

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 com-

mon 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 $p_{i,j,a}(\cdot)$ 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:

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

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_{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 INTER-DRIVE, 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:

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

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 \uparrow	Kinematic \uparrow	Interactive \uparrow	Map \uparrow	mADE \downarrow
-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 INTER-DRIVE, 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 1×10^{-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 pre-train 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 1×10^{-5} using an effective batch size of 32 for around 50,000 iterations, which takes around 12 hours on 8 8xA6000 GPUs and 32 CPU cores.

Additionally, we incorporate an auxiliary non-collision loss from Appendix 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 1xA6000 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 ϵ used to corrupt the trajectory, the model directly outputs the clean denoised trajectory $\hat{\tau}_0$. The predicted mean for reconstructing τ_{k-1} from τ_k is defined as:

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

Where β_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 α_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 1xA6000 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 ($0.81 \rightarrow 0.73$) and interactive realism ($0.79 \rightarrow 0.76$), 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_{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 τ is:

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

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

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

where N is the total number of agents, T is the planning horizon, and τ_i represents the state of agent i at time τ . If no disks overlap, J_{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 WOMB-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.