Intend, Reflect, Refine: An Adaptive Multimodal Reflection Framework for Autonomous Driving

Zisheng Chen[1,*], **Yuping Qiu**[2,*], **Jianhua Han**[3], **Tao Tang**[1], **Xiuwei Chen**[1], **Likui Zhang**[1], **Ying-Cong Chen**[2], **Hang Xu**[3], **Xiaodan Liang**[1,†]

[1] Shenzhen campus of Sun Yat-sen University [2] HKUST(GZ)
[3] Yinwang Intelligent Technology Co. Ltd.

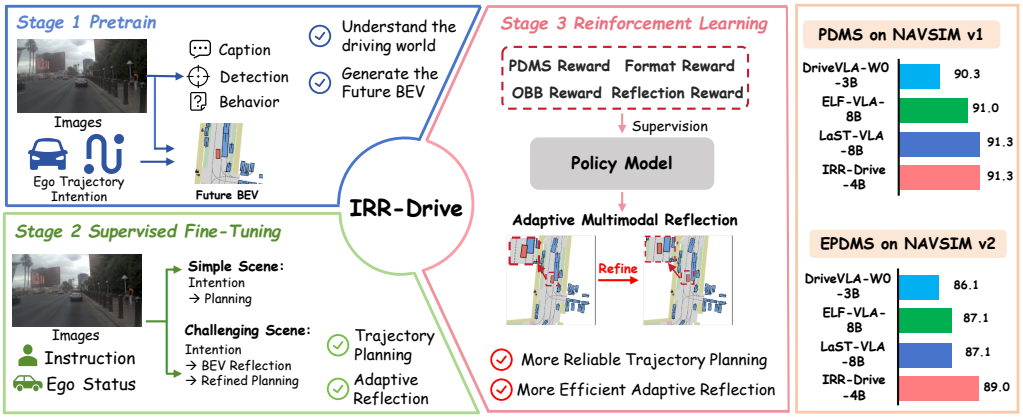


Figure 1: Visual overview of IRR-Drive. The model learns through three progressive stages. We first pretrain the model on driving world understanding and future BEV generation to establish foundational perception and future forecasting capability. Then, we fine-tune the model, applying direct execution for simple scenes and BEV-based reflection for challenging ones to learn adaptive reasoning. Finally, we use reinforcement learning, incorporating multiple reward functions to refine the policy, enabling more reliable trajectory planning and more efficient adaptive reflection. The results on the right show that IRR-Drive-4B achieves highly competitive PDMS and superior EPDMS scores on NAVSIM benchmarks despite a smaller parameter scale.

Abstract

Recent Vision-Language-Action (VLA) models have advanced end-to-end autonomous driving by incorporating reasoning for better interpretability and planning quality. However, most existing VLA approaches directly generate the final trajectory without explicitly examining its future consequences, which limits their reliability in complex and dynamic environments. To address this limitation, we propose **IRR-Drive** (**Intend, Reflect, Refine**), an adaptive multimodal reflection framework for autonomous driving. Specifically, to tightly couple high-level reasoning with physical constraints, IRR-Drive first generates an initial textual trajectory intention and anticipates potential interactions by predicting future semantic bird's-eye view (BEV) representations. This dual-modal (Text + BEV) reflection space explicitly models anticipated scene evolution, enabling the model to perform grounded self-correction and refine its initial intent before generating

*These authors contributes equally
†Corresponding Author

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

the final trajectory. Furthermore, to balance planning performance and computational efficiency, we construct reflection-oriented training data and design an adaptive reflection reward, enabling the model to adaptively select its reasoning mode according to scene complexity. Instead of using reasoning primarily as an auxiliary interpretation, IRR-Drive directly integrates an adaptive reflection mechanism into the planning framework, enabling grounded, decision-aware trajectory correction that is driven by scene complexity. Our method achieves state-of-the-art performance on the NAVSIM benchmark in both PDMS and EPDMS. Extensive experiments demonstrate the effectiveness of our multimodal reflection framework and validate the efficacy of the proposed adaptive reflection strategy.

1 Introduction

Recent Vision-Language-Action (VLA) approaches have advanced end-to-end autonomous driving by integrating perception, reasoning, and planning within a unified framework [19; 46; 41; 43]. By incorporating reasoning mechanisms, these models improve interpretability and enable more informed decision-making than purely reactive policies. However, the majority of current methods produce trajectories in a single pass, lacking an explicit assessment of how candidate actions will shape future states. Effective driving requires not only producing a feasible trajectory, but also evaluating and revising it before execution. This motivates reflective planning, where candidate actions are assessed and refined according to their anticipated outcomes. Despite recent progress, current reasoning paradigms remain limited in supporting such grounded reflective decision-making.

A key challenge lies in the lack of a suitable representation space for plan verification and correction. Existing reasoning-based driving methods [20; 19; 46] often express intermediate reasoning through textual or symbolic abstractions, which are interpretable but too coarse to capture fine-grained spatial interactions required for trajectory refinement. In contrast, vision-based approaches [43; 21] preserve richer physical details, yet typically use them as implicit features or auxiliary supervision rather than as an explicit space for evaluating planned actions. As a result, reasoning remains weakly grounded in the physical consequences of actions, limiting its effectiveness in complex driving scenarios.

To address these limitations, we propose **IRR-Drive** (**Intend, Reflect, Refine**), an action-conditioned multimodal framework with adaptive reflection for autonomous driving. Given an input scene, IRR-Drive first generates an initial trajectory intention. Conditioned on both the current scene and this intention, it predicts future semantic bird's-eye view (BEV) representations, which serve as a structured reflection space for analyzing future interactions and risks. The model then refines the initial intention based on this multimodal reflection and produces the final trajectory. To improve efficiency, IRR-Drive further learns adaptive reflection behavior, directly outputting trajectories in simple scenes while invoking reflective refinement in challenging ones. Thus, reflection is no longer a static post-hoc explanation, but an adaptive, decision-aware process for trajectory correction.

In implementation, we first pretrain the model on driving understanding data and semantic BEV generation data, then construct NAVSIM-derived [6] non-reflection and multimodal reflection data for supervised fine-tuning. This enables the model to operate in both non-reflective and reflective reasoning modes. Finally, we apply GSPO reinforcement learning with rewards that jointly consider planning quality, trajectory alignment, and reflection efficiency, encouraging a better trade-off between closed-loop performance and computational cost. Our main contributions are summarized as follows:

- We design a multimodal reflection module that integrates textual reasoning with semantic BEV prediction, providing a grounded space for trajectory verification and refinement.
- We develop an adaptive reflection strategy that selects direct prediction or reflective refinement based on scene complexity, improving efficiency while preserving planning quality.
- We conduct extensive experiments on NAVSIM v1 and v2, demonstrating strong closed-loop planning performance and validating the effectiveness of multimodal reflection, adaptive mode selection, and reinforcement learning.

2 Related Work

Reasoning and Self-Reflection in VLA. Building on the VLA paradigm, recent work has placed increasing emphasis on explicit reasoning in driving decision-making. FSDrive [43] replaces purely

textual reasoning with visual spatio-temporal chain-of-thought by generating future visual states, thereby preserving finer spatial details during deliberation. AutoVLA [46] unifies reasoning and action generation within a single autoregressive framework and incorporates adaptive reasoning together with reinforcement fine-tuning to improve planning performance. AdaThinkDrive [19] introduces fast and slow reasoning modes, allowing the model to allocate inference depth according to scene complexity. Beyond descriptive reasoning, self-reflective planning has also emerged as an important direction. Counterfactual VLA (CF-VLA) [24] combines time-segmented meta-actions with counterfactual analysis to revise planned behaviors based on simulated outcomes. ELF-VLA [20] leverages teacher-generated structured diagnoses and corrected trajectories to perform failure-driven refinement during training.

Vision-Language and Vision-Language-Action Models for Autonomous Driving. The integration of large vision-language models into autonomous driving enables semantic-level decision-making grounded in scene understanding and commonsense knowledge. DriveGPT4 [38] is among the earliest efforts to jointly model driving interpretation and action prediction within a unified vision-language framework. To address the limited precision of language models in continuous planning, Senna [11] decouples semantic decision-making from trajectory prediction, using language-based meta-actions as the interface between a high-level LVLM and a low-level planner. ReCogDrive [16] further addresses domain gap, language–action mismatch, and imitation bias through a three-stage pipeline consisting of driving VQA pretraining, a cognitive-guided diffusion planner, and reinforcement learning fine-tuning. These works demonstrate the promise of the VLA paradigm for autonomous driving, while most of them still rely on single-pass generation without an explicit feedback mechanism for validating the physical consequences of planned behaviors.

