TOWARDS SELF-IMPROVING AUTONOMOUS DRIVING THROUGH IMAGINED INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models have shown strong potential for autonomous driving, yet their learning remains largely confined to fixed offline demonstrations. Such offline training provides little supervision from the states and consequences induced by the policy's own decisions, limiting continual self-improvement. We present **InteractVLA**, a framework for self-improving autonomous driving through *imagined interaction*. Rather than using predicted futures only for reasoning or action evaluation, InteractVLA converts action-conditioned future transitions into multi-step policy-induced experience for policy optimization. To enable efficient imagined interaction, we introduce an *Asymmetric Action-Transition Dual Transformer* that jointly performs action generation and transition modeling within a single forward pass, while compact *transition tokens* capture action-induced changes without reconstructing complete future observations. We further propose *interaction-aware feedback* to maintain reliable supervision as imagined states deviate from logged traffic, together with an interaction-specific *return-to-go* objective for temporal credit assignment under GRPO. Experiments on NAVSIM and Bench2Drive demonstrate strong planning performance and consistent gains under successive interaction. InteractVLA achieves 93.2 PDMS on NAVSIM and a 91.13 Driving Score on Bench2Drive, and reaches 88.3 PDMS under 5-second continuation on NAVSIM. These results demonstrate the effectiveness of imagined interaction for self-improving autonomous driving.

1 INTRODUCTION

Existing Vision-Language-Action (VLA) models (Wang et al., 2025; Zhou et al., 2025b; 2026b) for autonomous driving are primarily trained on fixed offline demonstrations. Although such data provides strong supervision for imitation learning, the training experience does not change with the policy itself: as the policy evolves, it receives little new supervision from the states and outcomes induced by its own decisions. A self-improving driving policy should instead learn from such policy-induced experience. By interacting with the environment, the policy can explore alternative actions, observe their consequences, and use the resulting feedback to improve subsequent behavior (Kaelbling et al., 1996; Ross et al., 2011). In practice, however, collecting this experience in physical traffic is difficult to scale, since repeated exploration is costly and imperfect actions may introduce safety risks. This raises a fundamental question: *how can driving models acquire sufficient interaction experience for self-improvement without extensive real-world trial and error?*

World models offer a promising route toward self-improvement by predicting plausible future states. However, existing approaches remain largely *decision-centric*: imagined futures are used to improve the current decision rather than to create new interaction experience for updating the policy. In Fig. 1(a), future predictions serve as auxiliary visual reasoning for the current observation (Wang et al., 2026a; Shang et al., 2026). In Fig. 1(b), they evaluate or select among candidate trajectories generated from the same observed state (Liu et al., 2026a; Yan et al., 2026a). More recent methods jointly model future scenes and driving actions (Fig. 1(c)) (Li et al., 2026b; Agarwal et al., 2026; Zhan et al., 2026), but the predicted future primarily remains part of the current prediction process. Across these formulations, the imagined state is not explicitly turned into the next policy-induced training state, evaluated for feedback, and used to update the policy. Consequently, the learning loop between *action, consequence, feedback, and policy update* remains open.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

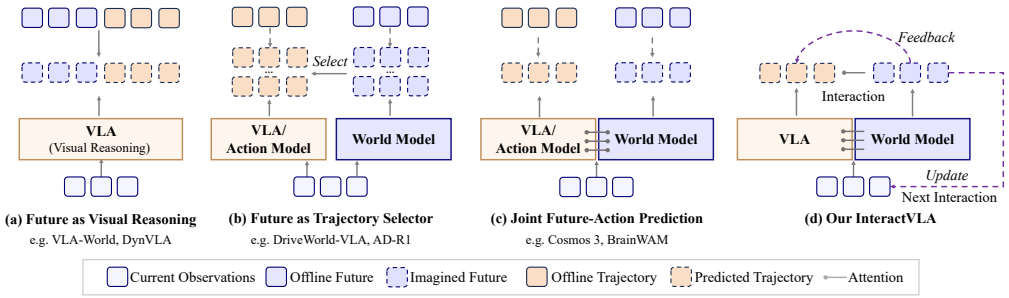

Figure 1: **From decision-centric world modeling to imagined interaction.** Existing approaches primarily use imagined futures within the current planning step, including (a) visual reasoning, (b) candidate trajectory selection, and (c) joint modeling of future scenes and driving actions. (d) **InteractVLA** instead learns through imagined interactions for self-improvement.

