Generative Scenario Rollouts for End-to-End Autonomous Driving

Rajeev Yasarla[1✉] **Deepti Hegde**[1] **Shizhong Han**[1] **Hsin-Pai Cheng**[1] **Yunxiao Shi**[1]
Meysam Sadeghigooghari[2] **Shweta Mahajan**[1*] **Apratim Bhattacharyya**[1]
Litian Liu[1] **Risheek Garrepalli**[1] **Thomas Svantesson**[2] **Fatih Porikli**[1]
Mohammad Ghavamzadeh[1] **Hong Cai**[1✉]

[1]Qualcomm AI Research[†] [2]Qualcomm Technologies, Inc.

{ryasarla, hongcai}@qti.qualcomm.com

Abstract

Vision-Language-Action (VLA) models are emerging as highly effective planning models for end-to-end autonomous driving systems. However, current works mostly rely on imitation learning from sparse trajectory annotations and underutilize their potential as generative models. We propose Generative Scenario Rollouts (GeRo), a plug-and-play framework for VLA models that jointly performs planning and generation of language-grounded future traffic scenes through an autoregressive rollout strategy. First, a VLA model is trained to encode ego vehicle and agent dynamics into latent tokens under supervision from planning, motion, and language tasks, facilitating text-aligned generation. Next, GeRo performs language-conditioned autoregressive generation. Given multi-view images, a scenario description, and ego-action questions, it generates future latent tokens and textual responses to guide long-horizon rollouts. A rollout-consistency loss stabilizes predictions using ground truth or pseudo-labels, mitigating drift and preserving text-action alignment. This design enables GeRo to perform temporally consistent, language-grounded rollouts that support long-horizon reasoning and multi-agent planning. On Bench2Drive, GeRo improves driving score and success rate by +15.7 and +26.2, respectively. By integrating reinforcement learning with generative rollouts, GeRo achieves state-of-the-art closed-loop and open-loop performance, demonstrating strong zero-shot robustness. These results highlight the promise of generative, language-conditioned reasoning as a foundation for safer and more interpretable end-to-end autonomous driving.

1 Introduction

Autonomous driving systems require robust navigation and planning capabilities in complex and dynamic environments. Recent approaches train end-to-end (E2E) motion planners that map raw sensor inputs to future ego-vehicle trajectories [25, 51, 57, 42, 18, 54]. Within this paradigm, Vision-Language-Action (VLA) models use large language models (LLMs) to integrate linguistic context into motion planning, enabling language-informed driving decisions [49, 38, 59, 46, 10, 4, 11].

Despite this progress, current VLA models exhibit critical limitations and leave several directions underexplored. (1) Sparse language-action supervision: Driving datasets [40, 37, 11] often provide scene-level descriptions and Q&A pairs but rarely include fine-grained actions tied to temporal phases of driving events, making models brittle in ambiguous or long-tail scenarios, e.g., distinguishing overtaking from merging, localizing the ego vehicle's intent within multi-step maneuvers [11]. (2) Underutilized generative capability: Existing VLA methods rely only on ground-truth trajectories

*Work done while employed at Qualcomm AI Research.

†Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

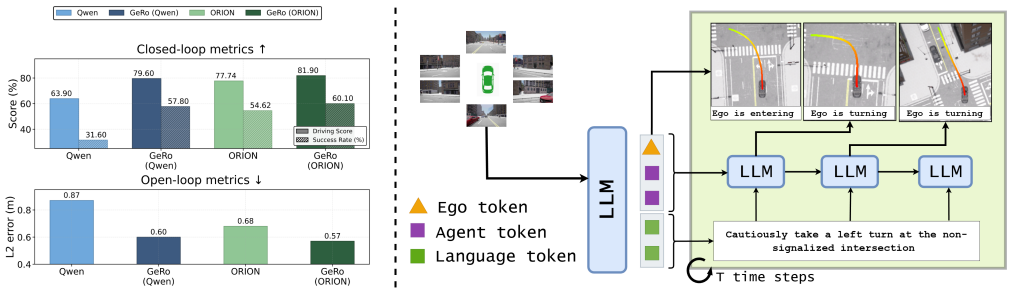


Figure 1: (*Left*) Closed-loop metrics (Driving Score and Success Rate) and open-loop metric (Trajectory L2 error) for different models. GeRo consistently improves both closed-loop and open-loop performance across baselines like Qwen2.5 [2] and ORION [11]. (*Right*) GeRo scenario rollout pipeline: given scenario description and multi-view observations, GeRo generates temporally consistent agent behaviors using a large language model (LLM) across time steps.

for planning, overlooking the potential of autoregressive generation for scenario-level reasoning and exploration. (3) Descriptive vs. procedural language: Current language supervision typically describes what is happening rather than how actions unfold, limiting the model's ability to capture procedural nuances necessary for planning. (4) Language-action misalignment: Many datasets generate instruction–action pairs after collecting expert driving data [38], so models often infer from visual cues only and ignore language, causing failures like "stop for red light" paired with acceleration. To address these issues, we propose Generative Scenario Rollouts (GeRo), a plug-and-play training framework that unifies VLA models with scenario generation for language-aligned action prediction and improved planning robustness (Figure 1). To the best of our knowledge, this is the first work to jointly perform scenario generation with motion prediction, planning, and visual question-answering. GeRo has two phases:

- **Pretraining:** In this stage, the VLA model is trained to encode ego and agent dynamics into a compact shared token space. The model is jointly supervised for planning, multi-agent motion prediction with auxiliary bounding-box cues, and visual question answering. This stage is crucial for binding linguistic and behavioral representations, enabling text-aligned generation beyond paired annotations, thus reducing language-action misalignment.
- **Language-Conditioned Scenario Rollout:** In this phase, the model autoregressively predicts future latent tokens and ego-aligned action descriptions based on past latent tokens. The tokens are fed back to the VLA model to guide long-horizon reasoning. Predictions are stabilized using a consistency loss that mitigates drift and enforces text-action consistency. Specifically, we enforce temporal consistency by aligning the latent distribution of rollout predictions with the pretrained latent distribution for future time steps using KL-Divergence. When ground-truth data is available, imitation learning is applied by supervising the model with corresponding waypoints and motion trajectories to enforce scenario generation. However, relying solely on imitation learning can lead to suboptimal performance; therefore, we introduce reinforcement learning based feedback with GRPO [39] for the scenario rollouts. This incorporates rewards relating to collision avoidance, time-to-collision (TTC), and language-aligned scenario generation using question–answer pairs relating to the action of the ego-vehicle.

Our contributions are threefold:

1. We introduce **GeRo**, a generative scenario rollout framework that integrates autoregressive scenario generation into VLA models. This design couples temporal reasoning with linguistic context, enabling scenario rollouts that predict future motions of traffic agents and ego actions for robust planning.
2. We propose a novel set of reward functions for GRPO-based reinforcement learning that jointly optimize trajectory accuracy and semantic alignment with language descriptions during scenario rollouts. These rewards incorporate safety-critical metrics such as collision avoidance and time-to-collision, ensuring high-fidelity and interpretable planning behavior.
3. **GeRo** introduces an interactive Visual Question Answering (VQA) component that grounds ego-vehicle intent in natural language, allowing the model to answer scenario-specific questions during rollouts. This enhances interpretability and provides a mechanism for language-guided reasoning in complex driving environments.

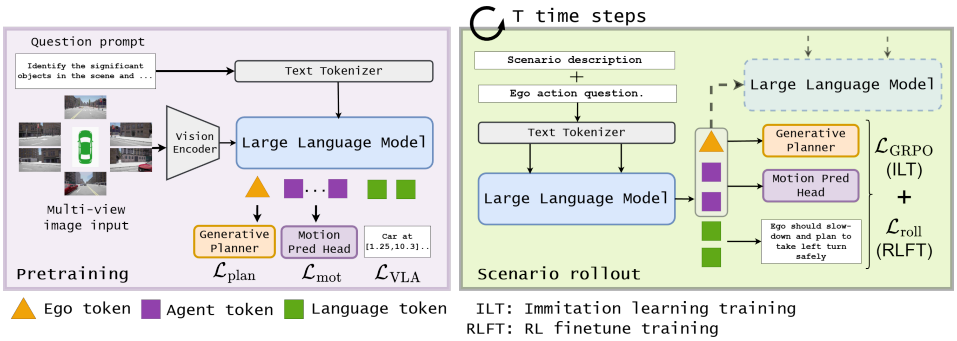


Figure 2: **Overview of GeRo's two stage framework.** (*Left*) GeRo pretraining stage of the Vision–Language–Action (VLA) model. Multi-view images and scenario prompts are encoded into visual and text tokens and projected by the LLM head to compute ego and agent tokens. These tokens are passed to the *generative planner* and the *motion prediction head*. The planning, and motion and language prediction tasks are supervised by their respective loss functions ($\mathcal{L}_{\text{plan}}$, $\mathcal{L}_{\text{mot}}$, $\mathcal{L}_{\text{VLA}}$). (*Right*) GeRo's autoregressive rollout stage. Given a scenario description and ego-action questions, GeRo performs scenario rollout to predict future tokens for T steps, decoding them into ego waypoints, multi-agent trajectories, and language outputs. Rollouts are optimized with consistency loss ($\mathcal{L}_{\text{roll}}$) and reinforcement feedback ($\mathcal{L}_{GRPO}$) for robust scenario grounding.

Through extensive closed-loop and open-loop experiments, we show that GeRo significantly outperforms VLA baselines [11] and other E2E motion planners [25, 18, 24], particularly in corner cases and zero-shot planning, highlighting the importance of scenario-level reasoning and language-action alignment for E2E autonomous driving.