Future Scene Modeling for Autonomous Driving. Another closely related line of work incorporates future prediction or world modeling into driving planning. ReasonPlan [18] combines self-supervised next-scene prediction with decision chain-of-thought to jointly model scene forecasting and decision reasoning in closed-loop driving. Epona [44] proposes an autoregressive diffusion world model for long-horizon driving video generation and applies the learned world model to planning. DriveVLA-W0 [15] addresses the supervision deficit in VLA training by using future image prediction as a dense self-supervised signal. DriveWorld-VLA [10] further couples world dynamics with the planner in latent space, enabling decision-making to directly leverage latent representations of future scene evolution. Collectively, these methods incorporate future scenario modeling in different forms, including auxiliary supervision, reasoning support, and latent planning.

3 Method

In this section, we describe **IRR-Drive**. Sec. 3.1 presents the semantic BEV tokenizer, which converts BEV images into discrete codes. Sec. 3.2 describes the construction of pretraining, supervised fine-tuning, and RL training data. Sec. 3.3 introduces the two-stage supervised fine-tuning procedure, and Sec. 3.4 presents reinforcement learning for improving planning performance and adaptive reflection.

3.1 Semantic BEV Tokenizer

Inspired by autoregressive image generation paradigms [7; 31] and unified vision-language models (VLMs) [9; 4; 34], we develop a semantic BEV tokenizer that discretizes BEV into code tokens, thereby enabling unified autoregressive reasoning and planning. We adopt Qwen-VL visual-semantic features to represent BEV, aligning the BEV representation with the semantic space of multi-view visual inputs and facilitating their unified modeling within the VLM. Furthermore, BEV provides a unified spatial representation of road structures, surrounding vehicles, and drivable areas, while NAVSIM offers trajectory-conditioned future BEV observations for direct supervision of future scene prediction.

Specifically, given a BEV representation $B \in \mathbb{R}^{H\times W\times C}$, we first extract semantic features using a frozen encoder $E_\theta(\cdot)$: $F = E_\theta(B)$. The features are projected via an MLP $g_\phi(\cdot)$ into a quantization space: $H = g_\phi(F)$. We then apply vector quantization with a learnable codebook $Z = \{z_k\}_{k=1}^{K}$, where feature is mapped to its nearest code:

$$\mathbf{Z_q}, I_q = \underset{k\in\{1,\dots,K\}}{\arg\min} \|H - \mathbf{Z}[\mathbf{k}]\|_2^2 \tag{1}$$

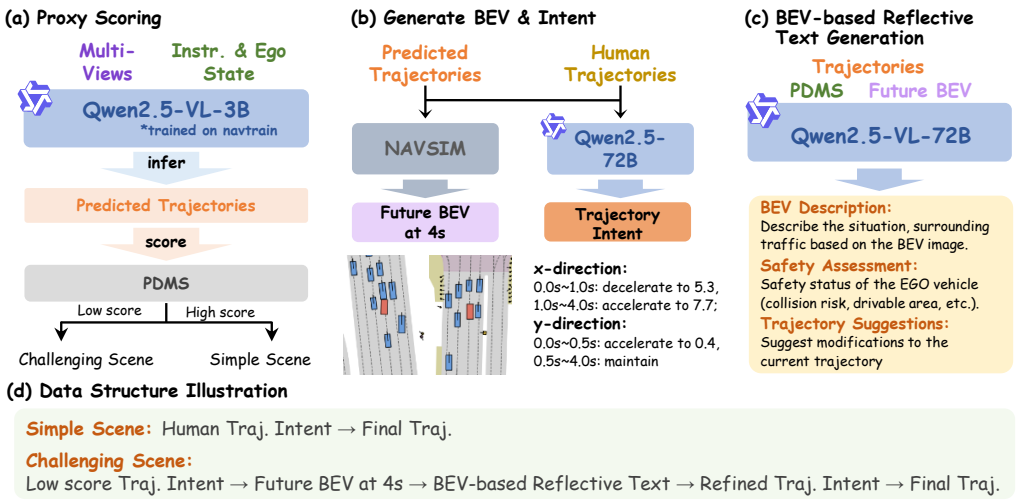


Figure 2: **Adaptive multimodal reflection data construction.** (a) A lightly fine-tuned planner is used to split the NAVSIM navtrain set into challenging and simple scenes based on predicted PDMS. (b) The simulator generates future BEV representations, while the LLM generates trajectory intents. (c) A VLM generates BEV-grounded reflective text. (d) The resulting data structures for simple and challenging scenes.

Where K is the codebook size, $\mathbf{Z_q}$ is the quantized feature, and I_q is the quantized index. The quantized features are then decoded by $D_\psi(\cdot)$ to reconstruct the semantic representation: $\hat{F} = D_\psi(Z_q)$. The tokenizer is trained with a standard VQ objective:

$$\mathcal{L} = \mathcal{L}_{sem}(F, \hat{F}) + \|\text{sg}[H] - Z_q\|_2^2 + \beta\|\mathbf{H} - \text{sg}[\mathbf{Z}_q]\|_2^2 \tag{2}$$

where $\mathcal{L}_{sem}$ represents the mean squared error loss, $\text{sg}[\cdot]$ denotes the stop-gradient operator.

The semantic BEV tokenizer converts BEV representations into discrete codes, thereby enabling seamless integration with existing VLA frameworks. Please refer to **Supplementary Material A.1** for the semantic training framework and the corresponding preprocessing illustrations.

3.2 Data Preparation

To equip the VLM with driving scene understanding, BEV semantic forecasting, and adaptive multimodal reflection capabilities, we perform a data preparation stage as follows:

a) Pretraining Data: To adapt VLMs to autonomous driving scenarios, we leverage diverse multimodal datasets for both driving scene understanding and BEV modeling. Specifically, we incorporate open-source understanding datasets, including DriveLM [28], NuInstruct [36], nuScenesQA [25], CODA-LM [14], and others [22; 38; 16]. In addition, we construct BEV reconstruction and prediction data using the NAVSIM v1 [6] simulator. The reconstruction data maps current multi-view observations to BEV representations, capturing the present spatial layout. The prediction data maps observations and future trajectories to future BEV representations, enabling trajectory-conditioned scene forecasting.

b) Supervised Fine-Tuning Data: To endow the model with trajectory planning capability, we utilize the official NAVSIM v1 navtrain split, denoted as $\mathcal{D}_{traj}$. Each sample takes scene context and ego states as inputs and provides future trajectories over a 4-second horizon with a 0.5-second sampling interval as outputs. To support adaptive multimodal reflection, we further construct reflection data from $\mathcal{D}_{traj}$. As shown in Fig. 2, we first train a base VLA on $\mathcal{D}_{traj}$ and use its predicted PDMS scores to split the data into challenging and simple scenes. The bottom 4,000 low-scoring samples are labeled as challenging scenes ($\mathcal{D}^{+}_{refl}$), while the remaining samples are treated as simple scenes ($\mathcal{D}^{-}_{refl}$). For each sample, we use the predicted trajectory to generate an intent-conditioned future BEV with the NAVSIM v1 simulator, and use the human trajectory to generate the target trajectory intention with an LLM. For low-scoring samples, failures such as collisions or insufficient safety margins become spatially observable in the predicted future

