffusion model, which jointly models multi-agent behaviors while conditioning on text inputs, enabling flexible and scalable AV testing. We propose a novel *closed-loop training strategy* for diffusion models to improve closed-loop realism and reduce error accumulation during iterative rollouts. Contrary to prior diffusion model works [2, 10, 35] that adopt inference techniques such as constraint and guidance methods to steer predictions, which can slow inference and require careful balancing of multiple objectives, we introduce a closed-loop training strategy that explicitly enhances model stability and realism during closed-loop simulation. Additionally, guidance methods can still be applied post-training to refine behavior further if needed.

To enable language-conditioned simulation, we introduce INTERDRIVE, a comprehensive dataset with 150k human-labeled annotations, specifically focusing on *interactive* agent-agent behaviors (e.g., merging, yielding, and passing) with rich language annotations. In addition, we include heuristic-based labels for single-agent behaviors such as directional intent and lane changes. This dataset ensures high-quality annotations for interaction modeling, providing the necessary diversity and richness for training models that capture both interactive and individualistic driving behaviors.

Our contributions are as follows:

- **LANGTRAJ: The First Diffusion Model Directly Conditioned on Language for Interactive Simulation.** We introduce LANGTRAJ, the first diffusion-based trajectory generation model that directly conditions on natural language inputs for interactive simulation.
- **Novel Closed-Loop Training Strategy for Diffusion Models.** We propose a novel closed-loop training strategy explicitly tailored for diffusion models to enhance stability and realism during closed-loop simulation.
- **INTERDRIVE Dataset.** We present INTERDRIVE, a new dataset of 150k human-labeled interactive traffic scenarios, supplemented with heuristically labeled single-agent behaviors, enabling scalable training of language-conditioned simulation models.

## 2. Related Work

### 2.1. Traffic Simulation and Controllable Diffusion

Traffic simulation methods are either heuristic-based or learning-based. Heuristic models like IDM [28] rely on predefined rules but struggle with real-world accuracy.

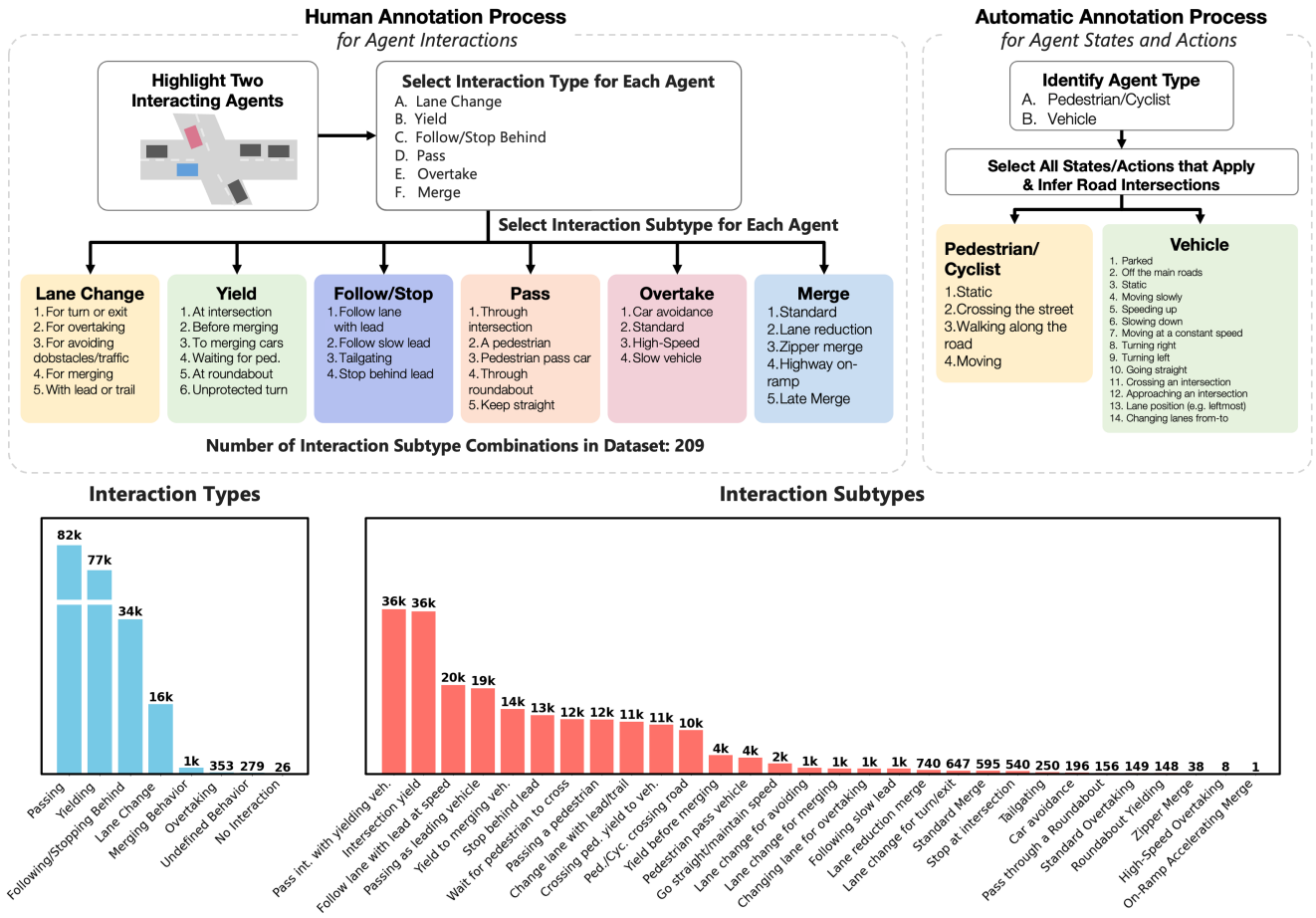


Figure 2. **Overview of INTERDRIVE dataset.** INTERDRIVE captures nuanced agent-agent interactions in real-world driving contexts. It includes human-labeled traffic interaction annotations from Waymo Motion and NuPlan datasets, along with single-agent behavioral labels generated through heuristic annotations, offering a comprehensive view of agent actions and interactions in diverse traffic scenarios. The top part of the figure shows the annotation processes for the human and heuristic annotations. The bottom part of the figure shows the counts of each interaction and interaction subtypes in INTERDRIVE.

Learning-based approaches [17, 24, 25, 29] leverage real-world data for more natural behavior. TrafficSim [24] employs a variational autoencoder for scene-level simulation, while BITS [29] combines goal inference with imitation learning. The Waymo SimAgents challenge [14] highlights the need for realistic driving distributions, but controllable scenario generation remains underexplored.

Controllable diffusion models have advanced traffic simulation [2, 9, 10, 34, 35]. Diffusion-ES [30] combines evolutionary search with diffusion models. CTG [35] introduces test-time guidance, while SAFE-SIM [2] models adversarial behaviors. CTG++ [34] enhances controllability with GPT-4 [16]-generated guidance, and SceneDiffuser [10] improves inference efficiency. However, none train with direct language conditioning.