Closing this loop appears straightforward: one could recursively feed the predicted future state back to the VLA, turning the joint action-future prediction in Fig. 1(c) into the multi-step interaction illustrated in Fig. 1(d). At each step, the VLA predicts an action, the world model predicts its consequence, and the resulting state becomes the input to the next planning step. Such rollouts expose the policy to states induced by its own decisions and therefore provide new experience for self-improvement. Making this idea practical, however, is non-trivial. First, repeatedly alternating between action generation and future prediction makes multi-step interaction computationally expensive. Second, as imagined states progressively depart from logged traffic, conventional replay-based evaluation (Yan et al., 2026a; Shang et al., 2026) becomes increasingly unreliable. Third, optimizing individual interaction steps independently ignores how early decisions affect later states and downstream planning outcomes. Thus, the challenge is not merely to predict plausible futures, but to transform policy-induced futures into *efficient, reliable, and temporally coherent learning experience*.

To this end, we introduce **InteractVLA**, a framework for self-improving autonomous driving through *imagined interaction*. Unlike the decision-centric formulations in Fig. 1(a-c), InteractVLA explicitly closes the interaction-learning loop in Fig. 1(d): the VLA produces an action, the world model predicts the corresponding state transition, and the predicted state is fed back to the policy to continue interaction. The resulting multi-step trajectory is then evaluated to obtain feedback, which is propagated across interaction steps and used to update the VLA. The updated policy can subsequently generate new interactions, allowing its own decisions and consequences to continually expand the experience used for optimization.

To make this process efficient, we introduce an *Asymmetric Action-Transition Dual Transformer* with a VLA and a World Model pathway, which performs action generation and action-conditioned future prediction within a single forward pass. Instead of reconstructing complete future observations, the world-modeling pathway represents action-induced changes between consecutive observations using compact *transition tokens*, substantially reducing the cost of imagined rollouts. The world-modeling pathway is used only during training to generate imagined interactions and is removed at inference. To maintain reliable learning signals as imagined states depart from logged traffic, we propose *interaction-aware feedback* that adaptively combines replay-based evaluation with learned estimates according to visual correspondence. Third, we introduce an interaction-specific *return-to-go* to propagate feedback across interaction steps and incorporate it into Group-Relative Policy Optimization (GRPO) (Shao et al., 2024) for self-improvement from multi-step imagined experience.

We evaluate InteractVLA on NAVSIM (Dauner et al., 2024) and Bench2Drive (Jia et al., 2024), together with continuation settings that assess planning over successive interaction steps. InteractVLA achieves 93.2 PDMS under the standard 4-second NAVSIM setting and a Driving Score of 91.13 on Bench2Drive. Under 5-second continuation on NAVSIM, it further achieves 88.3 PDMS, outperforming the previous state-of-the-art DynVLA by 10.5% relatively. These results demonstrate strong standard planning performance while validating the effectiveness of imagined interaction for continued planning under policy-induced states. Our contributions are summarized as follows:

- We formulate *imagined interaction* as a learning paradigm for self-improving autonomous driving, where a VLA generates experience from its own decisions through action-conditioned world-model rollouts and learns from the resulting consequences.

- We propose **InteractVLA**, which enables efficient and reliable imagined interaction through an *Asymmetric Action-Transition Dual Transformer*, *interaction-aware feedback*, and interaction-specific *return-to-go* optimization.
- Extensive experiments demonstrate strong conventional closed-loop planning performance and substantial gains across successive imagined interaction steps, validating the effectiveness of the proposed learning paradigm.

2 Related Work

VLA for Autonomous Driving. Vision-language models have recently been extended from driving scene understanding to trajectory planning, leveraging large-scale multimodal knowledge for complex driving decisions (Hwang et al., 2024). DriveVLM (Tian et al., 2024) and OmniDrive (Wang et al., 2025) integrate visual reasoning with planning, while OpenDriveVLA (Zhou et al., 2025a) further unifies multimodal perception and action generation within VLA architectures. Recent studies increasingly introduce reinforcement learning to strengthen planning and reasoning: AutoVLA (Zhou et al., 2025b) applies GRPO-based reinforcement fine-tuning, Drive-R1 (Li et al., 2026c) aligns reasoning with trajectory planning through task-specific rewards, and MindDrive (Sun et al., 2025) explores online reinforcement learning for driving VLAs. They primarily optimize policies from offline supervision or predefined rewards. Instead, InteractVLA constructs the learning environment itself through world modeling, enabling the policy to learn from multi-step imagined interaction.