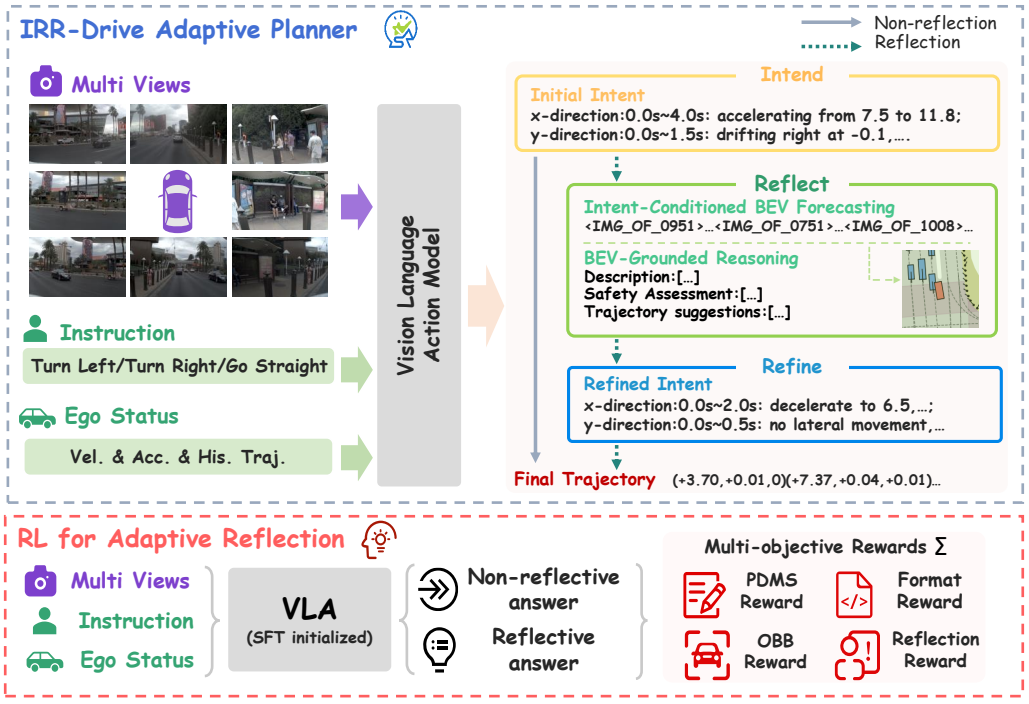


Figure 3: We present **IRR-Drive**, an end-to-end autonomous driving framework that adaptively selects between "Non-Reflection" and "Reflection" modes depending on scene complexity. Within its reflection framework, it integrates visual and textual reasoning to refine intentions and trajectories. In the reinforcement learning stage, multiple rewards, including PDMS, Format, and OBB rewards, are combined with the proposed adaptive reflection reward.

BEV, providing grounded evidence for reflection. We then feed the trajectory, future BEV representation, and corresponding PDMS score into an open-source VLM to generate the textual reflection component. Fig. 2(d) shows the data structure of the final reflection dataset. For simple scenes, the model infers the intention and generates the trajectory. For challenging scenes, the model infers the intention, performs multimodal reflection, refines it, and then generates the final trajectory. For detailed data examples, please refer to **Supplementary Material A.2**.

*c) **RL Training Data:*** We observe that during the RL training stage, a substantial portion of rollout sequences receive maximal rewards, resulting in a lack of discriminative training signals. To address this, we introduce a score-and-variance-based sampling strategy to select more informative training samples. Specifically, we estimate the reward distribution of each sample in $\mathcal{D}_{refl}$ using the SFT-trained model under stochastic rollouts. Based on the resulting score statistics, we first select samples with low rewards, and then identify those with high score variance as informative instances to form the subset $\mathcal{D}^{rl}_{refl}$. The pseudocode is available in **Supplementary Material A.3**.

3.3 Two-Stage Supervised Fine-tuning

In the first stage, we fine-tune the model using the pretraining data from Sec. 3.2, including driving VQA datasets and BEV reconstruction and prediction data. This stage enables the model to perceive drivable regions, understand road conditions, and learn the mapping from multi-view images to BEV.

In the second stage, we perform supervised fine-tuning on $\mathcal{D}_{traj}$ and $\mathcal{D}_{refl}$ (defined in Sec. 3.2) to equip the model with capabilities for trajectory planning and adaptive multimodal reflection. For each scene query $q = \{q_{cam}, q_{ego}, q_{his}\}$, outputs are supervised by o^{traj} from $\mathcal{D}_{traj}$ and o^{refl} from $\mathcal{D}_{refl}$, with the objective of maximizing conditional likelihood:

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(q,o)\sim\{\mathcal{D}^{traj},\mathcal{D}^{refl}\}}\left[-\log \pi_\theta(o \mid q)\right]. \tag{3}$$

This stage enables model to produce diverse response modes for subsequent reinforcement learning.

3.4 Adaptive Reflection via Reinforcement Learning

The two-stage supervised fine-tuning equips the model with an adaptive multimodal reflection mechanism, which directly drives its trajectory planning capabilities. However, imitation learning is constrained by predefined reflection labels and demonstration trajectories. Since these static targets may be suboptimal under closed-loop evaluation, the learned policy is limited to reproducing the training distribution and cannot explore alternative behaviors through interaction with the environment. To overcome this limitation and optimize the planner under closed-loop physical feedback, we further introduce reinforcement learning.

Due to the substantial length difference between reflective and non-reflective responses, token-level policy optimization can bias training toward longer reflective outputs and destabilize mode selection. To mitigate this issue, we adopt GSPO [45], which optimizes at the sequence level and provides a more balanced learning signal across reasoning modes. Furthermore, our reward function is composed of four components:

a) PDMS Reward: We directly utilize the PDMS score as the reward signal, which assesses trajectory safety and performance via the NAVSIM v1 [6] simulator. The score is represented as a continuous value ranging from 0 to 1, and is formulated as follows:

$$\text{PDMS} = \text{NC} \cdot \text{DAC} \cdot \frac{5 \cdot \text{TTC} + 5 \cdot \text{EP} + 2 \cdot \text{C}}{12} \tag{4}$$

where PDMS integrates five sub-metrics: No At-Fault Collision (NC), Drivable Area Compliance (DAC), Time-to-Collision (TTC), Comfort (C), and Ego Progress (EP) to produce a comprehensive closed-loop planning score.

b) Concentric OBB-FDE Reward: To evaluate spatial precision, we introduce a concentric OBB-FDE reward (R_{obb}), shown in Fig. 4 (a). Instead of continuous distance penalties, this mechanism assigns a discrete, decaying reward based on the tightest bounding box from a set of concentrically scaled tiers (e.g., L1 for $1.0\times$ scale, L2 for $1.25\times$ scale) that encloses the predicted endpoint. This stepwise design prevents overfitting to suboptimal human trajectories, enabling broader reinforcement learning exploration. Crucially, by projecting the spatial deviation into the local coordinate frame of the ground truth pose to evaluate this required scaling factor against the ego-vehicle's asymmetric physical dimensions, specifically the front overhang (l_{front}), rear overhang (l_{back}), and half-width (w_{half}), R_{obb} intrinsically accounts for anisotropic physical occupancy. This provides a more accurate geometric constraint. Detailed computation of the Concentric OBB-FDE Reward is provided in the **Supplementary Material A.4**.

c) Adaptive Reflection Reward: To prevent the model from collapsing into a single mode of reasoning, we introduce an adaptive reflection reward (R_{refl}). Since the scene tags in D_{refl} are generated in an earlier stage, they are static and no longer aligned with the model's current capabilities, making simple policy alignment unsuitable. In contrast, our design introduces a dynamic mechanism to adapt to changes in the model's capabilities during training, as shown in

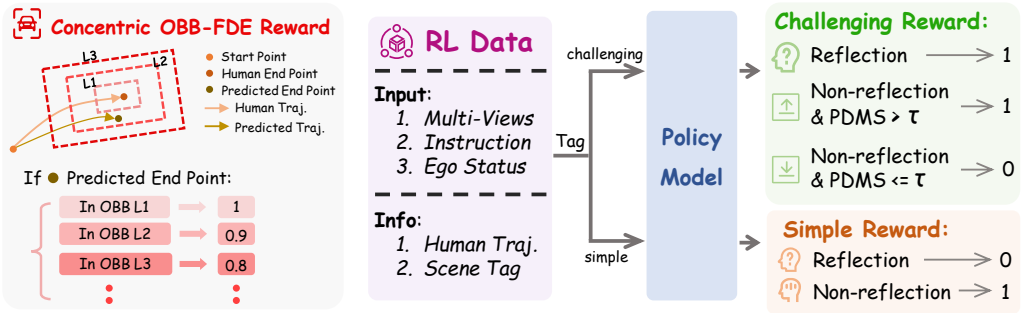


Figure 4: (a) The concentric OBB-FDE reward assigns tiered endpoint rewards using heading-aware bounding boxes around the expert endpoint. (b) The adaptive reflection reward encourages direct prediction in simple scenes, while allowing non-reflection in challenging scenes only when the current rollout already achieves sufficiently high PDMS.

Table 1: Comparison on NAVSIM v1. Abbreviation: Diff. (Diffusion), Comf. (Comfort), Cam (Camera), L (LiDAR).