## 2.2. Language in Autonomous Driving

LLMs have enabled language integration into autonomous driving. Early datasets [18, 31] focus on spatial annotations, while DriveVLM's Graph VQA explores vision-language reasoning. WOMD-Reasoning [12] introduces 409K QAs on traffic-rule interactions, and ProSim-Instruct-520k [27] pairs 10M Llama3-70B-generated prompts with 520K scenarios from the Waymo Open Dataset. In contrast, INTERDRIVE is directly constructed from the interactive subset of the Waymo Open Dataset, ensuring a targeted selection of interactive scenarios. Additionally, our annotations are collected from human experts rather than LLMs, focusing specifically on high-quality interactive behavior labeling.

Language-conditioned simulation methods vary. LCTGen [26] generates trajectories in an *open-loop* manner based on discrete attributes, while ProSim [27] uses an autoregressive framework for goal-driven simulation. We

adopt a diffusion-based approach for greater flexibility beyond the training distribution, enabling more adaptive and interactive simulations, as discussed in Sec. 6.5.

## 3. INTERDRIVE Dataset

Natural language conditioning for traffic simulation relies heavily on the scale and quality of the underlying data. To address this, we introduce INTERDRIVE, a central contribution of this work, designed to capture nuanced agent-agent interactions in real-world driving contexts. Our dataset is constructed through an extensive human-labeling effort on the Waymo Motion [3] and NuPlan [1] datasets, with additional single-agent behavioral labels generated via heuristic annotations.

Unlike previous datasets such as ProSimInstruct [27], which provide general trajectory labels, we focus specifically on *interactive agents*, ensuring that agent-agent interactions—such as merging, yielding, and passing—are explicitly labeled. This emphasis enables more realistic and flexible training of language-conditioned diffusion models for diverse traffic scenarios.

### 3.1. Dataset Composition

The INTERDRIVE dataset includes annotations from both the Waymo [3] and NuPlan [1] datasets, covering a diverse range of environments, agent types, and interaction scenarios. For Waymo, 125k scenes were annotated with human-generated interaction labels and 405k scenes annotated with heuristic labels. For nuPlan, 12k scenes were annotated with interaction labels and 150k scenes annotated with heuristic labels, each lasting 15 seconds.

### 3.2. Human-Labeling Process

To ensure high-quality and efficient interaction annotations, our labeling protocol was carefully designed to leverage human expertise in defining complex driving behaviors. Labelers were provided with top-down video representations of each scene, enabling them to accurately observe and annotate agent interactions. Using these visualizations, we developed a scalable and efficient pipeline for annotating complex interactions through a structured multi-choice question framework. The process consists of the following steps:

1. *Identify Interacting Agents:* For Waymo, we used the interacting agent IDs provided by the dataset. For nuPlan, we asked labelers to manually identify one agent within each scene that exhibits an interaction with the ego vehicle.
2. *Classify Interaction Type:* Labelers select the primary behavior of each interaction agent, choosing from six interaction types: (1) Lane Changing, (2) Following/Stopping Behind, (3) Yielding, (4) Passing, (5) Overtaking, and (6) Merging.
3. *Classify Interaction Subtypes:* For each chosen interaction type, labelers provide a more granular interaction subtype, capturing specifics like "Changing lane for overtaking," and "Intersection yielding". A full list of interaction subtypes is provided in Sec. E and in Fig. 2.

As shown in Fig. 2, most real-world driving behaviors can be effectively captured by our carefully designed interaction types, providing a comprehensive framework for annotating complex agent interactions.

### 3.3. Heuristic Annotation Process

In addition to interaction labels, INTERDRIVE contains labels for single-agent states/actions, which were generated automatically using a set of carefully calibrated heuristics. For the pedestrians and cyclists labels, we selected all that applied from the following list: static, crossing the street, walking along the road, and moving. For the vehicle labels, we selected all that apply from the following list: parked, off the main roads, static, moving slowly, speeding up, slowing down, moving at a constant speed, turning right, turning left, going straight, crossing an intersection, approaching an intersection, lane position (e.g. rightmost), changing lanes from-to (e.g. middle-to-rightmost). This enabled comprehensive behavioral modeling in scenarios with both explicit and implicit agent interactions.

Our human-annotated dataset provides rich annotations for interactive simulation and behavior studies, enabling in-depth analysis of agent interactions and unlocking future possibilities for language-to-simulation research.

## 4. Problem Formulation

In traffic simulation, we model behaviors of $N$ agents through a centralized function $g_\theta$, enabling realistic and controllable agent actions via language-conditioned commands $\mathbf{e}_{\text{lang}}$. The joint agent state at each timestep $t$ is $\mathbf{s}_t = [\mathbf{s}_t^1, \ldots, \mathbf{s}_t^N]$ with actions $\mathbf{a}_t = [\mathbf{a}_t^1, \ldots, \mathbf{a}_t^N]$, transitioning via $f$ based on unicycle dynamics: $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t)$.

Each agent shares context $\mathbf{c}_t$, including HD map $I$ and historical states $\mathbf{S}_{t-T_{\text{hist}}:t}$. The function $g_\theta$ generates future trajectories $\{\mathbf{s}_{t:t+T}^i\}_{i=1}^N$ based on $(\mathbf{c}_t, \mathbf{e}_{\text{lang}})$, trained on real-world data to ensure realism and flexibility to user-defined scenarios.

We use diffusion models to produce text-conditioned trajectories, reversing a forward noising process from real trajectories $\tau_0 \sim q(\tau_0)$ into noisy sequences $(\tau_1, \ldots, \tau_K)$ via Gaussian noise:

$$q(\tau_{1:K}|\tau_0) := \prod_{k=1}^{K} q(\tau_k|\tau_{k-1}),$$

$$q(\tau_k|\tau_{k-1}) := \mathcal{N}(\tau_k; \sqrt{1-\beta_k}\tau_{k-1}, \beta_k \mathbf{I}).$$

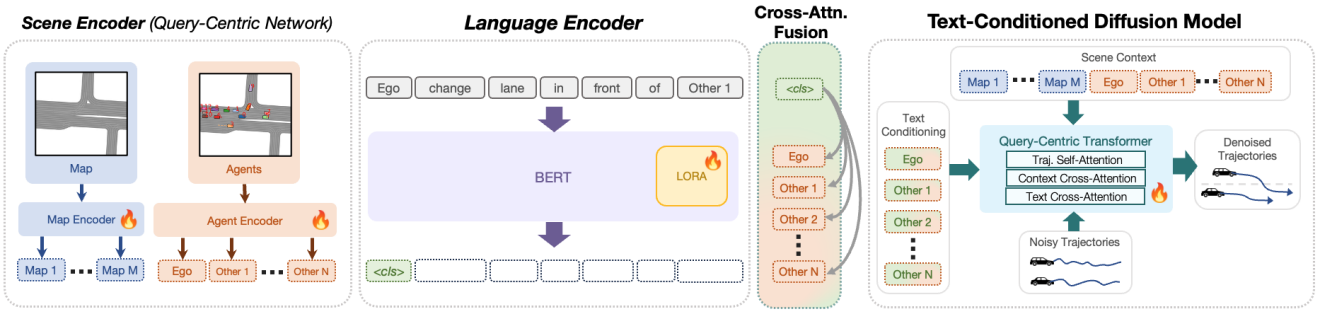


Figure 3. **Overview of LANGTRAJ.** We introduce LANGTRAJ, a novel language-controlled diffusion-based model for trajectory simulation that incorporates HD maps, agent histories, and text descriptions, enabling behaviorally nuanced trajectory generation.