2 Related Works

2.1 VLA for Autonomous Driving

While conventional E2E models like VAD [25] and UniAD [18] offer fast inference and streamlined integration of perception, prediction, and planning, they often struggle with generalization in long-tail scenarios. Recent LLM-based approaches [47, 7, 11, 49, 38, 59, 46, 10, 4, 54, 38, 16] integrate world-model reasoning with language, improving robustness in long-tail scenarios. GPT-Driver [33] reformulates planning as text generation, converting observations into language prompts and leveraging chain-of-thought reasoning [50]. EMMA [20] enables models like Gemini [4] to predict discrete textual plans with strong open-loop performance. Hybrid dual-system architectures combine fast E2E models with slower, reasoning-capable VLMs. DriveVLM [46] employs a dual system where a VLM predicts low-frequency trajectories refined by a conventional planner. Senna [10] introduces meta-actions to guide VAD for semantically aligned motion trajectories. ORION [11] proposes a unified E2E framework that aligns VLM reasoning with trajectory prediction for closed-loop scenarios. SimLingo [38] further improves language-scene alignment, achieving competitive accuracy on Bench2Drive, leveraging additional, higher-quality training data.

2.2 Driving Scenario Generation

Generative models, particularly diffusion-based ones, have shown promise in modeling the stochastic nature of driving behavior and generating diverse, multimodal trajectories. DiffusionDrive [29] introduces a truncated diffusion process with anchor priors, enabling real-time multimodal planning. GoalFlow [53] applies goal-point guidance with flow matching for high-quality trajectory generation, while DiffusionPlanner [58] redefines planning as future trajectory generation, jointly producing plans and motion forecasts. InfGen proposes to do arbitrary resolution image generation in the latent space with diffusion models [15]. In the VLA context, ORION extends its framework with a VAE-based generative planner that translates LLM outputs into trajectory distributions [11]. DiVLA [32] combines autoregressive reasoning with diffusion policy learning, introducing a reasoning injection module that enhances interpretability and generalization. These generative VLA models establish a foundation for explainable, multimodal driving policies, motivating our generative rollout approach.

2.3 Reinforcement Learning for Driving

Reinforcement learning (RL) has emerged as a powerful paradigm for improving policy generalization and robustness in autonomous driving. Unlike imitation learning, which suffers from covariate shift [35], RL enables agents to learn from interaction and adapt to novel scenarios. RAD [13]

trains E2E agents in photorealistic 3DGS environments, demonstrating improved generalization. CarPlanner [55] introduces an autoregressive policy that generates consistent multimodal trajectories, outperforming imitation-based baselines through expert-guided reward shaping. Recent efforts also explore synergy between RL and VLMs: GenDrive [19] combines diffusion models with RL and reward modeling, while [26, 31, 27] incorporate GRPO-based training to enhance policy stability and generalization. By integrating VLMs into the RL loop, these approaches enable semantic grounding and improve decision-making in complex, long-tail scenarios.

3 Methodology

We present GeRo, a generative scenario rollout framework for VLA models that unifies planning and autoregressive scenario generation. GeRo is shown in Figure 2. In GeRo, rollout operates in a token space that is learned in a pretraining stage, where ego and agent states are represented as compact latent tokens. Given latent ego-tokens and agent tokens along with a high-level scenario description and a question regarding the actions of the ego-vehicle, the GeRo framework uses a VLA model to autoregressively generate future scenarios by predicting their latent ego and agent tokens. The model is also jointly trained to perform planning, motion prediction, and visual question-answering. These tasks are supervised under GRPO [39] with a series of specially designed reward functions. During rollout training, GRPO [39] complements the supervised losses with safety and semantic rewards. An overview of the rollout mechanism can be seen in Figure 3.

3.1 Problem Formulation

Consider a driving scene at time t consisting of an ego vehicle and a set of N_t traffic participants $\mathcal{A}_t$. Given a sequence of multi-view images, a scene description, and information about the state of the ego-vehicle x_t^e and the state of the other traffic participants $x_t^{(a_i)}$, the objective of GeRo is to jointly predict (i) an ego trajectory $\hat{\tau}_t^e$ that supports safe and efficient driving, (ii) future motions of surrounding agents $\{\hat{\tau}_t^{(a_i)}\}_{i=1}^{N_t}$,[3] and (iii) a scenario-based ego-action answer to question q. The ego state comprises of pose, velocity, and acceleration. The state of agent a_i consists of the agent's class, bounding box, motion trajectory, velocity, and acceleration.

3.2 Architecture

GeRo provides a general plug-and-play framework for Vision-Language-Action models. Thus, the architecture follows several recent end-to-end planners that leverage multi-modal LLMs for motion planning [11, 49, 44]. The vision encoder and the text tokenizer project image and text inputs to a common token embedding space. The central component is the LLM, which projects multi-modal tokens into a series of compact latent representations of the ego-vehicle and the surrounding agents. The LLM also predicts a scene description and answers questions related to the ego-vehicle behavior. Lastly, the framework contains a variational autoencoder (VAE) planning head and a motion prediction head.

3.3 Pretraining

The objective of the pretraining stage is to achieve a good initialization of action tokens for stable scenario generation in the token space. To this end, we first pretrain the VLA model to learn a compact tokenization of ego and agent dynamics. In this stage, we supervise three heads: (i) a planning head, which predicts future ego trajectory, (ii) the motion head, which predicts future multi-agent trajectories, and (iii) the large language model, which is trained to predict scene descriptions and perform visual question answering. The model is also trained to predict the detection bounding box of the agents along with their corresponding trajectories, $\{\hat{\tau}_t^{(a_i)}, \hat{\phi}_t^{(a_i)}\}_{i=1}^{N_t}$ to accelerate convergence of trajectory prediction in the motion head. This pretraining stage teaches the model to encode and decode latent tokens for both ego and agents that remain consistent and representative of the scenario.

Pretraining loss functions. The planning head is supervised by using two losses: an L1 loss for waypoint regression and a collision loss. The motion prediction head is supervised using (i) focal loss [30] for classification of agents' types, (ii) L1 loss for trajectory prediction, and (iii) L1 loss for 3D bounding box detection. The LLM is trained with a cross-entropy loss for language prediction, following [2]. Let $\mathcal{L}_{\text{plan}}$, $\mathcal{L}_{\text{mot}}$, and $\mathcal{L}_{\text{VLA}}$ denote these respective terms; the overall pretraining loss is,

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{plan}} + \mathcal{L}_{\text{mot}} + \mathcal{L}_{\text{VLA}}. \tag{1}$$

[3] For notational simplicity, we drop the agent indices when appropriate. For example, we refer to $\{\hat{\tau}_t^{(a_i)}\}_{i=1}^{N_t}$ as $\{\hat{\tau}_t^{(a_i)}\}$, and we refer to the set of ego and agent trajectories, $\{\hat{\tau}^e, \{\hat{\tau}_t^{(a_i)}\}_{i=1}^{N_t}\}$ as $\{\hat{\tau}_t\}$, when it is clear from the context.

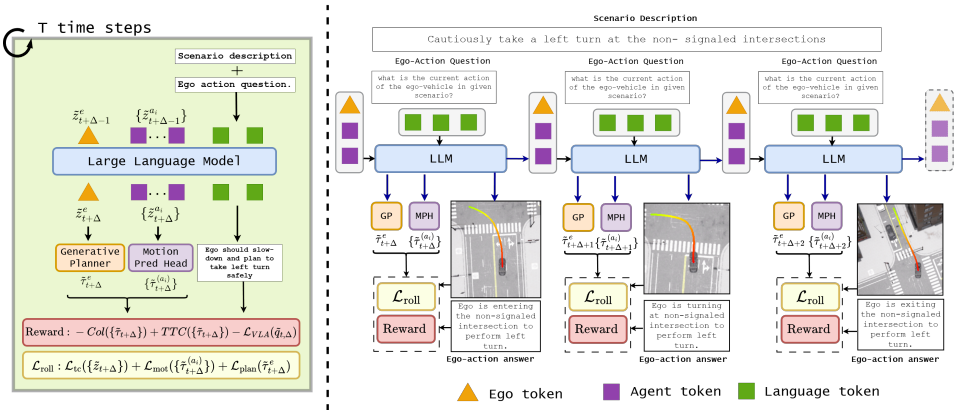


Figure 3: **Autoregressive scenario rollout in GeRo.** At time t, GeRo consumes ego and agent tokens along with scenario description and ego-action questions, and predicts next-step tokens using the LLM head. These tokens are decoded into ego trajectories, multi-agent trajectories, and language answers. The process repeats for T steps, with updated scenario descriptions and ego-action questions at each rollout step, enabling consistent rollouts aligned with text and environment dynamics. Rollouts are optimized using the consistency loss $\mathcal{L}_{\text{roll}}$ and the reinforcement feedback loss $\mathcal{L}_{\text{GRPO}}$ for robust planning and reasoning.

3.4 Generative Scenario Rollout

Generative scenario rollout training comprises two sequential phases. First, imitation learning with $\mathcal{L}_{\text{roll}}$ learns temporally consistent latent futures and accurate ego and agent trajectories from logged driving data. We then apply GRPO-based reinforcement learning with $\mathcal{L}_{\text{GRPO}}$ to refine the model using safety feedback from generated rollouts.