World Models in Autonomous Driving. World models have been increasingly explored for predictive modeling in autonomous driving. Early works focus on controllable future generation in image, occupancy, or latent spaces (Hu et al., 2023a; Wang et al., 2024a;b; Gao et al., 2024; Zheng et al., 2024; Li et al., 2025a). More recent studies incorporate future prediction into decision-making, including trajectory evaluation and selection (Li et al., 2025b; Zheng et al., 2025), joint future-action modeling (Chen et al., 2025; Zhou et al., 2026a), and VLA training or reasoning (Li et al., 2026b; Shang et al., 2026; Liu et al., 2026a;d; Wang et al., 2026a). Despite increasingly tight coupling with planning, predicted futures are primarily used as supervision, reasoning context, or decision-time evaluation. In contrast, **InteractVLA** treats the world model as an *interaction engine* for policy learning, enabling imagined interaction and optimization from feedback across successive decisions.

Policy Learning through Interaction and Imagination. Learning policies from imagined experience has been extensively studied in model-based reinforcement learning. PlaNet (Hafner et al., 2019b) performs planning with learned latent dynamics, while the Dreamer family (Hafner et al., 2019a; 2023) further optimizes policies from imagined rollouts generated by a learned world model. In autonomous driving, recent VLA approaches explore reinforcement learning through trajectory-level rewards or closed-loop interaction (Zhou et al., 2025b; Li et al., 2026c; Sun et al., 2025), while IRL-VLA (Jiang et al., 2025) introduces a reward world model for efficient policy optimization. Unlike these methods, InteractVLA targets multi-step interaction between a driving VLA and a visual world model, where successive imagined observations induce new decisions and feedback across the interaction is explicitly incorporated into policy optimization for self-improvement.

3 Methodology

3.1 Overview

We present **InteractVLA**, a framework for self-improving autonomous driving. InteractVLA augments a driving VLA with a lightweight transition pathway that models action-conditioned scene evolution during training. An *Asymmetric Action-Transition Dual Transformer* couples action generation and transition modeling within a shared forward pass while preserving an independent VLA pathway. Training proceeds in three stages. We first initialize action and transition prediction on logged driving data, then learn trajectory feedback beyond the logged context, and finally optimize the VLA using model-generated experience.

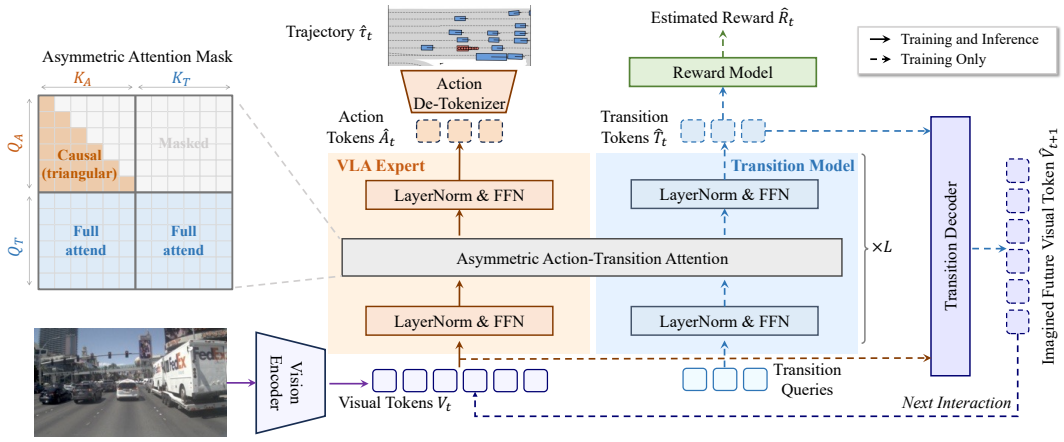

Figure 2: **Asymmetric Action-Transition Dual Transformer.** The VLA predicts action tokens, which are decoded into a continuous driving trajectory. Transition queries attend to action representations and predict transition tokens describing the corresponding scene changes. A transition decoder maps these tokens to future visual tokens, while the Reward Model provides trajectory feedback during training. Dashed components are removed at inference.

3.2 Asymmetric Action-Transition Dual Transformer

Modeling the consequence of a driving action typically requires a VLA for action prediction followed by a separate world model for future prediction. Such sequential execution introduces substantial overhead during training. We instead integrate the two processes into an *Asymmetric Action-Transition Dual Transformer*, as illustrated in Fig. 2. It consists of an *action pathway*, instantiated by the driving VLA, and a lightweight *transition pathway* that models the self-induced scene changes.