The model learns to denoise $\tau_K$ back to $\tau_0$, integrating text encoding $\mathbf{e}_{\text{text}}$ to influence mean predictions:

$$p_\theta(\tau_{k-1}|\tau_k, \mathbf{c}, \mathbf{e}_{\text{lang}}) := \mathcal{N}(\tau_{k-1}; \mu_\theta(\tau_k, k, \mathbf{c}, \mathbf{e}_{\text{lang}}), \Sigma_k).$$

This enables generation of scene- and text-aligned future trajectories.

## 5. LANGTRAJ

We introduce LANGTRAJ, a scene-diffusion model designed to capture the joint distribution of interactive behaviors among all agents within multi-agent environments. To achieve this, LANGTRAJ comprises two key components: 1) an encoder that effectively represents the scene context, including map features and historical agent behaviors, and 2) a denoiser capable of jointly predicting future agent behaviors while providing flexibility and user-driven control via natural language inputs. LANGTRAJ supports trajectory generation conditioned on textual prompts and diffusion guidance, ensuring both flexibility and precise control. Additionally, we propose a novel algorithm for closed-loop training of diffusion models, further enhancing the model's performance and applicability in closed-loop simulation settings.

### 5.1. Architecture

#### 5.1.1. Scene Encoder

Inspired by [22, 36], we adopt a query-centric approach combined with Graph Neural Networks (GNNs) to model spatiotemporal relationships in the scene. A key component of this approach is the attention mechanism, which encodes relative spatial and temporal information, capturing interactions and dependencies between scene elements. The encoder operates in a scene-independent local coordinate system. Each scene element extracts features within its own local reference frame, independent of global coordinates or the ego vehicle's position. This formulation allows for symmetric encoding across agents without being affected by variations in absolute positioning. Through this process, the encoder produces an embedding $\mathbf{z}^i_{\text{enc}} = E_{enc}(I, \mathbf{S}_{t-T_{\text{hist}}:t})$ for each agent $i$.

#### 5.1.2. Language Encoder

The Language Encoder module extracts agent-specific embeddings from natural language inputs and integrates them with the scene's spatiotemporal context (see Fig. 3). This module serves two key functions: 1) providing explicit agent-specific conditioning and 2) encoding spatiotemporal data to capture agent interactions.

To achieve agent-specific conditioning, we first process the input sentence through a language model, where agent roles are explicitly *rephrased* for direct conditioning. Specifically, for each agent, its role in the sentence is labeled as the "target agent," while other agents are designated as "other agent1," "other agent2," and so forth. This rephrasing ensures clear distinctions in agent roles within the sentence. After processing through the language model, we obtain a sentence embedding $\mathbf{e}_{\text{lang}}$ that captures the overall context.

We define the language encoder function $E_L$ to integrate the language and scene context. Specifically, $E_L$ combines the language-conditioned sentence embedding $\mathbf{e}_{\text{lang}}$ with each agent's context embedding $\mathbf{z}^i_{\text{enc}}$ from the Scene Encoder:

$$\mathbf{z}^i_{\text{lang}} = E_L(\mathbf{e}_{\text{lang}}, \mathbf{z}^i_{\text{enc}}),$$

where $\mathbf{z}^i_{\text{lang}}$ represents the final language-conditioned embedding for each agent $i$. This embedding incorporates both the spatial relationships within the scene and the agent-specific conditioning derived from the language input.

Our design is flexible regarding the choice of language model; following the practice of [21], we use DistillBERT [20], a smaller language encoder, to extract the <cls> token embedding as a summary of the sentence, reducing computation cost. Empirically, we find this sufficient for language conditioning. In practice, we train the language encoder end-to-end with the diffusion model using LoRA (Low-Rank Adaptation), enabling parameter-efficient fine-tuning.

**Algorithm 1** Closed-Loop Training of Diffusion Models

**Require:** Pretrained diffusion model $\theta_{\text{init}}$, dataset $\mathcal{D} = \{s_{0:T}^{(i)}\}_{i=1}^{N}$, distance metric $D(\cdot,\cdot)$, planning horizon $T_{\text{replan}}$, number of samples $M$, noise scale $\gamma$, denoising steps $K$

**Ensure:** Optimized model $\theta$

Initialize $\theta \leftarrow \theta_{\text{init}}$
**for** each trajectory $s_{0:T} \sim \mathcal{D}$ **do**
  **for** $t = 0$ **to** $T - T_{\text{replan}}$ **step** $T_{\text{replan}}$ **do**
    Extract target sequence: $s_{\text{target}} = s_{t:t+T_{\text{replan}}}$
    Sample noise level: $\tau \sim \mathcal{U}(1, K\gamma)$
    Add noise: $s_\tau = \sqrt{\alpha_\tau} s_{\text{target}} + \sqrt{1-\alpha_\tau}\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$
    Generate $M$ denoised candidates $\hat{s}^{(1:M)}$
    Select $\hat{s}^{(m^*)}$ via $\arg\min_m D(\hat{s}^{(m)}, s_{\text{target}})$
    Execute $\hat{s}^{(m^*)}$, update state
  **end for**
  Compute loss $\mathcal{L} = \|\hat{s}_{0:T} - s_{0:T}\|^2$, update $\theta$
**end for**
**return** $\theta$

#### 5.1.3. Denoiser

The denoiser consists of multiple stacked transformer blocks, each incorporating different types of attention to model complex agent behaviors and interactions. Specifically, the denoiser employs: 1) Query-centric attention across agents' future trajectories to capture inter-agent relationships, allowing each agent to attend to other agents' future actions; 2) Agent-Context Cross-Attention, where each agent attends to its context embedding $\mathbf{z}_{\text{enc}}^{i}$ generated by the Scene Encoder, ensuring that the agent's behavior aligns with the spatiotemporal context of the environment; and 3) Text-Cross Attention, where if a language description is provided, each agent also attends to the language-conditioned embedding $\mathbf{z}_{\text{lang}}^{i}$, incorporating condition from user-defined language inputs.

### 5.2. Closed-loop Training

Typically, trajectory diffusion models are trained in an open-loop fashion [2, 9, 10, 34]. However, this mismatch between training and inference can lead to distribution shifts, where the model encounters compounding errors due to deviations from the expected trajectory distribution. A common approach to mitigate this issue during inference is to apply guidance methods or hard constraints to steer model predictions toward physically feasible behaviors. However, these techniques slow down inference speed and require careful balancing of multiple objectives, limiting adaptability in real-time simulations.

In contrast, closed-loop training is a widely used technique for mitigating compounding errors in closed-loop rollouts, as seen in autoregressive models [24, 27], where past predictions are recursively used as input to better reflect real-world deployment scenarios. However, applying closed-loop training to diffusion models presents a unique challenge: the iterative denoising process involves a double for-loop, making it computationally expensive to incorporate self-generated states into training.

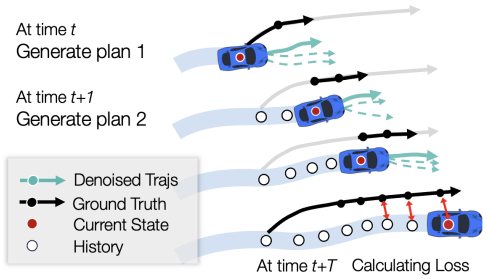


Figure 4. **Illustration of Closed-loop Training of diffusion models.** The figure demonstrates the procedure of training diffusion models in a closed-loop setting. First, the model generates multiple denoised trajectory candidates. The closest candidate to the ground truth is selected and then executed, enabling the model to experience its own distribution during training

To address this, we propose a closed-loop training tailored for diffusion models, inspired by [13, 33], which better aligns the training and inference distributions without requiring multiple denoising steps during training. Instead of training with purely ground-truth trajectories, we integrate model-generated samples into training to allow the model to experience its own predictions.