Method	Backbone	Input	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
Traditional End-to-End Methods								
VADv2 [3]	–	Cam	97.2	89.1	91.6	100	76.0	80.9
TransFuser [5]	–	Cam + L	97.7	92.8	92.8	100	79.2	84.0
Hydra-MDP++ [13]	–	Cam + L	98.3	96.0	94.6	100	78.7	86.5
Artemis [26]	–	Cam	98.3	95.1	94.3	99.8	81.4	87.0
DiffusionDrive [17]	–	Cam + L	98.2	96.1	94.8	100	82.2	88.1
DriveDPO [27]	–	Cam + L	98.5	98.1	94.8	99.9	84.3	90.0
VLA Methods w/o Explicit Reasoning								
DrivingGPT [38]	LLaMA2-7B [30]	Cam	98.1	90.7	94.9	95.6	79.7	82.4
Epona [44]	DiT-2.5B [23]	Cam	97.9	95.1	93.8	99.9	80.4	86.2
ReCogDrive [16]	InternVL3-8B [47]	Cam	98.2	97.5	95.2	99.9	83.5	89.6
DriveVLA-W0 [15]	Qwen2.5-VL-3B [32]	Cam	98.7	99.1	95.3	99.3	83.3	90.3
VLA Methods w/ Textual Reasoning								
AutoVLA [46]	Qwen2.5-VL-3B [32]	Cam	98.4	95.6	98.0	99.9	81.9	89.1
AdaThinkDrive [19]	InternVL3-8B [47]	Cam	98.4	97.8	95.2	100	84.4	90.3
AutoDrive-R^2 [42]	Qwen2.5-VL-7B [32]	Cam	98.5	95.9	95.4	100	82.7	89.1
ELF-VLA-8B [20]	InternVL3-8B [47]	Cam	98.9	98.1	96.0	100	85.3	91.0
VLA Methods w/ Visual Reasoning								
FSDrive [43]	Qwen2-VL-2B [33]	Cam	98.2	93.8	93.3	99.9	80.1	85.1
DAP [40]	MiMo-VL-7B [12]	Cam+L	95.2	97.4	96.5	100	82.2	87.2
PWM [8]	Show-o [35]	Cam	98.6	95.9	95.4	100	81.8	88.1
LaST-VLA [21]	InternVL3-8B [47]	Cam	98.7	97.9	95.6	100	86.8	91.3
IRR-Drive-3B	Qwen2.5-VL-3B [32]	Cam	98.0	97.4	94.4	100	87.4	90.6
IRR-Drive-4B	Qwen3-VL-4B [1]	Cam	98.0	98.3	93.7	100	88.5	91.3

Fig. 4 (b). Specifically, for simple scenes, we disable reflection and enforce a direct reasoning mode. For difficult scenes, however, we impose a score-based requirement: the model may use direct reasoning if the PDMS score exceeds τ, and is otherwise required to perform reflection for refinement. Through this adaptive mechanism, the model learns to allocate its computational resources more intelligently, which improves both reasoning quality and inference efficiency.

d) Format Reward: The format reward R_{format} enforces adherence to the predefined output format, including the correct placement and usage of all required tags as well as valid reflection and trajectory content.

The overall reward function in the reinforcement learning stage is defined as follows:

$$R = \lambda_1(R_{\text{PDMS}} + R_{\text{OBB}}) + \lambda_2 R_{\text{refl}} + \lambda_3 R_{\text{format}}. \tag{5}$$

where $\lambda_1, \lambda_2, \lambda_3$ denote the weight coefficients of each reward term, which balance their contributions to the total reward R.

Through these reward designs, the policy learns an adaptive reflection strategy during reinforcement learning and explores diverse alternatives around suboptimal human trajectories.

4 Experiments

4.1 Experimental Setup

Dataset: We perform comprehensive experiments on NAVSIM [6], a data-driven non-reactive simulation benchmark for autonomous driving planning, built upon OpenScene. In addition, the driving-domain pretraining data, the NAVSIM BEV reconstruction/prediction data, and the adaptive reflection data are constructed as described in Sec. 3.2.

Metric: We evaluate our method on closed-loop benchmark NAVSIM v1 [6] and NAVSIM v2 [2]. The NAVSIM v1 benchmark provides a nonreactive simulation environment and employs the Predictive

Table 2: Comparison with state-of-the-art methods on NAVSIM v2 with EPDMS.

Method	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
Traditional End-to-End Methods										
HydraMDP++ [13]	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
DriveSuprim [39]	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1
DiffusionDrive [17]	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
VLA Methods w/o Explicit Reasoning										
RecogDrive-8B [16]	98.3	95.2	99.5	99.8	87.1	97.5	96.6	98.3	86.5	83.6
WAM-Flow-1.5B [37]	98.5	94.5	99.5	99.8	86.9	96.8	97.4	97.6	73.9	84.7
DriveVLA-W0-3B [15]	98.5	99.1	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
VLA Methods w/ Textual Reasoning										
ELF-VLA-8B [20]	98.9	98.1	99.4	99.8	88.5	98.4	96.9	98.3	87.2	87.1
Senna-2-3B [29]	98.5	97.8	99.5	99.8	88.1	97.5	97.0	98.6	88.4	86.6
VLA Methods w/ Visual Reasoning										
LaST-VLA-8B [21]	98.7	97.9	99.2	99.7	90.3	98.2	96.6	98.3	86.3	87.1
IRR-Drive-3B	97.3	97.4	98.7	99.6	91.9	97.1	95.7	97.9	76.9	87.9
IRR-Drive-4B	97.0	98.3	98.9	99.5	92.3	96.8	95.8	97.6	82.2	89.0

Table 3: Comparison with different reasoning modes. **Runtime** denotes the average inference time per sample on navtest, measured on a single H800 GPU.

Model	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑	Runtime (s)
Non-Reflection	98.1	97.9	93.4	100	87.6	90.6	1.46
Reflection	98.5	97.9	95.2	100	86.2	90.8	3.03
IRR-Drive	98.0	98.3	93.7	100	88.5	91.3	1.70

Driver Model Score (PDMS) as its closed-loop planning metric, as defined in Equation 4. Similarly, the NAVSIM v2 introduces the Extended Predictive Driver Model Score (EPDMS). For the details of EPDMS, please refer to **Supplementary Material A.5**.

Training Details: We use Qwen2.5-VL-3B [32] and Qwen3-VL-4B [1] as the base model, with the corresponding semantic encoder used to initialize the encoder of the semantic BEV tokenizer. The ablation experiments are conducted using Qwen3-VL-4B. For more training details, please refer to **Supplementary Material A.6**.

4.2 Performance Comparison

NAVSIM Benchmark. Tab. 1 and Tab. 2 present comparisons of our method with others on the NAVSIM benchmark. On NAVSIM v1, our method achieves competitive performance, reaching 91.3 PDMS, outperforming traditional end-to-end methods and recent VLA methods. In addition to PDMS, our model consistently improves trajectory-related metrics such as DAC and EP, indicating enhanced decision consistency and control quality. On the more challenging NAVSIM v2 benchmark, our advantages become more pronounced. IRR-Drive achieves 89.0 EPDMS, surpassing the previous best result (e.g., ELF-VLA-8B, 87.1) by a clear margin of +1.9. Meanwhile, we achieve the highest EP (92.3), demonstrating stronger task-completion capability in complex scenes. These results suggest that our method achieves stronger generalization and robustness.

Adaptive Reflection Performance. We modify the format reward function in the reinforcement learning stage to enforce single-mode reasoning during exploration. As shown in Tab. 3, applying a uniform Reflection mode across all scenarios incurs a substantial computational overhead, increasing the runtime from 1.46s to 3.03s while yielding only a marginal +0.2 improvement in PDMS over the Non-Reflection baseline. This result highlights the inefficiency of indiscriminate reflective reasoning. In contrast, IRR-Drive adaptively allocates computation only when necessary. Consequently, it achieves the best trade-off, reaching 91.3 PDMS with only 1.70s runtime, demonstrating both the effectiveness and efficiency of the proposed adaptive reflection reward function.

Table 4: **Ablation on Reward Functions. Refl. Ratio** denotes the proportion of samples that adopt reflection during inference on the navtest.

ID	PDMS	Adaptive Reflection	Concentric OBB-FDE	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑	Refl. Ratio
1				98.6	95.6	95.4	100	81.1	87.6	11.1%
2	✓			98.0	95.9	91.7	99.9	84.1	88.9	1.9%
3	✓	✓		96.9	97.4	91.1	99.9	90.8	90.5	22.9%
4	✓	✓	✓	98.0	98.3	93.7	100	88.5	91.3	20.6%

Table 5: **Ablation on CoT content.** Text+BEV reasoning achieves the best performance after both SFT and RL, showing the complementarity between textual reasoning and BEV reflection.