Scenario Rollout. Given a sequence of multi-view images up to time t, GeRo first computes the latent ego token z_t^e, the latent agent tokens $\{z_t^{(a_i)}\}$, and scenario description s_t, as shown in Figure 2. Next, GeRo performs autoregressive rollouts in the token space for T total steps as shown in Figure 3. Conditioned on latent tokens $\{z_t\}$, multi-view images at time t, a scenario description s, and the ego-action question $q_{t,1} \in Q$, the generator predicts the next-step tokens $\{\tilde{z}_{t+1}\}$ and the responses to the ego-action questions. These tokens are decoded into the ego trajectory, agent motion trajectories, and language outputs. The predicted tokens are fed back to the model along with a scenario description s and the next ego-action question $q_{t,\Delta+1}$, where $\Delta \in [0, T-1]$. This enables the model to generate the subsequent tokens $\{\tilde{z}_{t+\Delta+1}\}$, resulting in a rollout from t to $t+T$.

Language Supervision. A central goal of GeRo is to generate scenario rollouts of action tokens while grounding them in language, enabling the model not only to produce semantically coherent sequences of future tokens but also to reason about the context. Rollout is guided by a scene description s provided by the LLM and a question q related to the action of the ego vehicle. Given each image sequence up to time t, its high-level scenario category, critical object information, and language annotations, the LLM is trained to then generate a rich scenario description s and corresponding ego-action questions Q that serve as language guidance for the autoregressive rollout.

Enforcing Rollout Consistency. During scenario rollout, temporal consistency is enforced through two complementary mechanisms. First, the latent distribution of rollout predictions are aligned with the pretrained latent distribution for future time steps using KL-Divergence. Second, the model is supervised using imitation learning when ground truth labels are available.

When ground truth trajectory labels are unavailable, they can be constructed from latent tokens $z_{t+\Delta}$ predicted by the pretrained VLA model and conditioned on the input image sequence with additional future frames. This setup promotes temporal consistency across extended rollouts and helps mitigate compounding errors commonly observed in autoregressive generation.

To stabilize training over long horizons, we apply step-wise temporal consistency supervision $\mathcal{L}_{tc}$ using pretraining heads under two regimes:

- **Ground-Truth Supervision:** When ground-truth data is available, we supervise the model using true states $\{x_{t+\Delta}\}$ and trajectories $\{\tau_{t+\Delta}\}$, providing accurate targets for each rollout step.

- **Model-Based Supervision:** In the absence of ground-truth, we rely on tokens $\{z_{t+\Delta}\}$ predicted by the pre-trained VLA model (refer to Section 3.3) using corresponding input image sequence as reference targets. These predictions serve as pseudo-labels to guide GeRo during unsupervised rollout steps. The token-consistency loss is defined as: $\mathcal{L}_{\text{tc}}\big(\{\tilde{z}_{t+\Delta}\}\big) = \text{KL}\big(\{\tilde{z}_{t+\Delta}\} \,\|\, \{z_{t+\Delta}\}\big)$, where $\text{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence. This formulation enforces temporal consistency in scenario generation by aligning the rollout predictions and the latent distribution of the pre-trained model.

This temporal consistency loss $\mathcal{L}_{\text{tc}}$ encourages alignment between the predicted and reference sequences—whether derived from ground-truth or VLA-generated pseudo-labels. As a result, this rollout loss reduces drift and improves the fidelity of long-horizon planning. The rollout loss is formulated by integrating both planning and motion prediction components as follows:

$$\mathcal{L}_{\text{roll}} = \sum_{\Delta=1}^{T} \Big[\mathcal{L}_{\text{tc}}(\{\tilde{z}_{t+\Delta}\}) + \mathcal{L}_{\text{plan}}(\{\tilde{\tau}_{t+\Delta}\}) + \mathcal{L}_{\text{mot}}(\{\tilde{\tau}_{t+\Delta}\})\Big] \tag{2}$$

RL with GRPO. After the imitation learning phase, we train our model using Group Relative Policy Optimization (GRPO) [39] with safety and semantic feedback. Given the multi-modal scene features x_t, we generate a group $\{\tau_g\}_{g=1}^{G}$ of $G = 8$ rollouts. Each rollout τ_g contains ego waypoints $\{\tilde{\tau}^{e}_{g,t+\Delta}\}_{\Delta=1}^{T}$ and surrounding-agent trajectories $\{\tilde{\tau}^{(a_i)}_{t+\Delta}\}$, where the ego waypoints are sampled from the generative planner and agent motion trajectories predicted through the motion prediction head. Each rollout receives a scalar reward aggregating collision (Coll), time-to-collision (TTC), and language-consistency terms over the rollout steps, i.e.,

$$R_g = \sum_{\Delta=1}^{T} \Big[-\text{Coll}(\{\tilde{\tau}_{g,t+\Delta}\}) + \text{TTC}(\{\tilde{\tau}_{g,t+\Delta}\}) - \mathcal{L}_{\text{VLA}}(\tilde{y}_{g,t+\Delta})\Big], \tag{3}$$

where $\tilde{y}_{g,t+\Delta}$ denotes the generated language response, which is scored against its reference using $\mathcal{L}_{\text{VLA}}$. We compute the group baseline and normalized advantage as

$$b = \frac{1}{G}\sum_{g=1}^{G} R_g, \qquad \sigma_R = \text{std}\{R_g\}_{g=1}^{G}, \qquad A_g = \frac{R_g - b}{\sigma_R + \zeta},$$

where ζ is a small positive number, and optimize the policy $\pi_\theta(\cdot|x)$ using the GRPO objective

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{G}\sum_{g=1}^{G} \min\Big(r_g(\theta)A_g,\ \text{clip}\big(r_g(\theta), 1-\epsilon, 1+\epsilon\big)A_g\Big) \tag{4}$$

where $r_g(\theta) = \frac{\pi_\theta\left(\{\tilde{\tau}_{g,t+\Delta}\}|x\right)}{\pi_{\theta_{\text{old}}}\left(\{\tilde{\tau}_{g,t+\Delta}\}|x\right)}$ and the clip parameter is set to $\epsilon = 0.1$. We use GRPO with batch size 256 and mini-batches of size 32.

3.5 Overall Supervision

The loss functions used in the two stages of GeRo are as follows:

- **Stage-1: Pretraining.** The VLA model is trained to learn a representative action token space under the pretraining objective $\mathcal{L}_{\text{pre}}$, summarized in Eq. 1.
- **Stage-2: Scenario Rollout Generation.** The VLA is optimized in two sequential phases: imitation learning using $\mathcal{L} = \mathcal{L}_{\text{roll}}$ (Eq. 2), followed by GRPO-based reinforcement learning using $\mathcal{L} = \mathcal{L}_{\text{GRPO}}$ (Eq. 4).

Here, $\mathcal{L}_{\text{roll}}$ enforces consistency across autoregressive rollout, while $\mathcal{L}_{\text{GRPO}}$ incorporates GRPO-based reinforcement feedback for robust scenario grounding.

4 Experimental setup

4.1 Dataset and Evaluation Metrics

Dataset. GeRo is trained and evaluated on the Bench2Drive [23] dataset, a closed-loop benchmark based on CARLA [6] for end-to-end autonomous driving. We use the official set of 1000 clips (950 for training, 50 for open-loop validation), each covering about 150 meters of driving in diverse traffic scenes. Closed-loop evaluation follows the official protocol on 220 routes across 44 interactive scenarios. For open-loop planning, we use the nuScenes [3] planning benchmark, which contains

Table 1: Comparison of closed-loop and open-loop performance of end-to-end autonomous driving methods on the Bench2Drive base set. Avg. L2 denotes the average trajectory error over 2 seconds at 2 Hz, following UniAD [18]. Abbreviations: NC = Navigation Command, TP = Target Point, DS = Driving Score, SR = Success Rate, M = Map, B = Bounding boxes, D = Depth, Mo = Motion prediction, O = Occupancy, S = Segmentation, E = Expert distillation. ‡ indicates models trained with geometric path waypoint supervision and evaluated using the PID controller from SimLingo.

Method	Condition	Aux. Sup.	Closed-loop Metric				Open-loop Metric
			DS↑	SR(%)↑	Efficiency↑	Comfortness↑	Avg. L2 ↓
TCP-traj [52]	TP	E+Speed	59.90	30.00	76.54	18.08	1.70
ThinkTwice [22]	TP	E+D+S+M	62.44	31.23	69.33	16.22	0.95
DriveAdapter [21]	TP	E+S	64.22	33.08	70.22	16.01	1.01
SimLingo [38]‡	TP	-	85.07	67.27	**259.23**	33.67	-
iPad [14]	TP	-	65.02	35.91	161.31	28.21	0.97
RAP-ResNet [9]	TP	3D Raster	66.42	37.27	165.47	23.63	-
UniAD-Base [18]	NC	M+B+Mo+O	45.81	16.36	129.21	43.58	0.73
VAD [25]	NC	M+B+Mo	42.35	15.00	157.94	46.01	0.91
GenAD [57]	NC	M+B+Mo	44.81	15.90	-	-	-
MomAD [41]	NC	M+B+Mo	44.54	16.71	170.21	**48.63**	0.87
DriveTransformer-Large [24]	NC	M+B+Mo	63.46	35.01	100.64	20.78	0.62
ReCogDrive [28]	NC	-	71.36	45.45	138.18	17.45	-
HiP-AD [43]‡	NC	M+B+Mo	86.77	69.09	203.12	19.36	0.69
Qwen2.5 [2]	NC	B+Mo	63.90	31.60	119.34	10.11	0.87
ORION [11]	NC	M+B+Mo	77.74	54.62	151.48	17.38	0.68
ORION (Qwen2.5) [11]	NC	M+B+Mo	76.88	53.90	146.16	24.75	0.72
GeRo (Qwen2.5) (ours)	NC	B+Mo	79.60	57.80	159.66	44.75	0.60
GeRo (ORION) (ours)	NC	M+B+Mo	81.90	60.10	176.51	40.22	0.57
GeRo (ORION) (ours)‡	TP	M+B+Mo	**87.16**	**68.76**	182.94	38.27	**0.56**

Table 2: Multi-ability performance Comparison for E2E-AD methods on Bench2Drive base set. NC = Navigation Command, TP = Target Point, M = Map, B = Bounding boxes, D = Depth, Mo = Motion prediction, O = Occupancy, S = Segmentation, E = Expert distillation. ‡ indicates models trained with geometric path waypoint supervision and evaluated using the PID controller from SimLingo.