As shown in Fig. 4, our method modifies the standard diffusion training process by incorporating model-generated samples into training. At each training step, we apply forward diffusion to perturb the ground-truth trajectory segment, followed by a one-step reverse diffusion to generate multiple candidate trajectories. We then select the best candidate using a predefined distance function and execute the selected trajectory before repeating the process. The final loss is computed between the executed and ground-truth states in the global coordinate space, allowing the model to learn how to recover from accumulated errors. For a detailed breakdown of this procedure, refer to Algorithm 1. In practice, we employ a teacher-forcing strategy, where a subset of agents follows ground-truth states during training. This stabilizes learning and improves adherence to map-related metrics, as shown in Tab. 6.

### 5.3. Controllable Behavior via Diffusion Guidance

To enhance control over trajectory generation, we can incorporate two types of guidance: *classifier-based guidance*, which optimizes differentiable objectives, and *classifier-free guidance*, which blends conditional and unconditional predictions.

| Datset | Text | Meta ↑ | Kinematic ↑ | Interactive ↑ | Map ↑ | mADE ↓ |
|---|---|---|---|---|---|---|
| INTERDRIVE | × | 0.72 | 0.41 | 0.80 | 0.80 | 2.65 |
| | ✓ | 0.72 | 0.42 | 0.80 | 0.79 | 2.29 |
| ProSim-Instruct | × | 0.72 | 0.42 | 0.79 | 0.79 | 2.89 |
| | ✓ | 0.72 | 0.43 | 0.80 | 0.78 | 2.52 |

Table 1. **Evaluation of INTERDRIVE:** We train LANGTRAJ separately on the INTERDRIVE and ProSim-Instruct-520k training sets and evaluate its performance on their respective test sets.

| Method | Text | Meta ↑ | Kinematic ↑ | Interactive ↑ | Map ↑ | mADE ↓ |
|---|---|---|---|---|---|---|
| LANGTRAJ | × | 0.72 | 0.42 | 0.79 | 0.79 | 2.89 |
| | ✓ | 0.72 | 0.43 | 0.80 | 0.78 | 2.52 |
| ProSim | × | 0.69 | 0.42 | 0.73 | 0.80 | 2.73 |
| | ✓ | 0.69 | 0.42 | 0.73 | 0.81 | 2.35 |

Table 2. **Evaluation of LANGTRAJ.** Closed-loop evaluation on the ProSim-Instruct-520k test set, with and without text conditioning. We compare the performance of our method (LANGTRAJ) against the publicly released ProSim model, both trained on the ProSim-Instruct-520k training set, ensuring a fair evaluation.

For classifier-based guidance, we modify the predicted mean at each denoising step using the gradient of an objective function $J(\tau)$, steering trajectories toward desired behaviors while maintaining realism. We adopt reconstruction guidance (clean guidance) from [5, 19] to improve stability:

$$\tilde{\tau}_0 = \hat{\tau}_0 - \alpha \sum_k \nabla_{\tau_k} J(\hat{\tau}_0). \tag{1}$$

The details of the adopted guidance functions can be found in Sec. D. For classifier-free guidance, we interpolate between conditional and unconditional predictions from $g$:

$$\hat{\tau}_0 = (1 + w) \cdot g(\mathbf{e}_{\text{lang}}, \mathbf{z}_{\text{enc}}) - w \cdot g(\emptyset, \mathbf{z}_{\text{enc}}) \tag{2}$$

where $g(\mathbf{e}_{\text{lang}}, \mathbf{z}_{\text{enc}})$ includes language input $\mathbf{e}_{\text{lang}}$, while $g(\emptyset, \mathbf{z}_{\text{enc}})$ omits it. The guidance weight $w$ controls the influence of language conditioning, allowing more flexible behavior synthesis without relying on predefined objectives.

By integrating these two diffusion-based approaches, we enable more flexible and adaptable simulations, which are crucial for testing autonomous vehicles in diverse and challenging scenarios.

# 6. Experimental Results

## 6.1. Preliminaries

We validate our framework through experiments on real-world driving data from the Waymo Open Motion Dataset (WOMD) [3]. We train LANGTRAJ on the training splits of INTERDRIVE and ProSim-Instruct-520k, described in Sec. A, and evaluate its ability to generate behaviors that are both *realistic* and *controllable*.

| Method | Meta ↑ | Kinematic ↑ | Interactive ↑ | Map ↑ |
|---|---|---|---|---|
| UniMM [13] | 0.769 | 0.491 | 0.811 | 0.874 |
| SMART-tiny-CLSFT [32] | 0.762 | 0.458 | 0.811 | 0.872 |
| VBD [7] | 0.720 | 0.417 | 0.814 | 0.776 |
| LANGTRAJ | 0.719 | 0.426 | 0.795 | 0.789 |
| ProSim [27] | 0.718 | 0.401 | 0.778 | 0.822 |
| SceneDiffuser [10] | 0.703 | 0.430 | 0.776 | 0.768 |
| SceneDMF [4] | 0.628 | 0.371 | 0.683 | 0.703 |

**Blue: Diffusion-Based Methods**

Table 3. **Results on WOSAC Test Set.** LANGTRAJ performs competitively among the best diffusion-based approaches, such as VBD [7] and SceneDiffuser [10].

| Text Conditioning | Meta ↑ | Kinematic ↑ | Interactive ↑ | Map ↑ | mADE ↓ |
|---|---|---|---|---|---|
| None | 0.72 | 0.41 | 0.80 | 0.80 | 2.65 |
| Direct Condition (Ours) | 0.72 | 0.42 | 0.80 | 0.79 | 2.29 |
| LLM-Based Guidance (CTG++ Style) | 0.70 | 0.42 | 0.75 | 0.80 | 2.70 |

Table 4. **Text Conditioning Evaluation.** We compare our proposed direct text conditioning method to the LLM-based guidance method proposed by CTG++ [34].

**Metrics.** For realism (denoted "Meta" in the tables), we follow the Waymo Open Sim Agent Challenge (WOSAC), which evaluates the distributional realism of generated trajectories [14]. The challenge assesses how well joint simulation rollouts recover held-out ground truth behavior across multiple aspects, including kinematics, agent interactions, and map adherence. Further details can be found in Sec. A.5. To evaluate **controllability**, we condition the model on ground-truth future behavior descriptions and measure the improvement in minADE compared to the unconditional model. This comparison quantifies the model's ability to align generated behaviors with user-specified input when provided with behavior descriptions.

## 6.2. Evaluation of INTERDRIVE

We train separate instantiations of LANGTRAJ on the training sets of INTERDRIVE (InterDrive) and ProSim-Instruct-520k, performing closed-loop evaluations with and without text conditioning. As shown in Tab. 2, language conditioning on INTERDRIVE yields a 13.6% reduction in minADE while maintaining a meta-realism score of 0.72, indicating that our language annotations provide useful supervisory signals. A similar improvement is observed on ProSim-Instruct-520k (12.8% minADE reduction). Notably, INTERDRIVE focuses on interactive agent-agent behaviors, whereas ProSim-Instruct primarily consists of single-agent maneuvers.