Action and Transition Representations. Given a driving observation O_t, the vision encoder extracts visual tokens $\mathbf{V}_t$. The VLA autoregressively predicts action tokens

$$\hat{A}_t \sim \pi_\theta(\cdot \mid \mathbf{V}_t), \tag{1}$$

which are converted by an action de-tokenizer $\mathcal{D}_A$ into a continuous driving trajectory,

$$\hat{\tau}_t = \mathcal{D}_A(\hat{A}_t). \tag{2}$$

Throughout the paper, A_t denotes the discrete policy output and τ_t its decoded trajectory.

Instead of predicting complete future observations, we represent scene changes using compact discrete *transition tokens* following Shang et al. (2026) (see Appendix C.1 for more details). A transition tokenizer is pretrained on consecutive driving observations and kept fixed thereafter. Conditioned on the representations produced during action generation, the transition pathway predicts transition tokens $\hat{T}_t$ describing the corresponding change of the visual scene.

Asymmetric Action-Transition Attention. Action and transition prediction have a directional dependency: transition prediction should condition on the generated action, whereas action generation should remain independent of transition representations. We enforce this dependency using

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{\text{causal}} & -\infty \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{3}$$

where rows and columns correspond to the action and transition streams. The action stream follows standard causal attention and cannot attend to the transition stream, while transition queries fully attend to the action representations. The two pathways share attention computation but retain independent normalization and feed-forward layers. This design allows transition prediction to reuse VLA representations without introducing additional dependencies into action generation.

Transition Decoding. The transition pathway predicts compact scene changes rather than explicit future observations. A fixed transition decoder $\mathcal{D}_T$ maps the predicted transition tokens to the corresponding future visual representation,

$$\tilde{\mathbf{V}}_{t+1} = \mathcal{D}_T\left(\mathbf{V}_t, \hat{T}_t\right). \tag{4}$$

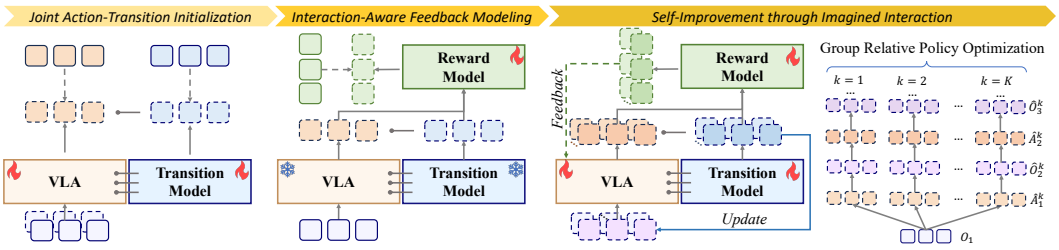

Figure 3: **Three-stage training pipeline of InteractVLA.** Stage I jointly initializes action and transition prediction. Stage II learns interaction-aware trajectory feedback. Stage III performs policy optimization from multi-step imagined interactions.

The decoder directly produces visual tokens instead of reconstructing pixel-space images, avoiding an additional image-generation and vision-encoding process. Since action generation does not depend on the transition pathway, all transition-related components can be removed at inference.

3.3 Three-Stage Training

3.3.1 Stage I: Joint Action-Transition Initialization

As shown in Fig. 3, we first initialize the action and transition pathways on logged driving data. Given logged visual tokens $\mathbf{V}_t^{\log}$, the action pathway is supervised with expert action tokens A_t^*, while the transition pathway predicts the corresponding transition-token target T_t^*.

The VLA is later exposed to visual representations produced by the transition decoder, whose distribution differs from that of visual tokens extracted from real observations. We therefore additionally include reconstructed contexts during initialization. Given the transition target between two logged observations, we reconstruct

$$\tilde{\mathbf{V}}_{t+1} = \mathcal{D}_T\left(\mathbf{V}_t^{\log}, T_t^*\right), \tag{5}$$

and supervise the VLA to predict the corresponding expert action from $\tilde{\mathbf{V}}_{t+1}$. These samples improve robustness to transition-decoder artifacts while preserving supervision on the logged distribution.

The Stage-I objective is

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_A^{\text{real}} + \lambda_T \mathcal{L}_T^{\text{real}} + \lambda_R \mathcal{L}_A^{\text{rec}}, \tag{6}$$

where $\mathcal{L}_A$ and $\mathcal{L}_T$ denote token-level action and transition prediction losses. We initialize $\lambda_R = 0$ and gradually increase it during training. Reconstructed contexts are generated without gradient propagation, and transition supervision is applied only to logged observation pairs.

3.3.2 Stage II: Interaction-Aware Feedback Modeling

Replay-based trajectory evaluation relies on logged future traffic and is most reliable when the evaluated context remains close to the recorded one. Once the visual context deviates from logged traffic, the recorded future may no longer provide an accurate reference for evaluation. We therefore introduce a *Reward Model* to estimate traffic-dependent trajectory quality from the generated context.