Text	BEV	SFT	RL					
		PDMS↑	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
✗	✗	87.3	97.5	97.0	93.0	100	87.4	89.9
✓	✗	87.2	97.9	97.6	93.5	100	87.9	90.6
✗	✓	87.2	98.8	97.5	96.4	100	84.8	90.5
✓	✓	87.6	98.0	98.3	93.7	100	88.5	91.3

4.3 Ablation Studies

Ablation on Reward Functions. Tab. 4 illustrates the impact of different reward combinations on closed-loop metrics. Experiment 1 reflects the pre-RL model performance and is adopted as the baseline. With only the basic PDMS reward, the model achieves a PDMS score of 88.9, representing an improvement of only 1.3 over the baseline. Additionally, its reasoning mode collapsed, with a reflection rate of only 1.9%. Adding the adaptive reflection reward significantly improves the score to 90.5, and its adaptive reflection mode is effectively explored, achieving a 22.9% reflection rate. Incorporating our proposed Concentric OBB-FDE reward further elevates the PDMS metric to a peak of 91.3, while maintaining a stable reflection rate of 20.6%. These results demonstrate that the Concentric OBB-FDE reward effectively strikes an optimal balance between accommodating sub-optimal human trajectories and facilitating RL exploration, thereby further enhancing the model's performance.

Ablation on CoT Content. Tab. 5 presents an ablation study on the impact of different CoT content on NAVSIM v1. Results show that relying on a single modality (either Text or BEV) yields limited gains in the RL stage. In contrast, combining Text and BEV yields the best overall performance. The multimodal CoT achieves the highest PDMS in the SFT stage (87.6) and further benefits from RL fine-tuning, reaching 91.3. These results suggest that the spatial geometric information from BEV and the reasoning capability of text are complementary, and their integration is crucial for improving CoT effectiveness in trajectory planning.

Table 6: **Ablation on Group Size.** A group size of 8 yields the best performance, indicating a favorable trade-off between rollout diversity and training efficiency.

Group Size	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
w/o RL	98.6	95.6	95.4	100	81.1	87.6
4	97.7	97.7	92.7	99.9	86.2	90.8
8	98.0	98.3	93.7	100	88.5	91.3
12	98.5	97.9	95.2	100	86.2	91.2

Table 7: **Ablation on Training Stages.** Pretraining improves generalization over SFT alone, while RL fine-tuning further enhances closed-loop planning performance, leading to the best overall result.

Model	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
SFT	97.7	93.3	93.9	100	80.5	85.7
Pre+SFT	98.6	95.6	95.4	100	81.1	87.6
Pre+SFT+RL	98.0	98.3	93.7	100	88.5	91.3

Ablation on Group Size. As shown in Tab. 6, while applying RL significantly elevates the PDMS over the baseline (87.6), the performance gain saturates as the group size increases. Specifically, the PDMS improves from 90.8 to 91.3 when scaling the group size from 4 to 8, but slightly drops to 91.2 at a group size of 12. This demonstrates that a group size of 8 is sufficient to achieve optimal planning capability without redundant computational overhead.

Table 8: **Ablation on high score thresholds. HST** means high score thresholds τ.

HST	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑	Refl. Ratio
0.90	98.0	98.2	93.1	100	87.9	90.2	14.9%
0.92	98.0	98.3	93.7	100	88.5	91.3	20.6%
0.94	98.2	98.0	94.2	100	88.1	91.2	29.1%
0.96	98.0	98.1	93.4	100	88.3	90.1	34.8%

Table 9: **Ablation study on the number of training data in RL.** *: Sampled based on score and variance. †: Random Sampled.

Num.	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
132k	96.2	97.3	93.4	100	88.5	87.7
15k†	96.1	96.6	89.8	100	86.2	86.2
15k*	98.0	98.3	93.7	100	88.5	91.3

Ablation on Training Stage. Tab. 7 validates the effectiveness of the proposed multi-stage training pipeline. Incorporating pretraining into the SFT baseline consistently improves generalization (PDMS +1.9). Further introducing reinforcement learning leads to substantial improvements in decision-making performance, with notable gains in DAC and EP. The full pipeline (Pre+SFT+RL) achieves the best overall performance, reaching a PDMS of 91.3 (+5.6 over SFT), thereby highlighting the critical role of RL in closed-loop execution.

Ablation on High Score Thresholds and Training Data in RL. Tab. 8 ablates the high score threshold (τ). The results indicate that $\tau = 0.92$ is the optimal operating point, yielding the highest PDMS of 91.3 while avoiding the computational overhead and performance degradation (90.1 PDMS) associated with excessive reflection ($\tau = 0.96$). Tab. 9 validates our score-and-variance-based sampling strategy during the RL stage. Fine-tuning with only 15k targeted informative samples (15k*) drastically outperforms not only the 15k random subset (86.2 PDMS) but also the full 132k dataset (87.7 PDMS), reaching 91.3 PDMS. This highlights that data quality and targeted exploration are far more crucial than sheer data volume for refining driving policies.

5 Conclusion

In this paper, we argue that existing reasoning paradigms for autonomous driving are weakly grounded and treat reflection as a static, post-hoc process. To address this, we propose IRR-Drive, a novel multimodal reflection framework that leverages predicted future semantic BEV representations as a structured space for plan verification. In addition, we enable adaptive reflection, allowing the model to dynamically adjust reasoning based on scene complexity through a reinforcement learning pipeline. Experiments on NAVSIM v1 and v2 demonstrate the effectiveness of IRR-Drive, showing that multimodal reasoning is complementary and that adaptive reflection balances inference efficiency and performance.

Limitations and Future Work. Despite its competitive performance and adaptive reasoning capability, IRR-Drive has several limitations. First, its current inference latency (1.70 s per frame on an H800 GPU) limits real-time deployment. Future work will explore model quantization, distillation, and accelerated decoding. Second, our experiments are conducted in short-horizon, non-reactive settings, which do not fully capture dynamic interactions among traffic participants or long-term error accumulation. Extending reflection to multi-step future prediction and interaction-aware replanning is an important direction. Finally, preliminary evaluations under adverse weather conditions suggest potential generalization challenges, motivating further investigation with more diverse scenarios and uncertainty-aware reflection.

Acknowledgement

This work is supported by National Key Research and Development Program of China(2024YFE0203100),Scientific Research Innovation Capability Support Project for Young Faculty (No.ZYGXQNJSKYCXNLZCXM-I28), National Natural Science Foundation of China (NSFC) under Grants No.62476293, Shenzhen Science and Technology Program(Grant No. QNXMA20250701093544048) and General Embodied AI Center of Sun Yat-sen University. This work is supported in part by The Major Key Project of PCL (Grant No. PCL2025A17).

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[2] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

[3] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning, 2024. URL `https://arxiv.org/abs/2402.13243`.

[4] Zisheng Chen, Chunwei Wang, Runhui Huang, Hongbin Xu, Xiuwei Chen, Jun Zhou, Jianhua Han, Hang Xu, and Xiaodan Liang. Semhitok: A unified image tokenizer via semantic-guided hierarchical codebook for multimodal understanding and generation. *arXiv preprint arXiv:2503.06764*, 2025.

[5] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *Pattern Analysis and Machine Intelligence (PAMI)*, 2023.

[6] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[7] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[8] Ignat Georgiev, Varun Giridhar, Nicklas Hansen, and Animesh Garg. Pwm: Policy learning with multi-task world models, 2025. URL `https://arxiv.org/abs/2407.02466`.

[9] Runhui Huang, Chunwei Wang, Junwei Yang, Guansong Lu, Yunlong Yuan, Jianhua Han, Lu Hou, Wei Zhang, Lanqing Hong, Hengshuang Zhao, et al. Illume+: Illuminating unified mllm with dual visual tokenization and diffusion refinement. *arXiv preprint arXiv:2504.01934*, 2025.

[10] Feiyang jia, Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, and Long Chen. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving, 2026. URL `https://arxiv.org/abs/2602.06521`.

[11] Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving, 2024. URL `https://arxiv.org/abs/2410.22313`.

[12] Jiaze Li, Jingyang Chen, Yuxun Qu, Shijie Xu, Zhenru Lin, Junyou Zhu, Boshen Xu, Wenhui Tan, Pei Fu, Jianzhong Ju, et al. Xiaomi mimo-vl-miloco technical report. *arXiv preprint arXiv:2512.17436*, 2025.

[13] Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M. Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation, 2025. URL `https://arxiv.org/abs/2503.12820`.

[14] Yanze Li, Wenhua Zhang, Kai Chen, Yanxin Liu, Pengxiang Li, Ruiyuan Gao, Lanqing Hong, Meng Tian, Xinhai Zhao, Zhenguo Li, et al. Automated evaluation of large vision-language models on self-driving corner cases. *arXiv preprint arXiv:2404.10595*, 2024.

[15] Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. Drivevla-w0: World models amplify data scaling law in autonomous driving, 2025. URL `https://arxiv.org/abs/2510.12796`.

[16] Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025.

[17] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving, 2025. URL `https://arxiv.org/abs/2411.15139`.