## 6.3. LLM-based Guidance v.s Direct Conditioning

In Tab. 4, we compare our proposed direct language conditioning method with the LLM-based guidance approach from CTG++ [34], which uses GPT to generate differentiable code-based guidance functions from language prompts (see Sec. A.3 for details). On INTERDRIVE scenarios, we find that LLM-based guidance leads to worse

alignment with the text prompt—even underperforming the no-text baseline. In contrast, direct conditioning produces behaviors more aligned with the ground truth (13.6% reduction) while preserving simulation realism. Direct conditioning also outperforms LLM-based guidance in counterfactual settings, as discussed in Sec. 6.5. While LLM-based guidance is effective for enforcing constraints like collision avoidance and on-road driving [34], it still struggles with interaction-level behavior descriptions, underscoring the benefit of direct language grounding.

### 6.4. Evaluation of LANGTRAJ

To enable a fair comparison with ProSim, we train LANGTRAJ on the ProSim-Instruct-520k training set and evaluate its performance against the publicly released ProSim model, which is trained on the same dataset. As shown in Tab. 2, both methods achieve comparable performance in simulation realism and gain in terms of minADE. However, LANGTRAJ employs diffusion-based modeling, which unlocks inference-time guidance and controlled sampling, offering greater flexibility beyond the training distribution that autoregressive-based methods like Prosim does not have. We discuss these advantages in Sec. 6.5.

Beyond instruction-following tasks, we evaluate LANGTRAJ on the Waymo Sim Agents Challenge Benchmark (unconditioned simulation), with results presented in Tab. 3. Compared to state-of-the-art diffusion models, LANGTRAJ performs competitively among the best diffusion-based approaches, such as VBD [7] and SceneDiffuser [10].

### 6.5. Text-Conditioned Safety-Critical Simulation

We demonstrate LANGTRAJ's ability to extend beyond the training distribution by generating safety-critical scenarios conditioned on text inputs using guided sampling, as illustrated in Tab. 5 and Fig. 1. Given the focus on collision-prone situations, standard interactive metrics are less informative; thus, we instead evaluate simulation quality using kinematics and map adherence metrics. Our findings indicate that LANGTRAJ successfully generates realistic traffic scenarios as well as targeted, rare, safety-critical events, achieving collision rates as high as 40%. Compared to LLM-based guidance, LANGTRAJ achieves a 10% higher collision rate and outperforms CTG++ across all metrics in safety-critical situations. Additionally, integrating direct text conditioning enhances map adherence even under collision-focused guidance, highlighting a promising approach that combines explicit guidance with textual inputs to improve scenario diversity and simulation control. Qualitative examples are provided in the supplementary videos.

### 6.6. Closed-loop Training

We analyze the effects of denoising steps, closed-loop training, and teacher forcing in Tab. 6. First, we find that reducing denoising steps from $K = 100$ to $K = 5$ maintains realism metrics, suggesting that trajectory simulation can be potentially effectively learned with fewer denoising steps, reducing computational cost. Next, introducing closed-loop training with teacher forcing improves both map adherence ($0.72 \rightarrow 0.79$) and meta realism ($0.68 \rightarrow 0.70$). Note that we also observe vehicles tend to slow down and drive off of the road when trained in a closed-loop manner without teacher forcing, despite achieving similar realism scores. This suggests that a portion of agents follow the ground-truth trajectory helps stabilize training, preventing the model from drifting into unrealistic behaviors

| | | COLLISION | | | |
|---|---|---|---|---|---|
| **Text Conditioning** | **Guidance** | **Rate ↑** | **Kinematic ↑** | **Map ↑** | **mADE ↓** |
| None | × | 0.04 | 0.42 | 0.81 | 3.04 |
| None | ✓ | 0.41 | 0.39 | 0.70 | 4.93 |
| Direct Condition (Ours) | ✓ | **0.43** | **0.41** | **0.74** | **4.43** |
| LLM-based Guidance (CTG++ Style) | ✓ | 0.33 | 0.37 | 0.72 | 4.67 |

Table 5. **Text-Conditioned Safety-Critical Simulation with LANGTRAJ.** We demonstrate LANGTRAJ's ability to extend beyond the training distribution by generating safety-critical scenarios conditioned on text inputs using guided sampling.

| **Setting** | **Meta ↑** | **Kinematic ↑** | **Interactive ↑** | **Map ↑** |
|---|---|---|---|---|
| Open-loop (K=100) | 0.68 | 0.42 | 0.77 | 0.73 |
| Open-loop (K=5) | 0.68 | 0.41 | 0.78 | 0.72 |
| Closed-loop | 0.69 | 0.38 | 0.78 | 0.75 |
| Closed-loop w/ Teacher | 0.70 | 0.39 | 0.78 | 0.79 |

Table 6. **Ablation Study on Closed-Loop Training.** Study conducted on validation set of WOSAC challenge.

## 7. Conclusion

LANGTRAJ advances autonomous vehicle simulation by leveraging language-conditioned diffusion models to generate diverse, behaviorally rich scenarios. Unlike prior works, it supports direct language conditioning for intuitive behavior specification and guidance-based control for counterfactual and targeted scenario generation. This flexibility makes LANGTRAJ well-suited for scalable, controllable AV simulation, enabling safety-critical testing and interactive scenario design.

A key component of our approach is INTERDRIVE, which focuses on interactive agents, providing rich human-labeled and heuristic annotations that enhance realism and diversity. By prioritizing agent-agent interactions, INTERDRIVE strengthens training for language-conditioned models. Empirical results on the Waymo Motion Dataset show that LANGTRAJ generates realistic, language-aligned trajectories while preserving simulation realism. In summary, LANGTRAJ demonstrates the potential of language-conditioned diffusion models for AV simulation by combining natural language with guidance-based control.

## Acknowledgements

This work was part of W.J. Chang's summer internship at NEC Labs America, and he is also supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 2146752. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This study was funded in part by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government via the Hong Kong Centre for Logistics Robotics. The authors would like to thank Yichen Xie and Rian Tian for their insightful discussions, and Chih-Ling Chang for her helpful suggestions and assistance with figures and presentations.

## References

[1] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex 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. 4

[2] Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, Wei Zhan, and Manmohan Chandraker. Safe-sim: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries. In *European Conference on Computer Vision*, pages 242–258. Springer, 2025. 2, 3, 6

[3] 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. 4, 7, 2

[4] Zhiming Guo, Xing Gao, Jianlan Zhou, Xinyu Cai, and Botian Shi. Scenedm: Scene-level multi-agent trajectory generation with consistent diffusion models. *arXiv preprint arXiv:2311.15736*, 2023. 7

[5] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 7

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 2

[7] Zhiyu Huang, Zixu Zhang, Ameya Vaidya, Yuxiao Chen, Chen Lv, and Jaime Fernández Fisac. Versatile scene-consistent traffic scenario generation as optimization with diffusion. *arXiv preprint arXiv:2404.02524*, 2024. 7, 8

[8] Boris Ivanovic, Guanyu Song, Igor Gilitschenski, and Marco Pavone. trajdata: A unified interface to multiple human trajectory datasets. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, New Orleans, USA, 2023. 2