Method	Condition	Aux. Sup.	Ability (%) ↑					
			Merging	Overtaking	Emergency Brake	Give Way	Traffic Sign	Mean
TCP-traj [52]	TP	E+Speed	28.84	24.29	51.67	40	46.28	34.22
ThinkTwice [22]	TP	E+D+S+M	27.38	18.42	35.82	50	54.23	37.17
DriveAdapter [21]	TP	E+S	28.38	28.38	47.50	50	56.43	42.14
SimLingo [38]‡	TP	-	**54.01**	57.04	88.33	53	**82.45**	67.03
UniAD-Base [18]	NC	M+B+Mo+O	8.89	9.33	20	20	14.21	14.49
VAD [25]	NC	M+B+Mo	11.43	11.43	18.64	20	19.15	18.07
DriveTransformer-L [24]	NC	M+B+Mo	17.57	35.00	48.36	40	52.10	38.60
HiP-AD [43]‡	NC	M+B+Mo	50.00	84.44	83.33	40	50.45	65.98
ReCogDrive [28]	NC	-	29.73	20.00	69.09	20	71.34	42.03
Qwen2.5 [2]	NC	B+Mo	14.27	28.91	30.11	30	24.73	25.60
ORION [11]	NC	M+B+Mo	25.00	71.11	78.33	30	69.15	54.72
GeRo (Qwen2.5) (ours)	NC	B+Mo	37.83	72.81	83.10	50	66.17	61.98
GeRo (ORION) (ours)	NC	M+B+Mo	40.06	78.24	87.32	50	76.83	66.49
GeRo (ORION) (ours)‡	TP	M+B+Mo	48.11	**81.16**	**88.66**	50	77.75	**69.14**

28,000 samples in a 22k/6k training/validation split. We additionally evaluate Qwen and GeRo (Qwen) with camera-only inputs on the NAVSIM-v1 NavTest split [5], a data-driven non-reactive simulation benchmark based on real-world driving scenes.

VQA Dataset. Language annotations for the scene description and visual question answering tasks are obtained from the ChatB2D [11] dataset in conjunction with Bench2Drive [23], and the DriveLM-nuScenes [40] dataset. We generate additional scenario descriptions and ego-action question-answer pairs for training GeRo during scenario generation rollout (more details in appendix section B).

Metrics. Bench2Drive closed-loop metrics include Driving Score (DS), Success Rate (SR), Efficiency, Comfort, and Multi-Ability. DS combines route completion and violation penalties; SR measures successful completion; Efficiency and Comfort assess speed and smoothness. Multi-Ability evaluates five advanced urban driving skills. Open-loop evaluation uses L2 trajectory error and collision rate, following ST-P3 [17]. Open-loop nuScenes evaluation uses L2 trajectory error and collision rate with ST-P3-style metrics [17] under the PARA-Drive[51]/TOKEN[45] evaluation setting. For NAVSIM, we report No at-fault Collision (NC), Ego Progress (EP), and Predictive Driver Model Score (PDMS); higher is better for all three metrics.

4.2 Implementation Details

Model Architecture. We implement the GeRo framework on two models. GeRo (Qwen) is built on the general multi-modal LLM Qwen2.5-3B-VL [1] trained for E2E motion planning. GeRo (ORION)



Figure 4: Qualitative examples of language-guided scenario rollouts using proposed GeRo on Bench2Drive. **Top:** Intersection scenarios with STOP signs and pedestrian interactions, where the ego vehicle demonstrates cautious deceleration, yielding, and safe left-turn. **Bottom:** Accident-handling scenarios under adverse weather, showing proactive hazard detection, lane-change planning, and smooth merging. Each frame includes generated textual reasoning aligned with the ego actions, highlighting temporal consistency and safety-aware decision.

is built on the VLA model ORION [11]. In each case, GeRo leverages the vision encoder, text encoder, and the LLM of Qwen2.5. The vision encoder is the EVA-pretrained vision transformer [8]. The motion prediction head is a stack of three MLP layers. The generative planning head is a variational autoencoder (VAE) from [11]. In GeRo (ORION), we utilize the auxiliary information and tasks used by the ORION model, while GeRo (Qwen) does not depend on HD maps. GeRo (Qwen) predicts trajectories conditioned only on the Navigation Command (NC), without explicit lane-center targets such as the Target Point (TP). GeRo (Qwen) also employs an anchor-free design, generating six trajectory modes aligned with those defined in Bench2Drive. Note, autoregressive scenario rollout is used during training and disabled by default at inference; Table 3 evaluates optional inference-time rollout.

Training. All experiments are conducted on 8 NVIDIA H100 GPUs. GeRo is trained for 24 epochs in Stage-1 to learn robust latent token representations, followed by 24 epochs in Stage-2 with autoregressive scenario rollouts of length T ($t \to t+T$) optimized using the proposed losses. Stage-2 consists of 20 epochs of imitation learning using the rollout loss $\mathcal{L}_{\text{roll}}$, followed by 4 epochs of reinforcement learning using the GRPO objective $\mathcal{L}_{\text{GRPO}}$. Stage-1 training takes approximately 7 days, while Stage-2 training takes approximately 10 days. We use $t = 4$ input frames, a rollout horizon $T = 8$, and a uniform stride $r \in [1, 4]$, covering up to 32 future frames. All Qwen2.5-3B parameters are updated in both stages. Training uses AdamW with cosine annealing, a learning rate of 2×10^{-4}, and weight decay 0.01. GRPO uses a rollout group size of $G = 8$, a minibatch size of 32, and a clipping parameter $\epsilon = 0.1$, with a frozen behavior policy and detached rewards, group baselines, and normalized advantages, as defined in Eq. 4.

5 Results and Discussion

We perform a thorough evaluation of GeRo in both closed-loop and open-loop planning settings. We demonstrate the capability of GeRo as a general plug-and-play framework for VLA models by implementing it on two end-to-end motion planners, Qwen2.5 and ORION. We compare the performance of GeRo (Qwen) and GeRo (ORION) against recent end-to-end planners as well as the baselines: 1) The original ORION VLA model as proposed in [11], 2) ORION (Qwen2.5) which replaces the original Vicuna language model with that of Qwen2.5, and 3) Qwen2.5 with added trajectory planning and motion prediction heads.

5.1 Closed-Loop Evaluation on Bench2Drive

Quantitative Results. Table 1 compares quantitative closed-loop performance on 220 Bench2Drive test routes. GeRo achieves substantial gains over prior state-of-the-art methods. The baseline Qwen2.5 model attains only 31.6% success rate and a driving score of 63.9, whereas GeRo (Qwen), equipped with guided scenario-generation rollouts and GRPO-based reinforcement learning, boosts performance to 79.6 driving score (+15.7) and 57.8% success rate (+26.2). Applying GeRo to ORION further improves this baseline, raising driving score from 77.74 to 81.90 (+4.16) and success rate from 54.62% to 60.10% (+5.5%). Notably, this model outperforms both baseline ORION models, which depend on HD map and bounding box annotations. A comparison of multi-ability performance is shown in Table 2. This table highlights similar trends: GeRo (Qwen) improves mean ability by 140% over the baseline Qwen2.5 (*i.e.* 25.6 $\rightarrow$ 61.98), while GeRo (ORION) delivers a 21.5%

Table 3: Driving performance using inference-time autoregressive rollout on Bench2Drive. Driving Score (DS) and Success Rate (SR) measure closed-loop driving. Ego Avg. L2 and agent minADE/minFDE measure open-loop prediction errors. ‡ indicates using geometric path waypoint supervision and SimLingo PID controller.

Method	Inference Rollout	Closed-loop		Open-loop			Latency (second/frame)↓
		DS↑	SR(%)↑	Ego Avg. L2↓	Agent minADE↓	Agent minFDE↓	
ORION [11]	No	76.9	53.9	0.63	4.16	4.85	0.292
ORION [11]	Yes	77.7	54.4	0.61	3.67	4.17	0.918
GeRo (ORION)	No	81.9	60.1	0.57	1.88	2.08	0.306
GeRo (ORION)	Yes	85.7	63.9	0.46	1.10	1.36	0.940

Table 4: Evaluation of E2E methods on nuScenes [3] validation, using PARA-Drive/TOKEN setting. † indicates zero-shot testing trained on Bench2Drive and evaluated on nuScenes. Ego-status is not used.