[18] Xueyi Liu, Zuodong Zhong, Yuxin Guo, Yun-Fu Liu, Zhiguo Su, Qichao Zhang, Junli Wang, Yinfeng Gao, Yupeng Zheng, Qiao Lin, Huiyong Chen, and Dongbin Zhao. Reasonplan: Unified scene prediction and decision reasoning for closed-loop autonomous driving, 2025. URL https://arxiv.org/abs/2505.20024.

[19] Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, Long Chen, Bing Wang, and Zhi xin Yang. Adathinkdrive: Adaptive thinking via reinforcement learning for autonomous driving, 2025. URL https://arxiv.org/abs/2509.13769.

[20] Yuechen Luo, Qimao Chen, Fang Li, Shaoqing Xu, Jaxin Liu, Ziying Song, Zhi xin Yang, and Fuxi Wen. Unleashing vla potentials in autonomous driving via explicit learning from failures, 2026. URL https://arxiv.org/abs/2603.01063.

[21] Yuechen Luo, Fang Li, Shaoqing Xu, Yang Ji, Zehan Zhang, Bing Wang, Yuannan Shen, Jianwei Cui, Long Chen, Guang Chen, et al. Last-vla: Thinking in latent spatio-temporal space for vision-language-action in autonomous driving. *arXiv preprint arXiv:2603.01928*, 2026.

[22] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pages 252–269. Springer, 2024.

[23] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[24] Zhenghao "Mark" Peng, Wenhao Ding, Yurong You, Yuxiao Chen, Wenjie Luo, Thomas Tian, Yulong Cao, Apoorva Sharma, Danfei Xu, Boris Ivanovic, Boyi Li, Bolei Zhou, Yan Wang, and Marco Pavone. Counterfactual vla: Self-reflective vision-language-action model with adaptive reasoning, 2025. URL https://arxiv.org/abs/2512.24426.

[25] 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. *arXiv preprint arXiv:2305.14836*, 2023.

[26] Jihao Qiu, Yuan Zhang, Xi Tang, Lingxi Xie, Tianren Ma, Pengyu Yan, David Doermann, Qixiang Ye, and Yunjie Tian. Artemis: Towards referential understanding in complex videos, 2024. URL https://arxiv.org/abs/2406.00258.

[27] Shuyao Shang, Yuntao Chen, Yuqi Wang, Yingyan Li, and Zhaoxiang Zhang. Drivedpo: Policy learning via safety dpo for end-to-end autonomous driving, 2025. URL https://arxiv.org/abs/2509.17940.

[28] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer, 2024.

[29] Yuehao Song, Shaoyu Chen, Hao Gao, Yifan Zhu, Weixiang Yue, Jialv Zou, Bo Jiang, Zihao Lu, Yu Wang, Qian Zhang, and Xinggang Wang. Senna-2: Aligning vlm and end-to-end driving policy for consistent decision making and planning, 2026. URL https://arxiv.org/abs/2603.11219.

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[31] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[32] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[33] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[34] Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *International Journal of Computer Vision*, 134(1):39, 2026.

[35] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[36] Ding Xinpeng, Han Jinahua, Xu Hang, Laing Xiaodan, Hang Xu, Zhang Wei, and Li Xiaomeng. Holistic autonomous driving understanding by bird's-eye-view injected multi-modal large models. 2024.

[37] Yifang Xu, Jiahao Cui, Feipeng Cai, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Neng Zhang, Yaoyi Li, Jia Cai, and Siyu Zhu. Wam-flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving. In *CVPR*, 2026.

[38] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024.

[39] Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning, 2025. URL `https://arxiv.org/abs/2506.06659`.

[40] Bowen Ye, Bin Zhang, and Hang Zhao. Dap: A discrete-token autoregressive planner for autonomous driving. *arXiv preprint arXiv:2511.13306*, 2025.

[41] Yuqi Ye, Zijian Zhang, Junhong Lin, Shangkun Sun, Changhao Peng, and Wei Gao. AutoDrive-Pi^3: Unified chain of perception-prediction-planning thought via reinforcement fine-tuning. *arXiv preprint arXiv:2603.28116*, 2026.

[42] Zhenlong Yuan, Chengxuan Qian, Jing Tang, Rui Chen, Zijian Song, Lei Sun, Xiangxiang Chu, Yujun Cai, Dapeng Zhang, and Shuo Li. AutoDrive-R^2: Incentivizing reasoning and self-reflection capacity of vla models in autonomous driving. *arXiv preprint arXiv:2509.01944*, 2025.

[43] Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, Xing Wei, and Ning Guo. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving, 2025. URL `https://arxiv.org/abs/2505.17685`.

[44] Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive diffusion world model for autonomous driving, 2025. URL `https://arxiv.org/abs/2506.24113`.

[45] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[46] 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, 2025. URL `https://arxiv.org/abs/2506.13757`.

[47] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Supplementary Material

A.1 Semantic Tokenizer

As shown in Fig. 5, we demonstrate how to train a semantic BEV tokenizer and how to use it to convert BEV maps into discrete special tokens compatible with the autoregressive VLM interface.

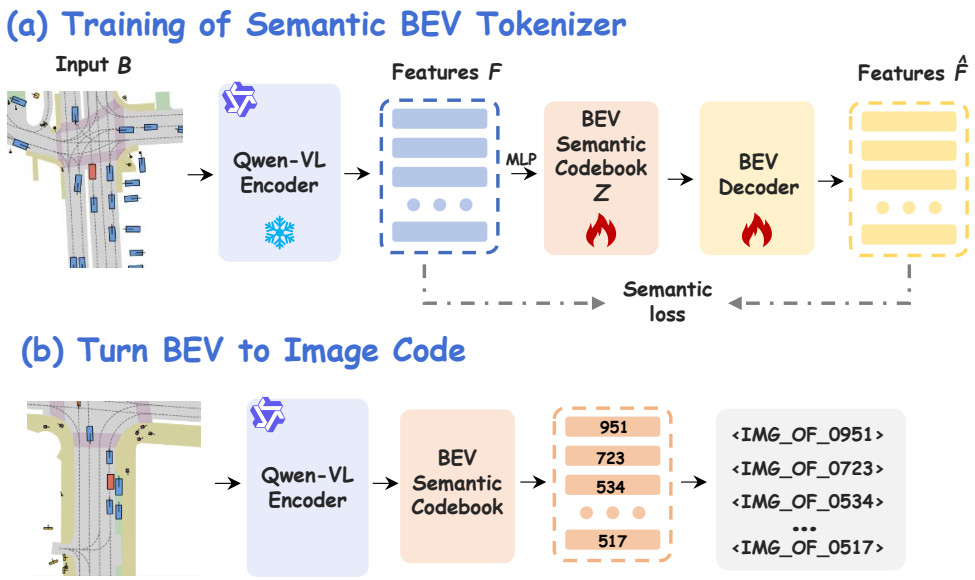


Figure 5: (a) Training framework of the semantic BEV tokenizer. A frozen semantic encoder processes the BEV representation to extract features, which are then discretized and decoded to reconstruct the continuous semantic features. (b) The trained semantic BEV tokenizer is used to convert BEV representations into discrete codes. To align with the autoregressive workflow of the VLM, these codes are further mapped into special tokens in the format <IMG_OF_XXXX>.

A.2 Adaptive Data Examples

We present examples of data from simple and challenging scenarios in Fig. 6 and 7, respectively.

A.3 Pseudocode for data construction and filtering

We present the pseudocode for data construction, as shown in Algs 1 and 2.

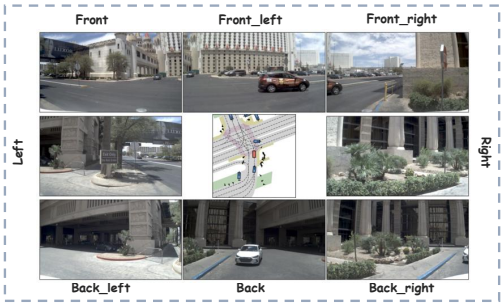
A.4 Detailed computation of the Concentric OBB-FDE Reward

We provides the formal mathematical definition of the concentric OBB-FDE reward (R_{obb}) introduced in the main text. The calculation is decoupled into two stages: computing the geometric spatial deviation and mapping it to a discrete reward score.