[9] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9644–9653, 2023. 3, 6

[10] Chiyu Max Jiang, Yijing Bai, Andre Cornman, Christopher Davis, Xiukun Huang, Hong Jeon, Sakshum Kulshrestha, John Lambert, Shuangyu Li, Xuanyu Zhou, Carlos Fuertes, Chang Yuan, Mingxing Tan, Yin Zhou, and Dragomir Anguelov. Scenediffuser: Efficient and controllable driving simulation initialization and rollout. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2, 3, 6, 7, 8

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[12] Yiheng Li, Chongjian Ge, Chenran Li, Chenfeng Xu, Masayoshi Tomizuka, Chen Tang, Mingyu Ding, and Wei Zhan. Womd-reasoning: A large-scale language dataset for interaction and driving intentions reasoning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. 3

[13] Longzhong Lin, Xuewu Lin, Kechun Xu, Haojian Lu, Lichao Huang, Rong Xiong, and Yue Wang. Revisit mixture models for multi-agent simulation: Experimental study within a unified framework. *arXiv preprint*, arXiv:2501.17015, 2025. Version 1, 28 Jan 2025. 6, 7

[14] Nico Montali, John Lambert, Paul Mougin, 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, 2024. 3, 7, 2

[15] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 3

[16] OpenAI. Gpt-4 technical report. Technical report, 2023. Available at https://cdn.openai.com/papers/gpt-4-technical-report.pdf. 3, 1

[17] Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *International Conference on Learning Representations (ICLR 2024)*, 2024. 3

[18] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4542–4550, 2024. 3

[19] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13756–13766, 2023. 7

[20] V Sanh. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 5, 2

[21] Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections. *arXiv preprint arXiv:2403.12910*, 2024. 5

[22] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 5

[23] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 3

[24] 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. 3, 6

[25] Simon Suo, Kelvin Wong, Justin Xu, James Tu, Alexander Cui, Sergio Casas, and Raquel Urtasun. Mixsim: A hierarchical framework for mixed reality traffic simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9622–9631, 2023. 3

[26] Shuhan Tan, Boris Ivanovic, Xinshuo Weng, Marco Pavone, and Philipp Kraehenbuehl. Language conditioned traffic generation. In *Conference on Robot Learning*, pages 2714–2752. PMLR, 2023. 3

[27] Shuhan Tan, Boris Ivanovic, Yuxiao Chen, Boyi Li, Xinshuo Weng, Yulong Cao, Philipp Kraehenbuehl, and Marco Pavone. Promptable closed-loop traffic simulation. In *Conference on Robot Learning*, pages 5087–5105. PMLR, 2025. 3, 4, 6, 7

[28] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000. 2

[29] 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)*, pages 2929–2936. IEEE, 2023. 3

[30] Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15342–15353, 2024. 3

[31] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 3

[32] Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. *arXiv preprint arXiv:2412.05334*, 2024. 7

[33] 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 Computer Vision and Pattern Recognition Conference*, pages 5422–5432, 2025. 6

[34] 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. 3, 6, 7, 8, 1

[35] 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, 3

[36] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17863–17873, 2023. 5

# LANGTRAJ: Diffusion Model and Dataset for Language-Conditioned Trajectory Simulation

Supplementary Material

## A. Experimental Details

### A.1. INTERDRIVE

The training split of INTERDRIVE includes 100k human-annotated language-trajectory pairs and 405k pairs annotated heuristically. The test split of INTERDRIVE contains 25k prompt-scenario pairs with both human and heuristic annotations. To address open-set language input, we augment INTERDRIVE's categorical annotations using GPT-4 [16] to generate approximately 20 rephrasings for each annotated behavior. This augmentation expands the range of language variations the model encounters, improving its robustness to diverse user inputs.

During training, we also use *compositional* prompts that combine agent-agent interaction descriptions with heuristic action labels (e.g., *speed up*, *turn left*, *wait*) into unified instructions. This compositionality supports the flexibility of behavior expression, and we apply it consistently across both training and evaluation. Note that our instructions do not include explicit temporal conditioning, which we leave for future work.

### A.2. ProSim-Instruct-520k

ProSim-Instruct-520k is a multimodal dataset designed for promptable traffic simulation, containing over 10 million text prompts paired with 520,000 driving scenarios. Each scenario includes goal points, route sketches, action tags describing agent behaviors, and text instructions generated by Llama3-70B. In contrast, INTERDRIVE is directly constructed from the interactive subset of the Waymo Open Dataset, ensuring a targeted selection of interactive scenarios. Additionally, our annotations are collected from human experts rather than LLMs, focusing specifically on high-quality interactive behavior labeling.

### A.3. LLM-Based Guidance Details

In this section, we provide implementation details for LLM-based guidance conditioning (CTG++ style [34]) method in Tab. 4, which generates differentiable loss functions conditioned on text descriptions. We use the o3-mini model via OpenAI APIs to generate loss functions that guide vehicle behaviors, following the implementation. The method leverages the same backbone and weights for all experiments to ensure consistency.

Since INTERDRIVE uses a fixed vocabulary, we generate a unique function for each interactive description and heuristic, and combine them by scaling the loss to a common range. LLM-based guidance may not work in the first iteration, as the generated functions often contain errors or inconsistencies. Common failure cases include issues with array shape mismatches, map-related functions, and the assumption of unseen functions. To address this, we manually correct the generated functions by providing more specific instructions to GPT. This process typically requires 3-5 cycles to refine the guidance functions and ensure there are no compilation errors.

### A.4. Testing Subsets

We evaluate all experiments on a 2% subset of the data, consisting of approximately 1,100 scenarios. Specifically, INTERDRIVE is tested on the validation interactive subset of the Waymo dataset, while ProSim-Instruct-520k is evaluated on the validation subset.

### A.5. WOSAC Challenge Metrics

The Waymo Open Sim Agent Challenge (WOSAC) evaluates simulation quality by computing negative log-likelihood (NLL) scores over nine predefined statistical features, covering kinematics, agent interactions, and map adherence, where each feature is evaluated independently. The challenge requires simulating up to 128 agents per scene for 8 seconds, generating 32 joint agents future samples per scenario. The negative log-likelihood is then computed based on an approximate empirical distribution constructed from the simulated trajectories.

For a given scenario $i$ and target agent $a$, the likelihood of the true trajectory under the empirical distribution of simulated samples is given by:

$$\text{NLL}(i, a, t, j) = -\log p_{i,j,a}(F_j(x^*(i, a, t))) \quad (3)$$

where $p_{i,j,a}(.)$ is the empirical histogram distribution of statistic $F_j$ obtained from the generated samples. A lower NLL indicates that the simulated distribution closely matches real-world behavior.

To obtain a per-scenario metric, the NLL values are aggregated over all valid timesteps:

$$m(a, i, j) = \exp\left(-\left[\tfrac{1}{N(i,a)} \textstyle\sum_t v(i, a, t)\text{NLL}(i, a, t, j)\right]\right) \quad (4)$$

where $N(i, a) = \sum_t v(i, a, t)$ represents the number of valid timesteps for target agent $a$. The final scenario-level metric is then computed by averaging over the target agents:

$$m(i, j) = \frac{1}{A_{\text{target}}} \sum_a m(a, i, j) \quad (5)$$

where $A_{\text{target}}$ represents the number of target agents.

To adapt WOSAC for language-conditioned interactive driving, we focus on evaluating target agents that have explicit interactive descriptions in natural language on INTERDRIVE, which is drawn from the validation interactive set, referred to as the objects of interest field in the Waymo Open Dataset format [3].

For ProsimInstruct evaluation, since the annotations are constructed from the validation set and may not contain objects of interest labels, we follow the original target agents from the validation set.