Method	Aux. Sup.	L2 (↓)				Collision Rate (↓)			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
UniAD	M+B+Mo+O	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
VAD-Base	M+B+Mo	0.54	1.15	1.98	1.22	0.04	0.39	1.17	0.53
GenAD	M+B+Mo	0.36	0.83	1.55	0.91	0.06	0.23	1.00	0.43
Senna	M+B+Mo	0.37	0.54	0.86	0.59	0.09	0.12	0.33	0.18
Qwen2.5	B+Mo	0.52	0.80	1.56	0.96	0.10	0.37	1.32	0.60
ORION	M+B+Mo	0.43	0.64	1.01	0.69	0.09	0.29	0.94	0.44
GeRo (Qwen2.5)†	B+Mo	0.24	0.41	0.70	0.45	0.11	0.22	0.42	0.25
GeRo (Qwen2.5)	B+Mo	0.16	0.27	0.50	0.31	0.06	0.10	0.26	0.14
GeRo (ORION)†	M+B+Mo	0.16	0.32	0.61	0.36	0.08	0.16	0.39	0.21
GeRo (ORION)	M+B+Mo	0.12	0.24	0.46	0.27	0.05	0.09	0.22	0.12

Table 5: NAVSIM-v1 NavTest results. Qwen and GeRo (Qwen) use camera-only inputs. NC: No at-fault Collision; EP: Ego Progress; PDMS: Predictive Driver Model Score. Higher is better.

Method	NC↑	EP↑	PDMS↑
DiffusionDrive [29]	98.2	82.2	88.1
Qwen	97.7	73.0	84.9
GeRo (Qwen)	98.9	87.1	89.7

Table 6: Ablation study of GeRo (Qwen) on Bench2Drive. Baseline: Qwen2.5-3B. Metrics: Driving Score (DS) and Success Rate (SR).

ID	Sc Desc	$\mathcal{Q}$	Pretraining			Scenario Generation Rollout						Closed-loop metric	
			$\mathcal{L}_{plan}$	$\mathcal{L}_{\text{mot}}$	$\mathcal{L}_{\text{VQA}}$	$\mathcal{L}_{roll}$			$\mathcal{L}_{\text{GRPO}}$			DS	SR
						$\mathcal{L}_{\text{tc}}$	$\mathcal{L}_{plan}$	$\mathcal{L}_{\text{mot}}$	Coll	TTC	$\mathcal{L}_{\text{VLA}}$		
1	✗	✗	✓									58.2	27.4
2	✗	✗	✓	✓								61.1	28.2
3	✓	✗	✓	✓	✓							63.9	31.6
4	✓	✗	✓	✓	✓	✓						67.8	40.1
5	✓	✗	✓	✓	✓	✓	✓					69.2	46.6
6	✓	✗	✓	✓	✓	✓	✓	✓				74.4	50.4
7	✓	✗	✓	✓	✓	✓	✓	✓	✓			76.9	54.4
8	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓		78.5	55.7
9	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	79.6	57.8

(*i.e.* 54.72 → 66.49) improvement over ORION. Performance gains span critical skills such as merging, overtaking, emergency braking, yielding, and traffic-sign compliance—underscoring the effectiveness of language-guided scenario rollouts and RL for robust action grounding. Appendix A.5 further analyzes behavior-level transfer between emergency braking and overtaking.

Qualitative Results. Figure 4 illustrates GeRo 's language-conditioned rollouts in complex urban environments, encompassing challenges such as occlusions, pedestrian crossings, yielding interactions, and wet road conditions. The ego vehicle dynamically adapts its trajectory in response to evolving scene context—slowing near crosswalks, yielding to oncoming or turning vehicles, and proceeding only after potential conflicts have cleared. These behaviors align with the reasoning articulated in the generated ego-action responses. Such alignment between textual intent and physical execution underscores how scenario-grounded rollouts enable safe, context-aware planning.

Inference-Time Rollout. Table 3 compares predictions with and without autoregressive rollout at inference. For GeRo (ORION), rollout improves DS/SR from 81.9/60.1 to 85.7/63.9 and reduces ego Avg. L2 from 0.57 to 0.46 and agent minADE/minFDE from 1.88/2.08 to 1.10/1.36. The geometric-waypoint/SimLingo-controller variant also improves from 87.2/68.8 to 89.9/71.2 DS/SR. These gains come with higher latency: GeRo (ORION) increases from 0.306 to 0.940 seconds per frame. We therefore disable inference-time rollout by default, while retaining it as an optional accuracy–latency tradeoff.

5.2 Open-Loop Evaluation

We further evaluate open-loop trajectory prediction on the nuScenes and Bench2Drive planning benchmarks. This assesses both generalization and motion-forecasting quality. Our proposed GeRo delivers substantial gains over Qwen2.5 and ORION baselines. In Table 1, GeRo shows a consistent

drop in trajectory error across both baselines on Bench2Drive. Table 4 contains open-loop planning results on nuScenes. Compared to Qwen2.5, GeRo (Qwen) reduces L2 trajectory error from 0.96 to 0.31 (-67.7%) and collision rate from 0.60 to 0.14 (-76.7%). In the zero-shot setting, errors still drop to 0.45 (-53.1%) with collisions at 0.25 (-58.3%), highlighting strong cross-dataset generalization enabled by scenario-grounded rollouts. Using ORION as the backbone, GeRo (ORION) further improves L2 trajectory error from 0.69 to 0.27 (-60.9%) and collision rate from 0.44 to 0.12 (-72.7%). Zero-shot again shows notable gains (0.36, -47.8%; 0.21, -52.3%). Appendix A.3 details the controlled Qwen2.5 baseline and progressive training ablation.

NAVSIM Evaluation. Table 5 complements displacement-error evaluation with planning-oriented metrics on NAVSIM-v1 NavTest. With camera-only inputs, GeRo (Qwen) improves NC from 97.7 to 98.9, EP from 73.0 to 87.1, and PDMS from 84.9 to 89.7 over Qwen. We include DiffusionDrive [29] as a literature reference. These results provide additional evidence of planning gains beyond Bench2Drive on real-world driving scenes.

5.3 Ablation Study and Further Performance Analysis

Table 6 summarizes the ablation experiments of GeRo on the Bench2Drive dataset using Qwen2.5-3B as the baseline model. We report the closed-loop metrics Driving Score (DS) and Success Rate (SR). Additional ablation study and performance analysis can be found in the appendix, including planner design, pretraining, rollout parameters, cross-skill evaluation, and rollout realism.

Pretraining. IDs 1–3 in Table 6 show the impact of pretraining losses $\mathcal{L}_{\text{plan}}, \mathcal{L}_{\text{mot}}, \mathcal{L}_{\text{VLA}}$. Incorporating scenario descriptions and visual question answering (VQA) improves baseline performance, indicating that language grounding during pretraining enhances planning robustness.

Scenario Rollout. IDs 4–6 evaluate contributions of rollout-consistency losses $\mathcal{L}_{\text{roll}}$ within the proposed GeRo framework. Even under model-based supervision—where ground truth is not required and we rely on predicted tokens $\{z_{t+1}\}$ from the pretrained VLA using input image sequence $I_{1:t}$ as reference targets—computing $\mathcal{L}_{\text{tc}}$ yields notable gains (+3.9 DS, +8.5% SR). Further incorporating ground-truth trajectories $\{\tau_{t+\Delta}\}$ for $\mathcal{L}_{\text{mot}}$ and $\mathcal{L}_{\text{plan}}$ boosts performance by an additional +6.6 DS and +10.3% SR, demonstrating the value of accurate trajectory supervision during rollouts.

RL with GRPO. IDs 7–9 analyze the effect of GRPO-based reinforcement learning losses $\mathcal{L}_{\text{GRPO}}$. Using collision and time-to-collision (TTC) rewards during scenario rollouts improves DS by +4.1 and SR by +5.3%. Finally, introducing ego-action question–answer pairs $\mathcal{Q}$ (Figure 3) and leveraging the corresponding language alignment loss $\mathcal{L}_{\text{VLA}}$ as an additional reward further enhances performance (+1.1 DS, +2.1% SR), highlighting the benefit of grounding ego intent in multi-step maneuvers through language-conditioned rollouts.

6 Conclusion

In this work, we introduced GeRo, a generative scenario rollout framework that unifies planning, multi-agent motion prediction, and scenario generation for robust end-to-end planning with Vision-Language-Action models. GeRo leverages shared latent tokenization and language-conditioned autoregressive rollout to enable temporally consistent and context-aware driving behavior. By integrating reinforcement learning with GRPO and aligning textual reasoning with action execution, our proposed GeRo sets the new state-of-the-art performance with its robust, language-guided end-to-end planning. On standard autonomous driving benchmarks, GeRo provides significant improvements under both closed-loop and open-loop settings, as well as strong zero-shot sim-to-real generalization.

Broader Impact. This work introduces Generative Scenario Rollouts (GeRo), a plug-and-play training framework for Vision-Language-Action (VLA) models that jointly performs planning and language-grounded future scenario generation via autoregressive rollouts in a learned latent token space. By enabling temporally consistent language-conditioned rollouts (and an interactive VQA component that verbalizes ego intent), GeRo can improve long-horizon reasoning and interpretability for end-to-end autonomous driving policies, pushing the boundaries of autonomous driving.

Limitations Although GeRo jointly predicts ego plans and multi-agent futures, it fundamentally plans only for the ego vehicle and treats other agents as forecasts rather than decision-making entities, which can limit fidelity in highly interactive scenarios (e.g., adversarial behaviors of other agents) where other agents react strategically.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[4] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14093–14100. IEEE, 2024.

[5] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[8] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19358–19369, 2023.

[9] Lan Feng, Yang Gao, Éloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

[10] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 910–919, 2024.

[11] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. *arXiv preprint arXiv:2503.19755*, 2025.

[12] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*, 2023.

[13] Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Haoran Yin, Xiangyu Li, Xinbang Zhang, et al. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. *arXiv preprint arXiv:2502.13144*, 2025.