Spatial Deviation Calculation Let $\mathbf{P}_{gt} = (x_{gt}, y_{gt}, \theta_{gt})$ denote the ground truth pose of the ego-vehicle at the final trajectory timestep, and $\mathbf{P}_{res} = (x_{res}, y_{res})$ denote the predicted endpoint. We first calculate the global positional error and project it into the local coordinate frame of the ground truth pose:

$$\begin{bmatrix} x_{local} \\ y_{local} \end{bmatrix} = \begin{bmatrix} \cos\theta_{gt} & \sin\theta_{gt} \\ -\sin\theta_{gt} & \cos\theta_{gt} \end{bmatrix} \begin{bmatrix} x_{res} - x_{gt} \\ y_{res} - y_{gt} \end{bmatrix}. \tag{6}$$

To account for the asymmetric physical dimensions of the ego-vehicle, we define the front overhang as l_{front}, the rear overhang as l_{back}, and the half-width as w_{half}. The spatial deviation is quantified by the minimum concentric scaling factor k required for the ego-vehicle's oriented bounding box to fully enclose the predicted endpoint.



Simple Scene Reasoning

Initial Intention

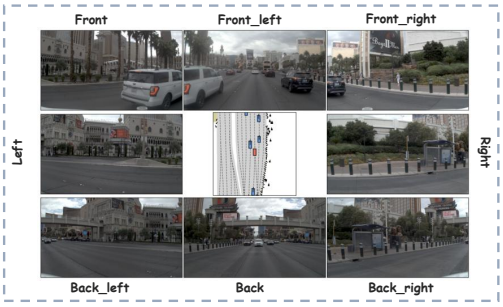
x-direction: 0.0s~1.0s: accelerate to 0.7, 1.0s~3.0s: accelerate to 2.9, 3.0s~4.0s: decelerate to 2.7; y-direction: 0.0s~1.5s: no lateral movement, 1.5s~4.0s: accelerating left to -3.3;

Reflect

This trajectory does not need adjustment.

Final Trajectory

(+0.05,+0.00,+0.00)(+0.39,-0.01,-0.01)(+1.13,-0.02,-0.05)(+2.17,-0.10,-0.12)(+3.47,-0.33,-0.27)(+4.94,-0.90,-0.51)(+6.42,-1.99,-0.78)(+7.78,-3.63,-1.02)



Simple Scene Reasoning

Initial Intention

x-direction: 0.0s~4.0s: decelerate to 3.4; y-direction: 0.0s~4.0s: no lateral movement;

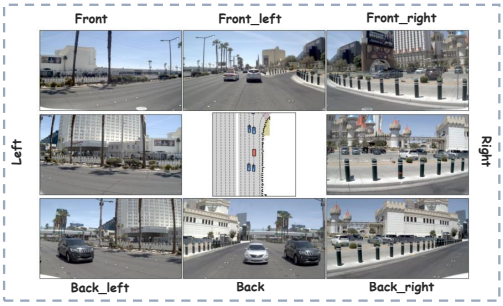
Reflect

This trajectory does not need adjustment.

Final Trajectory

(+4.14,+0.01,+0.00)(+8.05,+0.03,+0.01)(+11.65,+0.05,+0.00)(+14.86,+0.06,+0.00)(+17.68,+0.06,+0.00)(+20.08,+0.07,+0.00)(+22.11,+0.05,+0.00)(+23.82,+0.04,+0.00)

Figure 6: Illustrative examples of simple scenes.



Challenging Scene Reasoning

Initial Intention

x-direction: 0.0s~1.0s: decelerate to 3.5, 1.0s~2.0s: decelerate to 1.5, 2.0s~3.0s: decelerate to 0.0, 3.0s~4.0s: decelerate to -0.5; y-direction: 0.0s~0.5s: drifting left at 0.1, 0.5s~2.5s: no lateral movement, 2.5s~4.0s: drifting right at -0.1;

Reflect

1.Intent-conditioned BEV Forecasting

<IMG_OF_1008><IMG_OF_0511><IMG_OF_0367><IMG_OF_0890><IMG_OF_0474><IMG_OF_0748><IMG_OF_0817><IMG_OF_0306><IMG_OF_0466><IMG_OF_0256><IMG_OF_0033><IMG_OF_0952><IMG_OF_0952><IMG_OF_0306><IMG_OF_0730><IMG_OF_0227><IMG_OF_0179>...

2.BEV Grounded reasoning

Global Description

The ego vehicle is positioned at the center of a multi-lane road, surrounded by several other vehicles ahead and to the sides. A stop line is visible ahead, indicating an upcoming intersection or controlled crossing. The road includes clearly defined lanes and a curved section to the right, bordered by a pedestrian walkway. The ego is aligned with the lane centerline and appears to be approaching the stop line with safe spacing from surrounding traffic.

Safety Assessment

The ego vehicle maintains safe margins from all nearby vehicles and is fully within the drivable area. Its current progress reflects moderate efficiency, suggesting room for smoother acceleration without compromising safety. No immediate collision risk is present.

Trajectory suggestions

direction: keep, speed: accelerate

Refine

1.Refined Intent

x-direction: 0.0s~4.0s: decelerate to 2.9; y-direction: 0.0s~4.0s: no lateral movement

2.Final Trajectory

(+2.36,-0.01,+0.00)(+4.61,-0.01,+0.00)(+6.80,-0.02,+0.00)(+8.92,-0.01,+0.00)(+10.94,+0.01,+0.01)(+12.81,+0.03,+0.01)(+14.50,+0.03,+0.01)(+15.94,+0.05,+0.01)



Challenging Scene Reasoning

Initial Intention

x-direction: 0.0s~3.0s: accelerating to 4.8, 3.0s~4.0s: decelerating to 4.7; y-direction: 0.0s~2.0s: accelerating left to 1.5, 2.0s~3.5s: maintaining left drift at 1.9, 3.5s~4.0s: decelerating left to 1.6;

Reflect

1.Intent-conditioned BEV Forecasting

<IMG_OF_0847><IMG_OF_0777><IMG_OF_1007><IMG_OF_0711><IMG_OF_0073><IMG_OF_0620><IMG_OF_0519><IMG_OF_0847><IMG_OF_0649><IMG_OF_0958><IMG_OF_0630><IMG_OF_0493><IMG_OF_0011><IMG_OF_0428><IMG_OF_0227><IMG_OF_0506><IMG_OF_0543><IMG_OF_0495>...

2.BEV Grounded reasoning

Global Description

The ego vehicle is positioned centrally in a multi-lane road, flanked by multiple vehicles ahead and to both sides. The road includes clear lane markings and is bordered by walkways on either side. No stop lines or intersections are visible ahead. The ego is aligned with the centerline of its lane, surrounded by other vehicles maintaining consistent spacing.

Safety Assessment

The ego vehicle is operating safely with no immediate collision risk and is fully within the drivable area. The trajectory is efficient, showing good progress without unnecessary delays or deviations. Surrounding vehicles maintain safe margins, and the environment remains predictable and clear.

Trajectory suggestions

direction: turn right, speed: accelerate

Refine

1.Refined Intent

x-direction: 0.0s~0.5s: accelerate to 4.2, 0.5s~4.0s: maintain speed at 5.4; y-direction: 0.0s~0.5s: accelerating left to 0.3, 0.5s~4.0s: maintain left drift at 0.7;

2.Final Trajectory

(+2.06,+0.09,+0.08)(+4.23,+0.29,+0.12)(+6.53,+0.55,+0.14)(+8.96,+0.86,+0.14)(+11.52,+1.20,+0.14)(+14.20,+1.56,+0.14)(+16.97,+1.95,+0.14)(+19.86,+2.41,+0.15)

Figure 7: Illustrative examples of challenging scenes.

Instead of employing a piecewise function, this factor can be elegantly computed as the maximum required expansion ratio across both the longitudinal and lateral axes in a single unified expression:

$$k = \max\left(\frac{x_{local}}{l_{front}}, \frac{-x_{local}}{l_{back}}, \frac{|y_{local}|}{w_{half}}\right). \tag{7}$$

This formulation naturally handles the longitudinal asymmetry, as the $\max()$ operator inherently filters out the irrelevant negative term depending on whether the prediction falls in front of or behind the vehicle's geometric center.

Stepwise Reward Calculation Once the spatial scaling factor k is obtained, it is mapped to a discrete reward score $R_{obb} \in [0, 1]$. We employ a stepwise decaying function based on predefined scale thresholds, representing concentrically expanded bounding boxes. The discrete reward is assigned as follows:

$$R_{obb}(k) = \begin{cases} 1.0, & \text{if } k \leq 1.0 \quad \text{(Within physical ego box)} \\ 0.9, & \text{if } 1.0 < k \leq 1.25 \\ 0.8, & \text{if } 1.25 < k \leq 1.5 \\ 0.6, & \text{if } 1.5 < k \leq 2.0 \\ 0.4, & \text{if } 2.0 < k \leq 3.0 \\ 0.2, & \text{if } 3.0 < k \leq 5.0 \\ 0.0, & \text{otherwise (Severe deviation)} \end{cases}. \tag{8}$$

This step-down mechanism avoids zero-sum binary penalties and provides a smooth, progressive learning signal during the reinforcement learning process while strictly adhering to the vehicle's physical collision boundaries.