To compute the final composite metric for ranking submissions, WOSAC takes a weighted average over all component metrics:

$$M_K = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} w_j m_K(i,j), \quad \sum_{j=1}^{M} w_j = 1 \tag{6}$$

where $M = 9$ represents the nine statistical features, and $w_j$ are manually assigned weights for each metric. We detail the definitions of the component metrics below:

**Kinematic Metrics:**

- **Linear Speed:** Measures the magnitude of the first derivative of position, $\|v\| = \left\| \frac{x_{t+1}-x_t}{\Delta t} \right\|_2$, reflecting the agent's velocity in 3D space.
- **Linear Acceleration Magnitude:** Represents the magnitude of the second derivative of position, $\frac{\|v_{t+1}-v_t\|}{\Delta t}$, describing the agent's acceleration.
- **Angular Speed:** Calculates the rate of change of the agent's heading, $\omega = \frac{d(\theta_{t+1},\theta_t)}{\Delta t}$, where $d(\cdot)$ is the minimal angular difference on the unit circle.
- **Angular Acceleration Magnitude:** Measures the rate of change of angular speed, $\frac{d(\omega_{t+1},\omega_t)}{\Delta t}$.

**Interaction Metrics:**

- **Distance to Nearest Object:** The signed distance to the nearest object in the scene, calculated using the GJK distance algorithm.
- **Collisions:** Detected when the signed distance to the nearest object becomes negative, indicating that two objects have collided.
- **Time-to-Collision (TTC):** Estimates the time before a collision occurs, assuming constant velocities.

**Map Metric:**

- **Distance to Road Edge:** The signed distance to the nearest road edge in the scene.
- **Road Departure:** Indicates whether an agent has left the road at any point in time, based on the signed distance to the road edge.

For more details on each metric, refer to the original WOSAC Challenge paper [14].

| CFG Weight | Meta ↑ | Kinematic ↑ | Interactive ↑ | Map ↑ | mADE ↓ |
|---|---|---|---|---|---|
| -1.0 | 0.72 | 0.40 | 0.79 | 0.81 | 3.38 |
| 0.0 | 0.72 | 0.41 | 0.79 | 0.81 | 2.90 |
| 0.5 | 0.71 | 0.41 | 0.79 | 0.78 | 2.90 |
| 1.0 | 0.70 | 0.41 | 0.78 | 0.77 | 2.97 |
| 2.0 | 0.68 | 0.40 | 0.76 | 0.73 | 3.21 |

Table 7. **Analysis of Text Conditioning Strength.** We evaluate the impact of text conditioning in LANGTRAJ by varying the classifier-free guidance (CFG) weight. CFG=-1.0 represents the unconditional setting.

# B. Implementation Details

## B.1. Training Details

To maximize use of the human annotations in INTERDRIVE, we apply a biased sampling strategy to balance the training data. Specifically, we upsample human-annotated samples to represent 50% of each training batch and include 30% of heuristic descriptions in a given scene per sample. This approach allows for the simultaneous training of language-conditioned and unconditional diffusion models, optimizing both modes within the framework.

The training process for LANGTRAJ consists of two stages. In the first stage, the scene encoder and diffusion model are trained without text conditioning for 60,000 iterations using a batch size of 32. In the second stage, the scene encoder, language encoder, and diffusion model are trained with text conditioning for an additional 20,000 iterations using a batch size of 2048. The language encoder is implemented with a LoRA module [6], which updates only the linear projection layers of the query and key matrices in DistillBERT [20], configured with $R = 16$ and $\alpha = 0.4$. Both stages employ the Adam optimizer [11] with an initial learning rate of $1 \times 10^{-3}$. The diffusion model implementation follows methodologies from open-source repositories [8, 35].

To distill our pretrained scene-diffusion model from $K = 100$ to $K = 5$, we train the model using a new denoising schedule for 20,000 iterations with a batch size of 256. The original pretrained scene-diffusion model has a prediction horizon of $T = 16$ with a frequency of 0.5 Hz.

## B.2. Closed-loop Training Details

For closed-loop training of diffusion models, we first pretrain our scene-diffusion model with $K = 100$ denoising steps. We then distill the denoiser to $K = 5$ steps before applying closed-loop training, as described in Sec. 5.2.

We modify the model's prediction horizon from $T = 16$ to $T = 8$ under 2 Hz to accommodate multi-step unrolling with ground truth actions, using a replanning interval of $T_{\text{replan}} = 2$. Given 16 steps of ground truth future trajectories, we can perform four iterations of closed-loop training. During this process, the best sample among $M = 8$ can-

didates is selected for execution, and the adopted forward diffusion ratio $\gamma = 0.6$.

The teacher-forcing ratio is set to 50%. When applied, 70% of agents are randomly sampled to follow the ground truth states throughout the unrolling process. The model is trained with a learning rate of $1 \times 10^{-5}$ using an effective batch size of 32 for around 50,000 iterations, which takes around 12 hours on 8 8×A6000 GPUs and 32 CPU cores.

Additionally, we incorporate an auxiliary non-collision loss from Sec. D.3 with a relative weighting of 0.1. To enhance robustness, we randomly drop text conditioning and agent history with a probability of 50% during training.

### B.3. Inference Frequency

Per-sample inference with 5 de-noising steps takes 4.66±0.06 seconds per 8-second simulation (1 Hz replan) using 1×A6000 GPU and 4 CPU cores.Speedups via map caching, parallelism, and distillation can 069 be adopted for scalability.

### B.4. Denoising Process

At each denoising step, the model predicts the mean of the next denoised action trajectory. Instead of predicting the noise $\epsilon$ used to corrupt the trajectory, the model directly outputs the clean denoised trajectory $\hat{\tau}_0$. The predicted mean for reconstructing $\tau_{k-1}$ from $\tau_k$ is defined as:

$$\tau_{k-1} = \mu_\theta(\tau_k, \hat{\tau}_0) = \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\hat{\tau}_0 + \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\tau_k, \quad (7)$$

Where $\beta_k$ represents the variance from the noise schedule in the diffusion process, $\alpha_k = 1 - \beta_k$ denotes the incremental noise reduction at each step, and $\bar{\alpha}_k = \prod_{j=0}^{k} \alpha_j$ is the cumulative product of $\alpha_j$ up to step $k$.

### B.5. Diffusion Process and Inference Details

For the diffusion process, we utilize a cosine variance schedule, with the number of diffusion steps set to $K = 100$ for pretrained diffusion model and $K = 5$ for the closed-loop trained diffusion models. The cosine variance scheduler followed [15], with $s = 0.008$. The model operates on a 1.1-second trajectory history and is trained to predict the next 8.0 seconds with a time step $\Delta t = 0.5$. During inference, we use a DDIM sampler [23] with a stride step 1 during inference. During inference, we sample $M = 64$ joint future samples for all agent, and only select the one joint agent sample with lowest collision loss. The **per-sample inference** time is 4.66±0.06 seconds for an 8-second simulation, with a 1 Hz replan rate. This process utilizes a 1×A6000 GPU and 4 CPU cores. To improve scalability, speedups can be achieved through techniques such as map caching, parallelism, and distillation.