[14] Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. ipad: Iterative proposal-centric end-to-end autonomous driving. *arXiv preprint arXiv:2505.15111*, 2025.

[15] Tao Han, Wanghan Xu, Junchao Gong, Xiaoyu Yue, Song Guo, Luping Zhou, and Lei Bai. Infgen: A resolution-agnostic paradigm for scalable image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17941–17950, 2025.

[16] Deepti Hegde, Rajeev Yasarla, Hong Cai, Shizhong Han, Apratim Bhattacharyya, Shweta Mahajan, Litian Liu, Risheek Garrepalli, Vishal M Patel, and Fatih Porikli. Distilling multi-modal large language models for autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27575–27585, 2025.

[17] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022.

[18] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023.

[19] Zhiyu Huang, Xinshuo Weng, Maximilian Igl, Yuxiao Chen, Yulong Cao, Boris Ivanovic, Marco Pavone, and Chen Lv. Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3445–3451. IEEE, 2025.

[20] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, Yin Zhou, James Guo, Dragomir Anguelov, and Mingxing Tan. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

[21] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *ICCV*, 2023.

[22] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023.

[23] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.

[24] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[25] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.

[26] Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025.

[27] Derun Li, Jianwei Ren, Yue Wang, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Zhongpu Xia, Peng Jia, Xianpeng Lang, et al. Finetuning generative trajectory model with reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025.

[28] Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

[29] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12037–12047, 2025.

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[31] Haochen Liu, Tianyu Li, Haohan Yang, Li Chen, Caojun Wang, Ke Guo, Haochen Tian, Hongchen Li, Hongyang Li, and Chen Lv. Reinforced refinement with self-aware expansion for end-to-end autonomous driving. *arXiv preprint arXiv:2506.09800*, 2025.

[32] Jiaqi Liu, Peng Hang, Xiao Qi, Jianqiang Wang, and Jian Sun. Mtd-gpt: A multi-task decision-making gpt model for autonomous driving at unsignalized intersections. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 5154–5161. IEEE, 2023.

[33] Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

[34] Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nick Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, Brandyn White, and Dragomir Anguelov. The Waymo Open Sim Agents Challenge. *arXiv preprint arXiv:2305.12032*, 2023.

[35] Youssef Al Ozaibi, Manolo Dulva Hina, and Amar Ramdane-Cherif. End-to-end autonomous driving in carla: A survey. In *IEEE Access*, pages 146866 – 146900. IEEE, 2024.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[37] 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.

[38] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11993–12003, 2025.

[39] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[40] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023.

[41] Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22432–22441, 2025.

[42] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024.

[43] Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. Hip-ad: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25605–25615, 2025.

[44] Ran Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. *arXiv preprint arXiv:2407.00959*, 2024.

[45] Thomas Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. In *Proceedings of The 8th Conference on Robot Learning*, pages 3656–3673. PMLR, 2025.

[46] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

[47] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[48] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[49] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*, 2024.

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[51] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024.

[52] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.

[53] Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1602–1611, 2025.

[54] Rajeev Yasarla, Shizhong Han, Hsin-Pai Cheng, Litian Liu, Shweta Mahajan, Apratim Bhattacharyya, Yunxiao Shi, Risheek Garrepalli, Hong Cai, and Fatih Porikli. Roca: Robust cross-domain end-to-end autonomous driving. *arXiv preprint arXiv:2506.10145*, 2025.

[55] Dongkun Zhang, Jiaming Liang, Ke Guo, Sha Lu, Qi Wang, Rong Xiong, Zhenwei Miao, and Yue Wang. Carplanner: Consistent auto-regressive trajectory planning for large-scale reinforcement learning in autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17239–17248, 2025.

[56] Lingjun Zhang, Changjie Wu, Linzhe Shi, Jiangyang Li, Jiaxin Liu, Lei Yang, Hang Zhang, Mu Xu, and Hong Wang. DeepSight: Long-horizon world modeling via latent states prediction for end-to-end autonomous driving. *arXiv preprint arXiv:2605.10564*, 2026.

[57] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 87–104. Springer, 2024.

[58] Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. *arXiv preprint arXiv:2501.15564*, 2025.

[59] Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025.

A Ablation Study

A.1 Ablation on Generative Planners

Table 7 presents an ablation study comparing different generative planners on Bench2Drive dataset under the closed-loop setting. The reported metrics include driving score (DS) and success rate (SR) for closed-loop performance, and average L2 trajectory error for open-loop trajectory prediction. We also provide an overall multi-ability average to summarize planner effectiveness. The results demonstrate that incorporating generative scenario rollouts significantly improves performance across all metrics. Specifically, our proposed approach (GeRo) consistently outperforms the baseline across different variants of generative planners architecture (MLP, Diffusion, or VAE). Among these, the VAE-based planner achieves the highest scores, with GeRo (ORION) w/ VAE reaching a Driving Score of 81.90 and a Success Rate of 60.10%, while reducing the L2 error to 0.57 m. These findings highlight the advantage of combining generative scenario rollouts with diverse planning strategies.

Table 7: Ablation comparison with diverse planners. DS and SR denote Driving Score and Success Rate respectively.

Generative Planner	Closed-loop		Open-loop	Multi-Ability Avg.
	DS↑	SR(%)↑	Avg. L2 (m)↓	
ORION w/ MLP	70.73	41.52	0.75	48.44
ORION w/ Diffusion	71.97	46.54	0.73	46.68
ORION w/ VAE	77.74	54.62	0.68	54.72
GeRo (ORION) w/ MLP	77.21	56.72	0.62	62.38
GeRo (ORION) w/ Diffusion	80.75	59.22	0.57	65.43
GeRo (ORION) w/ VAE	81.90	60.10	0.57	66.49

A.2 Ablation on Stage-1 Pretraining

Table 8 reports an ablation study evaluating the effect of Stage-1 pretraining on our generative planner based on Qwen2.5. For this experiment, we use Qwen2.5-3B as the baseline for GeRo. We measure closed-loop performance using driving score (DS) and success rate (SR), along with a multi-ability average summarizing overall capability. The baseline Qwen2.5 achieves a DS of 63.90 and SR of 31.60%, indicating limited performance without generative rollouts. Incorporating our approach (GeRo) without Stage-1 pretraining significantly improves results to 71.67 DS and 48.95% SR, demonstrating the benefit of scenario rollouts alone. Further applying Stage-1 pretraining yields the best performance, achieving 79.60 DS and 57.80% SR, with a Multi-Ability Average of 61.98. These results highlight the significance of stage-1 pretraining in scenario rollout generation framework of GeRo

Table 8: Ablation comparison on stage-1 pretraning. DS and SR denote Driving Score and Success Rate respectively.

Generative Planner	Closed-loop		Multi-Ability Avg.
	DS↑	SR(%)↑	
Qwen2.5	63.90	31.60	25.60
GeRo (Qwen2.5) w/o stage-1 pretraining	71.67	48.95	53.7
GeRo (Qwen2.5) w/ stage-1 pretraining	79.60	57.80	61.98

A.3 Controlled Qwen2.5 Baseline and Training Ablation

GeRo (Qwen2.5) uses the same backbone, tokenizer, decoder, inputs, and evaluator as the Qwen2.5 baseline, adding the proposed rollout-based training objectives. After Stage-1 pretraining, Stage-2 introduces rollout supervision for temporal consistency, agent motion, and ego planning, followed by GRPO-based safety feedback. The comparison thus isolates the effect of rollout training with the architecture, inputs, training data, and evaluation protocol held fixed. The baseline Avg. L2 of 0.96 is specific to this controlled setup; we do not claim that it is the best attainable Qwen2.5 result across protocols.

Table 9 reports the progressive ablation on nuScenes validation under the PARA-Drive/TOKEN evaluation protocol. Starting from Qwen2.5 trained with $\mathcal{L}_{\text{pre}}$, cumulatively adding temporal consistency, motion rollout, planning rollout, and GRPO feedback reduces Avg. L2 from 0.96 to 0.31 and

collision rate from 0.60 to 0.14. Each additional component improves both metrics. Table 6 in the main paper provides the complementary Bench2Drive ablation using closed-loop metrics and the Qwen2.5 baseline.

Table 9: Training ablation of GeRo (Qwen2.5) on nuScenes validation under the PARA-Drive/TOKEN evaluation protocol. Baseline: Qwen2.5-3B-VL. Objectives are added cumulatively; lower is better.

Stage	Model / added objective	Avg. L2↓	Collision rate↓
$\mathcal{L}_{\text{pre}}$	Qwen2.5	0.96	0.60
Stage-2 $\mathcal{L}_{\text{roll}}$	$+\mathcal{L}_{\text{tc}}$	0.61	0.31
Stage-2 $\mathcal{L}_{\text{roll}}$	$+\mathcal{L}_{\text{mot}}$	0.54	0.28
Stage-2 $\mathcal{L}_{\text{roll}}$	$+\mathcal{L}_{\text{plan}}$	0.39	0.20
Stage-2 $\mathcal{L}_{\text{GRPO}}$	$+\mathcal{L}_{\text{GRPO}}$	0.31	0.14

A.4 Ablation on T, Rollout Steps in Scenario Generation

Table 10 shows an ablation study on the number of rollout steps (T) used during scenario generation. For this experiment, we use Qwen2.5-3B as the baseline for GeRo. We report closed-loop performance using driving score (DS) and success rate (SR). Increasing T from 4 to 12 leads to consistent improvements, with DS rising from 78.31 to 80.15 and SR from 56.75% to 58.01%. This indicates that longer rollouts provide richer scenario context, improving planning quality. However, performance slightly drops at $T = 16$, suggesting diminishing returns and potential overfitting or increased complexity at very high rollout lengths.