A.5 Detail of EPDMS

The EPDMS metric is calculated as follows:

$$\text{EPDMS} = \text{NC} \cdot \text{DAC} \cdot \text{DDC} \cdot \text{TLC} \cdot \left(\frac{5\text{EP} + 2\text{LK} + 2\text{HC} + 5\text{TTC} + 2\text{EC}}{16}\right). \tag{9}$$

where NC, DAC, Driving Direction Compliance (DDC), Traffic Light Compliance (TLC), EP, TTC, Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC) jointly provide a comprehensive evaluation of safety, comfort, and driving performance.

A.6 More Training Details

The model is trained in three stages, as described in Sec. 3.3 and 3.4. In the first stage, we conduct SFT on pre-training data for 5 epochs with a learning rate of 2×10^{-5} and batch size 128. The second stage fine-tunes the model on supervised trajectory and reflection data (D_{traj}, D_{refl}) for 5 epochs using a learning rate of 1×10^{-5}, batch size 128. In the third stage, we apply reinforcement learning for 3 epochs on $\mathcal{D}^{rl}_{refl}$, with a learning rate of 2×10^{-6}, using a rollout size of 8 and a batch size of 128. In addition, in the Concentric OBB-FDE Reward, the front overhang (l_{front}), rear overhang (l_{back}), and half-width (w_{half}) are set to 2.0, 1.2, and 1.0, respectively. The default high score threshold τ for the Adaptive Reflection Reward is 0.92. All experiments are conducted on the H800.

A.7 More Ablation on CoT content.

Because IRR-Drive selectively triggers reflection in challenging scenarios ($\sim$20.6% of `navtest`), evaluating across the full benchmark dilutes the reflection gains. We therefore evaluate reflection modalities on the Text-Refl subset (2379 samples) in Tab. 10. Relying solely on either *BEV-only* or *Text-only* yields sub-optimal performance. In contrast, our combined **Text+BEV** achieves **86.5** PDMS and **85.6** EPDMS, outperforming *Text-only* by 1.7/2.8 and *BEV-only* by 2.3/3.7, which confirms the strong complementarity between textual reasoning and spatial BEV reflection.

A.8 Broader impacts

Autonomous driving is inherently safety-critical, a concern that becomes even more significant when language guidance is incorporated into VLA models. Such integration requires strong defenses

Table 10: **Ablation on reflection modalities on the Text-Refl Subset (2379 samples)**.

Method	V1 PDMS ↑	V2 EPDMS ↑
BEV only	84.2	81.9
Text only	84.8	82.8
Text+BEV (Ours)	**86.5**	**85.6**

against adversarial attacks, as well as mechanisms to detect and filter unsafe or malicious human instructions in advance. To reduce potential security risks, secure in-vehicle communication channels should be established, and model deployment should adopt a gated-release update strategy instead of relying on online continual reinforcement fine-tuning directly on individual vehicles.

A.9 LLM usage

As described in Sec 3.2, we use open-source LLMs and VLMs to construct the dataset. In addition, we utilize closed-source models (GPT, Gemini) to assist in paper writing.

Algorithm 1 Data Preparation for Reflection-Augmented Training

Input: driving understanding data $\mathcal{D}_{under}$, NAVSIM, trajectory data $\mathcal{D}_{traj}$, base planner $\mathcal{M}$, VLM $\mathcal{V}$, LLM $\mathcal{L}$, rollout number $K = 8$, temperature $\tau_{\text{temp}} = 1$, hard-sample number $N_{hard} = 4000$
Output: pretraining data $\mathcal{D}_{pre}$, reflection data $\mathcal{D}_{refl}$, RL data $\mathcal{D}^{rl}_{refl}$
Construct $\mathcal{D}_{pre}$ from $\mathcal{D}_{under}$, BEV reconstruction data, and BEV prediction data
Train $\mathcal{M}$ on $\mathcal{D}_{traj}$
for each sample $x \in \mathcal{D}_{traj}$ **do**
    Predict trajectory $\hat{\mathbf{y}} \leftarrow \mathcal{M}(x)$
    Compute PDMS score s with NAVSIM
end for
Rank all samples by PDMS score
Define the bottom N_{hard} samples as $\mathcal{D}^{+}_{refl}$ and the remaining ones as $\mathcal{D}^{-}_{refl}$
for each sample $x \in \mathcal{D}_{traj}$ **do**
    Obtain ground-truth trajectory $\mathbf{y}^*$
    Generate trajectory intention $\mathbf{z} \leftarrow \mathcal{L}(\mathbf{y}^*)$
    if $x \in \mathcal{D}^{+}_{refl}$ **then**
        Generate future BEV $\mathbf{B}^f$ with NAVSIM conditioned on $\hat{\mathbf{y}}$
        Generate textual reflection $r \leftarrow \mathcal{V}(\hat{\mathbf{y}}, \mathbf{B}^f, s)$
        Add reflective sample $(x, \mathbf{z}, \mathbf{B}^f, r, \mathbf{y}^*)$ to $\mathcal{D}_{refl}$
    else
        Add non-reflective sample $(x, \mathbf{z}, \mathbf{y}^*)$ to $\mathcal{D}_{refl}$
    end if
end for
Train an SFT model on $\mathcal{D}_{traj} \cup \mathcal{D}_{refl}$
for each sample $d \in \mathcal{D}_{refl}$ **do**
    Perform K rollouts with temperature τ_{temp}
    Compute rollout PDMS scores
    Compute mean score μ_d and variance σ_d^2
end for
Select samples with low mean score and high variance to form $\mathcal{D}^{rl}_{refl}$
return $\mathcal{D}_{pre}, \mathcal{D}_{refl}, \mathcal{D}^{rl}_{refl}$

Algorithm 2 Informative Sample Selection for RL Training

Input: Reflection dataset $\mathcal{D}_{refl}$, SFT model $\mathcal{M}_{sft}$, rollout number $K = 8$, temperature $\tau = 1$
Output: RL training subset $\mathcal{D}^{rl}_{refl}$
Initialize $\mathcal{D}^{rl}_{refl} \leftarrow \emptyset$
for each sample $d \in \mathcal{D}_{refl}$ **do**
    Sample K rollout trajectories $\{\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_K\}$ from $\mathcal{M}_{sft}$ with temperature τ
    Evaluate PDMS scores $\{s_1, \ldots, s_K\}$
    Compute mean score $\mu_d \leftarrow \frac{1}{K}\sum_{k=1}^{K} s_k$
    Compute variance $\sigma_d^2 \leftarrow \frac{1}{K}\sum_{k=1}^{K}(s_k - \mu_d)^2$
end for
Select samples with low μ_d and high σ_d^2
Add selected samples into $\mathcal{D}^{rl}_{refl}$
return $\mathcal{D}^{rl}_{refl}$

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: We mention the core contribution points of our methodology in the abstract( ): a new paradigm of multimodal reflection and adaptive reflective design

   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: We discuss limitations in Sec. 5.

   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: [No]

Justification: The paper does not involve the theoretical proof section.

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: We presented detailed experimental settings in Sec. 4 and A.6, and all experiments are developed based on open-source models on open-source datasets.

   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: [Yes]

Justification: We will release our code and dataset to facilitate community research.

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: All experimental data are divided with reference to the official open source settings.

   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 paper reports benchmark results under the official NAVSIM evaluation protocol, but does not include error bars, confidence intervals, or statistical significance tests. Running multiple full training seeds is computationally expensive due to the multi-stage SFT and RL pipeline. We instead support the main claims with extensive ablations on training stages, reward design, reflection content, and key hyperparameters.

   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 provides all necessary details on computational resources, including GPU type, memory usage, and number of epochs, in Sec. 4 and Sec. A.6.

   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: Our work is fully aligned with the NeurIPS Code of Ethics.

   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 includes a discussion of potential societal impacts in the supplementary material.

    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: [N/A]

    Justification: The paper poses no foreseeable risks.

    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 participants in the work on this thesis have signed. And we respect the licenses of open source models and open source data providers.

    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: [No]

    Justification: This paper does not introduce any new assets.

    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: [No]

    Justification: his paper does not involve research on crowdsourcing experiments and research with human subjects.

    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: [No]

    Justification: This paper does not involve research on human subjects.

    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: We use open-source LLMs and VLMs to construct training data, as described in Sec. 3.2 and Sec. A.9.

    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.