## C. Discussion and Limitation

While the model generally follows instructions well, we observe that failure cases often involve conflicts between language input and recent history (e.g., past 1s of motion). In such cases, the model tends to prioritize history, leading to behavior misaligned with the instruction. This is particularly noticeable when the vehicle is static, making conditioning signals harder to take effect. Additionally, minADE may not fully capture instruction adherence; future work could explore human evaluations to better assess language-action alignment.

## D. Guidance Details

### D.1. Classifier-Free Guidance

Contrary to previous diffusion models that rely on classifier guidance, direct conditioning enables control through the learned data distribution rather than predefined objectives. Classifier-free guidance leverages this distribution for generation without requiring domain-specific priors.

As shown in Tab. 7, we observe that moderate classifier-free guidance (weights 0.0–1.0) maintains realism and controllability, while higher weights (1.0) degrade map adherence (0.81 → 0.73) and interactive realism (0.79 → 0.76), suggesting that excessive text conditioning misaligns trajectories with the map structure. Qualitatively, while stronger guidance improves instruction-following, it also tends to produce more off-road samples.

### D.2. Collision Guidance

We define the collision cost as

$$J_{\text{coll}} = -\sum_{t=1}^{T} d(t),$$

where d(t) is the distance between the adversarial and target agents at time step t over the planning horizon T. Minimizing $J_{\text{coll}}$ encourages collisions. In Tab. 5, one of the interactive agent from the INTERDRIVE is designated as adversarial and the other as the target. Note that, we only compute the gradient with respect to the adversarial agent.

### D.3. Non-Collision Guidance

To detect collisions in a differentiable way, we approximate each agent $i$ with $D$ equally spaced disks of radius $r_i$ [24]. For any pair of agents $(i, j)$, let $d$ be the minimal distance between their respective disk centers at time $t$. If $d$ is less than the sum of their radii, the circles overlap. Formally, the pairwise collision loss for agents $i$ and $j$ at time $\tau$ is:

$$J_{\text{pair}}(\tau_i, \tau_j) = \begin{cases} 1 - \dfrac{d}{r_i + r_j}, & \text{if } d \le r_i + r_j, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

We sum over all agent pairs and all timesteps $t = 0, \ldots, T$ to obtain the total collision loss:

$$J_{\text{no collision}} = \frac{1}{N^2} \sum_{i \neq j} \max\Big(1, \sum_{\tau=0}^{T} J_{\text{pair}}\big(\tau_i, \tau_j\big)\Big), \quad (9)$$

where $N$ is the total number of agents, $T$ is the planning horizon, and $\tau_i$ represents the state of agent $i$ at time $\tau$. If no disks overlap, $J_{\text{pair}}$ is zero; fully overlapping disks produce a maximum penalty of 1.

## E. INTERDRIVE Interaction Type Definitions

This section defines the detailed behaviors considered in our study and the corresponding labeling process used to categorize them.

INTERDRIVE uses a scalable human labeling process based on multiple-choice questions organized into large categories and subcategories to define and categorize behaviors efficiently. This structured approach reduces ambiguity, provides clear guidance for annotators, and ensures consistent, high-quality labels. By leveraging this method, INTERDRIVE achieves 2.5 times the size of the WOMD-Reasoning dataset, with higher-quality annotations and slightly lower overall labeling costs.

In addition to categorizing interaction types, annotators also identify whether a pair of agents is interacting. This process accounts for the possibility of *asymmetric interactions*, where one agent interacts with another, but the reverse may not be true. For example, Agent 2 may adjust its behavior in response to Agent 1 (e.g., yielding or avoiding), while Agent 1 may proceed unaffected, exhibiting no interaction.

- **Lane Change:** Lane change interactions involve moving from one lane to another for various purposes:
  - **Changing lane for turn or exit:** Moving to another lane in preparation for turning or exiting.
  - **Changing lane for overtaking:** Moving to another lane to pass a slower vehicle.
  - **Lane-change for avoiding obstacles or slower traffic:** Changing lanes to bypass road obstacles or slower-moving vehicles.
  - **Lane-change for merging:** Changing lanes to merge into another stream of traffic.
  - **Changing lane with lead or trail:** Performing a lane change with another vehicle directly ahead (lead) or behind (trail), requiring extra caution.
- **Following/Stopping Behind:** These interactions involve adjusting speed and distance while following or stopping behind another vehicle:
  - **Following with a lead vehicle:** Adjusting speed to maintain a safe distance while following another vehicle.
  - **Following a slow-moving lead:** Driving slower than desired to follow a slower vehicle ahead.
  - **Tailgating:** Driving too closely behind another vehicle, often considered aggressive driving.
  - **Stopping behind a lead vehicle:** Coming to a stop behind another vehicle, typically at traffic lights or stop signs.
  - **Stopping behind an intersection:** Coming to a stop before entering an intersection.
- **Yielding:** Yielding involves giving the right of way to other road users in specific situations:
  - **Intersection yielding:** Yielding to oncoming traffic or other road users at intersections.
  - **Yielding before merging or lane-change:** Yielding to ongoing traffic when changing a lane or merging.
  - **Yielding to merging or lane-change cars:** Yielding to cars that change lanes or merge into the current lane.
  - **Waiting for a pedestrian to cross:** Yielding to a pedestrian at a crosswalk or intersection, allowing them to cross safely.
  - **Roundabout yielding:** Yielding to vehicles already in a roundabout before entering.
  - **Pedestrian yielding to vehicles:** Pedestrians pause and give way to oncoming vehicles before crossing the road.
- **Passing:** Passing involves moving past vehicles, pedestrians, or obstacles without yielding:
  - **Passing through an intersection with yielding vehicles:** Moving through an intersection without yielding, while other cars yield.
  - **Passing a pedestrian:** Moving past a pedestrian walking near the road or on a crosswalk, ensuring a safe distance.
  - **Pedestrian passing a vehicle:** A pedestrian moves around or crosses in front of a stationary vehicle.
  - **Passing through a roundabout:** Navigating through a roundabout without stopping, maintaining the right of way.
  - **Maintaining speed while driving:** Driving straight or turning while maintaining the original speed without yielding or merging.
  - **Passing as a leading vehicle:** Passing with other vehicles following.
  - **Pedestrian or cyclist crossing the road:** Pedestrians or cyclists crossing the road, usually with vehicles yielding.
- **Overtaking:** Overtaking involves actively moving ahead of another vehicle:
  - **Car avoidance:** Taking evasive action to avoid another vehicle, often involving swerving, braking, or accelerating.
  - **Standard overtaking:** Passing a slower vehicle by moving to an adjacent lane and returning to the original lane.
  - **High-speed overtaking:** Passing at higher speeds on

highways, requiring careful attention to speed and distance.

- **Merging:** Merging involves entering a lane of traffic from a merging lane, on-ramp, or after a lane reduction:
  - **Standard merge:** Entering the flow of traffic from a merging lane or on-ramp.
  - **Lane reduction merge:** Merging into an adjacent lane when a lane ends due to road conditions.
  - **Zipper merge:** A coordinated merge where vehicles in two lanes alternate into a single lane.
  - **Highway on-ramp accelerating merge:** Entering a highway while accelerating to match traffic speed.
  - **Late merge:** Merging closer to the end of the merging lane in congested traffic.
- **Other:**
  - **Undefined behavior:** For scenarios where a specific behavior type does not exist in the predefined options. In this case, the human labeler will type in behavior descriptions.
  - **No interaction:** For cases where no interaction occurs.
  - **Unknown status:** For scenarios where the behavior cannot be determined due to insufficient or unclear data.