Table 10: Ablation on T, the number of rollout steps in scenario generation.

Metric	$T = 4$	$T = 8$	$T = 12$	$T = 16$
Driving Score↑	78.31	79.60	80.15	79.08
Success Rate↑	56.75	57.80	58.01	57.17

A.5 Behavior Diversity and Cross-Skill Generalization

Future driving behavior can be multi-modal. GeRo encourages behavior-level diversity through language-conditioned rollout training, but is not a full probabilistic simulator covering all possible agent futures. The multi-ability results in Table 2 show improvements on multiple driving skills, including an increase in the mean score from 54.72 for ORION to 66.49 for GeRo (ORION). We further assess transfer between behavior modes using skill-specific training subsets.

Bench2Drive contains 44 scenarios grouped into five advanced urban-driving skills: Merging, Overtaking, Give Way, Traffic Sign, and Emergency Brake. We train models using only Emergency Brake routes or only Overtaking routes, and evaluate each model on both skills using the Bench2Drive multi-ability metric. During Stage-2 training of GeRo, we counterfactually perturb ego-action descriptions to alternative high-level behaviors, such as braking or overtaking. The model generates ego waypoints and agent trajectories conditioned on the perturbed description, while GRPO safety feedback encourages collision avoidance and high time-to-collision (TTC).

Table 11: Cross-skill evaluation on Bench2Drive. Scores use the multi-ability metric (%); higher is better. Full-training results are included for reference.

Model	Training routes	Emergency Brake↑	Overtaking↑
Qwen	Emergency Brake only	39.81	10.44
GeRo (Qwen)	Emergency Brake only	90.38	70.15
Qwen	Overtaking only	9.75	34.16
GeRo (Qwen)	Overtaking only	81.86	84.21
Qwen	Full Bench2Drive training	30.11	28.91
GeRo (Qwen)	Full Bench2Drive training	83.10	72.81

Table 11 shows substantial transfer gains: with Emergency Brake-only training, the Overtaking score improves from 10.44 for Qwen to 70.15 for GeRo (Qwen); with Overtaking-only training, the Emergency Brake score improves from 9.75 to 81.86. These results support behavior-level generalization across training subsets. They do not directly quantify the diversity or calibration of multiple plausible futures for a fixed scene, or isolate how such future diversity affects ego planning; these remain limitations of the analysis.

A.6 GPT Score, CIDEr, and BLEU Results

Table 12 presents the quality of scenario generation and ego-action question-answer (QA) pairs produced by GeRo during scenario generation rollouts. For this experiment, we use Qwen2.5-3B as the baseline for GeRo. We report quality of text generation or language outputs of GeRo using GPT score [12], CIDEr [48], and BLEU [36]. Our method demonstrates substantial improvements over the baseline across all language evaluation metrics on the Bench2Drive benchmark. For the Ego-Action task, GPT score increased from 70.82 to 77.54, indicating the effectiveness of language-grounded scenario rollouts in the GeRo framework. Similarly, CIDEr and BLEU scores improved by 7.31 and 11.69 points, respectively, reflecting stronger alignment with ego-planning actions. For Scenario Description, our approach achieved an 8.31-point gain in GPT score and a 10.80-point increase in CIDEr, confirming its ability to generate richer and more accurate scenario narratives. These consistent improvements across both tasks highlight the robustness of our method in autonomous driving.

Table 12: Language performance on Bench2Drive for ego-action QA and scenario description.

Method	Ego-Action			Scenario Description	
	GPT score↑	CIDEr↑	BLEU↑	GPT score↑	CIDEr↑
Qwen2.5	70.82	67.90	53.69	68.16	72.21
GeRo (Qwen2.5)	77.54	75.21	65.38	76.47	83.01

B Language Annotation Example

Consider a driving scene at time t consisting of an ego vehicle and a set of N_t traffic participants $\mathcal{A}_t$. Given a sequence of multi-view images, a scene description, critical object information (similar to ChatB2D [11]), and information about the state of the ego vehicle x_t^e and the states of other traffic participants $x_t^{(a_i)}$, the objective is to generate a scenario description and ego-action question-answer pairs ($\mathcal{Q}$) using Qwen3-VL-32B-VL. An example scenario description and ego-action QA pair are shown in Figure 5 and Figure 6.

Scenario Description and Ego-Action Question Answer pairs | Sample numerical annotation


```
"token": "f9878012c3f6412184c294c13ba4bac3",
"Senario description": " Pedestrain crossing at stop sign in suburban
                        street with wet road conditions",
"Critical Object information": "Pedestrian at <-2.31, 4.41> is crossing the
                               street from left to right, Pedestrian at <1.24,
                               3.83> is standing on the sidewalk. Traffic
                               signs at various distances ahead provide
                               speed limits."
"agent_attributes": {
    "c1": {
        "category": "pedestrian",
        "speed": 1.2 meters/second,
        "position": <-2.31, 4.41>,
        "assigned_cameras": [
            "CAM_FRONT",
            "CAM_FRONT_LEFT"
        ],
        "future movement": "moving",
        "future speed": "moving",
        "direction": "towards the ego vehicle",
        "interaction_with_ego_vehicle_type": "none",
        "probability_of_collision_with_ego_vehicle": "low"
    },

    ... [TRUNCATED] ...

    "c24": {
        "category": "car",
        "speed": 0.0,
        "assigned_cameras": [
            "CAM_BACK"
        ],
        "future movement": "stopped",
        "future speed": "not moving",
        "direction": "away from ego vehicle",
        "interaction_with_ego_vehicle_type": "none",
        "probability_of_collision_with_ego_vehicle": "low"
    }
```


Figure 5: An example of the text-description of the numerical annotations of a scene from Bench2Drive. This is appended to the user prompt for Qwen3-VL-32B-VL.

Generated Scenario Description and Ego-Action Question Answer pairs with Qwen3-VL-32B-VL

Scenario Description

```
"Scenario Description":  " The ego vehicle approaches an urban
intersection under heavy fog and wet road conditions, preparing
to cross while yielding to pedestrians and oncoming traffic.
After detecting a fast-approaching vehicle, it pauses
mid-intersection before safely completing the maneuver
and maintaining lane stability."
```

Ego-Action question answer pair

```
{
    "Question at step1": "What are the predicted future movements of the
    agents located to the front-left of the ego vehicle at time step 1?
    What are the predicted future movements of the ego vehicle
    at time step 1?",

    "Answer at step1": "The front-left vehicle is expected to remain
    stationary. The ego vehicle is fully stopped at the intersection,
    yielding to pedestrians crossing the road."
},
... [TRUNCATED] ...
{
    "Question at step8": "What are the predicted future movements of the
    ego vehicle at time step 8?",

    "Answer at step8": "The ego vehicle has resumed motion, entered the
    intersection, and is proceeding straight ahead, safely crossing."
},
```

Figure 6: Some examples of generated QA pairs. The perception, prediction, and planning pairs are generated with Llama-3. The behavior QA is created using the future motion of the ego vehicle.

C Scenarios

Bench2Drive contains 44 scenarios summarized into 5 advanced skills for urban driving: Merging, Overtaking, Give Way, Traffic Sign, Emergency Brake as below:

1. **Merging**
    - CrossingBicycleFlow
    - EnterActorFlow
    - HighwayExit
    - InterurbanActorFlow
    - HighwayCutIn
    - InterurbanAdvancedActorFlow
    - MergerIntoSlowTrafficV2
    - MergeIntoSlowTraffic
    - NonSignalizedJunctionLeftTurn
    - NonSignalizedJunctionRightTurn
    - NonSignalizedJunctionLeftTurnEnterFlow
    - ParkingExit
    - LaneChange

- SignalizedJunctionLeftTurn
- SignalizedJunctionRightTurn
- SignalizedJunctionLeftTurnEnterFlow

2. **Overtaking**
    - Accident
    - AccidentTwoWays
    - ConstructionObstacle
    - ConstructionObstacleTwoWays
    - HazardAtSideLaneTwoWays
    - HazardAtSideLane
    - ParkedObstacleTwoWays
    - ParkedObstacle
    - VehicleOpenDoorTwoWays
3. **Emergency Brake**
    - BlockedIntersection
    - DynamicObjectCrossing
    - HardBrakeRoute
    - OppositeVehicleTakingPriority
    - OppositeVehicleRunningRedLight
    - ParkingCutIn
    - PedestrianCrossing
    - ParkingCrossingPedestrian
    - StaticCutIn
    - VehicleTurningRoute
    - VehicleTurningRoutePedestrian
    - ControlLoss
4. **Give Way**
    - InvadingTurn
    - YieldToEmergencyVehicle
5. **Traffic Sign**
    - EnterActorFlow
    - CrossingBicycleFlow
    - NonSignalizedJunctionLeftTurn
    - NonSignalizedJunctionRightTurn
    - NonSignalizedJunctionLeftTurnEnterFlow
    - OppositeVehicleTakingPriority
    - OppositeVehicleRunningRedLight
    - PedestrianCrossing
    - SignalizedJunctionLeftTurn
    - SignalizedJunctionRightTurn
    - SignalizedJunctionLeftTurnEnterFlow
    - Tjunction
    - VanillaNonSignalizedTurnEncounterGiveWay
    - VanillaSignalizedTurnEncounterGiveWay
    - VanillaSignalizedTurnEncounterStopSign
    - VanillaNonSignalizedTurnEncounterStopSign
    - VehicleTurningRoute
    - VehicleTurningRoutePedestrian



Figure 7: Qualitative visualization of two complex urban driving scenarios under adverse weather conditions. **Top:** illustrates a left-turn maneuver at an intersection with limited visibility, requiring cautious execution and yielding to oncoming traffic. **Bottom:** depicts a pedestrian-crossing scenario where the ego vehicle detects a fast-approaching pedestrian, performs an emergency stop, and resumes motion safely after the pedestrian clears the road. These examples highlight the GeRo's ability to handle dynamic and safety-critical situations.

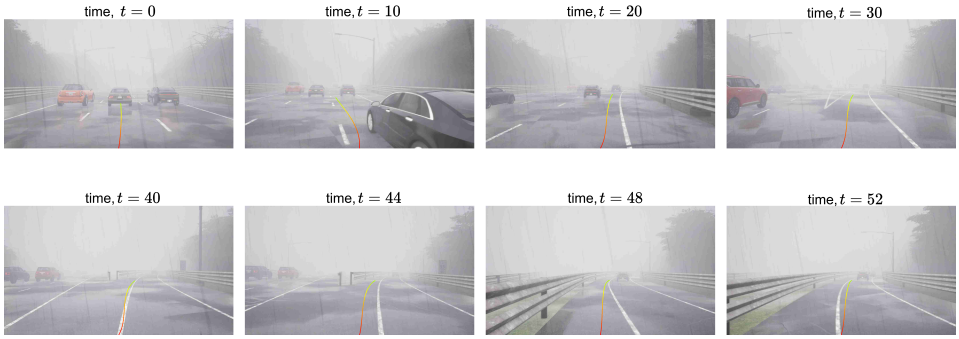


Figure 8: Failure case under extreme fog conditions. The ego vehicle initially performs a successful lane change to prepare for a highway exit (top row). However, due to low visibility, it confuses the shoulder lane with the exit lane and deviates from its route for approximately 23.476 meters (bottom row). The vehicle later corrects its trajectory and merges back into the exit lane.

D Visualizations

D.1 Qualitative Results

Figure 7 illustrates two challenging urban driving scenarios under adverse weather conditions, highlighting the ability of GeRo to adapt to dynamic and safety-critical events. In the first scenario (top row), the ego vehicle approaches an intersection with limited visibility and executes a cautious left turn while yielding to oncoming traffic. The trajectory shows smooth adjustments to maintain safety on a wet surface. In the second scenario (bottom row), the ego vehicle detects a fast-approaching pedestrian, performs an emergency stop, and remains stationary until the pedestrian clears the road before resuming motion. These examples demonstrate robust decision-making, safe maneuver execution, and compliance with traffic rules in unpredictable environments.

Qualitative Analysis of Hard Brake Route. Figure 9 illustrates the qualitative output of GeRo in the "Hard Brake Route" scenario. In this situation, surrounding traffic halts unexpectedly, requiring the ego vehicle to perform a controlled stop. The shortened trajectory indicates the ability of GeRo to decelerate smoothly and maintain safety. Once traffic resumes, the ego vehicle accelerates without abrupt maneuvers, following the planned trajectory. This behavior demonstrates the generative capability of GeRo to handle stop-and-go conditions effectively, ensuring both safety and comfort in dynamic environments.

D.2 Failure Case

Figure 8 illustrates a failure case of GeRo, where the ego vehicle deviated from its route lanes for approximately 23.476 meters. The top row of Figure 8 shows the ego vehicle successfully changing from the center lane to the rightmost lane of the highway, preparing to exit in the subsequent frames. In contrast, the bottom row depicts the impact of extreme fog and very low visibility, which caused the ego vehicle to confuse the shoulder lane with the highway exit lane. As a result, the vehicle deviated and traveled outside its designated lane for about 23.476 meters. After continuing for a short distance on the shoulder lane and encountering a roadside railing, the ego vehicle recognized the deviation and gradually merged back into the highway exit lane, resuming safe driving.

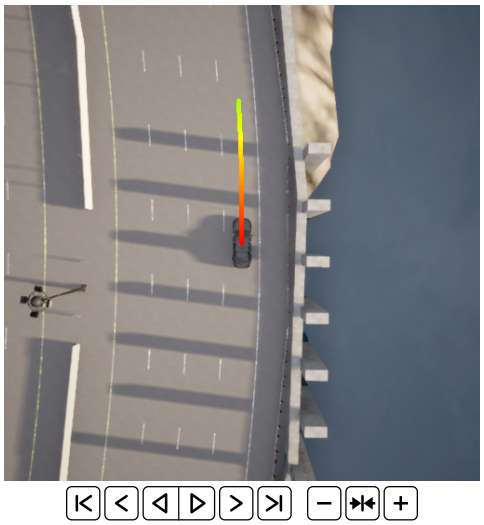


Figure 9: Visualization of the "Hard Brake Route" scenario.

E Inference Details

Table 13 reports the per-frame inference latency of GeRo measured on a single NVIDIA H100 (80GB) GPU with batch size 1. We use 10 warmup iterations followed by 50 timed iterations, and call `torch.cuda.synchronize()` before and after each timed iteration to ensure accurate GPU timing.

Table 13: **Per-frame inference cost on Bench2Drive (validation).** Time is measured in seconds per frame; compute is reported in TFLOPs.

Method	Inference	
	Time (s) ↓	TFLOPs ↓
ORION	0.292	0.796
GeRo (ORION)	0.306	0.798

F Rollout Realism

Closed-loop driving scores measure the deployed policy's performance, but do not directly assess the fidelity of the latent futures used during training. We therefore evaluate decoded rollout trajectories using metrics adapted from the Waymo Open Sim Agents Challenge [34], with ground-truth Bench2Drive trajectories as the reference distribution. All compared checkpoints are evaluated under the same protocol using the best checkpoint from each method. DeepSight [56] provides a relevant latent world-modeling comparison.

The kinematic metric averages likelihoods of linear speed, linear acceleration, angular speed, and angular acceleration, assessing agreement with reference motion statistics. The interactive metric averages likelihoods of distance to the nearest object, collision indication, and time-to-collision, assessing proximity and collision-risk statistics. The realism meta metric is a linear aggregation of these measures. We omit the map metric because distance-to-center values are unavailable for some methods. Thus, the evaluation measures trajectory-distribution agreement rather than latent-token reconstruction or exact matching of a single future.

Table 14: Latent rollout realism on Bench2Drive using adapted Waymo Sim Agents metrics. Higher values indicate better reference-distribution agreement.

Method	Meta↑	Kinematic↑	Interactive↑
ORION	0.391	0.263	0.518
SimLingo	0.421	0.368	0.573
DeepSight [56]	0.490	0.413	0.608
GeRo	0.553	0.454	0.620

Table 14 shows that GeRo improves all three metrics over the compared baselines, reaching a kinematic score of 0.454, an interactive score of 0.620, and a realism meta score of 0.553, compared with 0.391 for ORION. The interactive improvement is relevant because predicted surrounding-agent trajectories shape the collision and time-to-collision feedback used during rollout-based training. These results complement the driving evaluation by assessing agreement with reference motion and interaction statistics. They do not guarantee exact physical dynamics for every generated future or exclude exploitation of imperfect rewards.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately describe the proposed GeRo framework and its scope, and the stated claims align with the methods and experimental results presented, as summarized in the contributions at the end of the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the proposed approach are discussed in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theoretical results with stated assumptions or proofs, focusing instead on model design and empirical validation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper specifies datasets, evaluation protocols, model architectures, training procedures, hyperparameters, and compute setup in Sections 4.1 and 4.2 and the appendix, providing sufficient detail to reproduce the main experimental results that support the paper's claims.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: While all datasets used in the paper (e.g., Bench2Drive, nuScenes, ChatB2D) are publicly available, the code and trained models are not yet released due to pending institution legal and compliance review; therefore, full open access to data and code is not currently provided at the time of submission.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper clearly specifies datasets and splits, model architectures, training stages, hyperparameters, optimizers, rollout settings, and evaluation protocols in Sections 4.1–4.2 and the appendix, providing sufficient detail to understand and interpret the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experimental results are reported as single-run benchmark metrics without error bars or statistical significance analysis, which is standard practice for closed-loop autonomous driving benchmarks but does not quantify variability across runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the compute hardware used (8×NVIDIA H100 GPUs) and, with the added reporting of execution time per inference, provides sufficient information to estimate memory usage and reproduce the computational requirements of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics by using publicly available datasets, avoiding human-subject data collection, preserving anonymity, and presenting results transparently without ethical or legal violations.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses the broader impact in Section 6; there are no negative societal impacts associated with this work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer:[Yes]

Justification: The paper does not release models or code and relies solely on publicly available datasets; any future release of artifacts is subject to institution legal and compliance review, which serves as a safeguard to ensure responsible and controlled dissemination and to mitigate potential misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets used in the paper, including datasets (e.g., Bench2Drive, nuScenes, ChatB2D) and models (e.g., Qwen2.5, ORION), are properly cited, and their

licenses and terms of use are publicly available and respected; no assets are redistributed in violation of their licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: While the paper describes how scenario descriptions and ego-action question–answer pairs are generated and used during training, no new datasets, models, or code are released as standalone assets, so this question is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects; all experiments use simulated driving benchmarks and existing datasets without direct human participation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects; all experiments are conducted using simulated environments and publicly available datasets, so no study participants or IRB approval are applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Large language models are a core component of the proposed method, serving as the backbone for scenario description generation, ego-action question answering, and autoregressive rollout; their role, configurations, and usage are described throughout Sections 3 and 4 and in the appendix.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.