Following prior work (Shang et al., 2026; Zhou et al., 2025b; Liu et al., 2026a; Dauner et al., 2024), we use PDMS as the feedback signal. We retain the replay-based estimates of drivable-area compliance, ego progress, and comfort, and adapt the NC and TTC components using transition-based predictions.

Rather than replacing replay evaluation entirely, we combine the two estimates according to their visual correspondence. Let $s_t \in [0, 1]$ denote the similarity between an imagined visual context $\tilde{\mathbf{V}}_t$ and its logged counterpart $\mathbf{V}_t^{\log}$. For each traffic-dependent component $m \in \{\text{NC}, \text{TTC}\}$, we compute

$$\tilde{q}_t^m = s_t q_{t,\text{rep}}^m + (1 - s_t) q_{t,\text{rm}}^m, \tag{7}$$

where $q_{t,\text{rep}}^m$ and $q_{t,\text{rm}}^m$ denote the replay- and Reward-Model estimates (Fig. 3), respectively. Higher correspondence therefore assigns greater weight to replay evaluation, while the Reward Model becomes increasingly important as the generated context departs from logged traffic.

The fused NC and TTC estimates are combined with the directly evaluated PDMS components to produce the trajectory reward. The Reward Model is trained with replay-derived component scores while the action-transition model remains fixed. Please refer to Appendix C.2 for more details.

3.3.3 Stage III: Self-Improvement through Imagined Interaction

We finally use the learned transition and reward models to generate training experience for the VLA. We term the resulting closed-loop process *multi-step imagined interaction*.

Multi-Step Imagined Interaction. Each interaction starts from a logged visual context, $\hat{\mathbf{V}}_1 = \mathbf{V}_1^{\log}$. At step t, the VLA samples action tokens

$$\hat{A}_t \sim \pi_\theta \left(\cdot \mid \hat{\mathbf{V}}_t \right), \tag{8}$$

which are decoded into $\hat{\tau}_t = \mathcal{D}_A(\hat{A}_t)$. The transition pathway simultaneously predicts the corresponding transition tokens $\hat{T}_t$. Instead of advancing along the entire trajectory, we execute only an initial interval Δ. The corresponding transition tokens $\hat{T}_t^{\Delta}$ are decoded into the next visual context,

$$\hat{\mathbf{V}}_{t+1} = \mathcal{D}_T \left(\hat{\mathbf{V}}_t, \hat{T}_t^{\Delta} \right). \tag{9}$$

The generated visual tokens directly condition the next VLA decision. Repeating this procedure yields a sequence of actions, trajectories, and visual contexts induced by the current policy. For step t, the previously executed trajectory prefixes and current prediction form the continuation trajectory

$$\bar{\tau}_t = \left(\bigoplus_{j=1}^{t-1} \hat{\tau}_j^{[0:\Delta]} \right) \oplus \hat{\tau}_t, \tag{10}$$

where $\oplus$ expresses all trajectory segments in the coordinate frame of the initial observation. The interaction-aware feedback introduced in Stage II evaluates $\bar{\tau}_t$ under the current visual context, producing the step-level reward R_t.

Interaction Return. Optimizing each step independently ignores how an early action changes the contexts and planning encountered later. We therefore define an interaction-specific return-to-go

$$G_t = \sum_{j=t}^{T} w_{j-t} R_j, \tag{11}$$

where w_{j-t} controls the contribution of future interaction steps. Each action is thus optimized according to both its immediate trajectory quality and the downstream outcomes induced by its consequences. For each initial context, we sample K imagined interactions that share the same starting observation but evolve differently according to the sampled action tokens. The interaction returns are used to optimize the VLA with GRPO (Shao et al., 2024), as shown in Fig. 3. We further employ a horizon curriculum that progressively increases the interaction length. The complete GRPO objective and optimization details are provided in Appendix C.3.

Joint Updating. During Stage III, the VLA is optimized from imagined-interaction returns, while logged driving data continue to supervise transition prediction and reward estimation. This keeps both training-time components grounded in observed traffic as the policy evolves.

Notably, after training, the transition pathway, transition decoder, and Reward Model are discarded. Inference therefore retains only the vision encoder, VLA, and action de-tokenizer, introducing no additional transition-model or reward-model overhead.

4 Experiments

4.1 Experimental Setup

Benchmark settings. We evaluate InteractVLA on NAVSIM (Dauner et al., 2024) and Bench2Drive (Jia et al., 2024). On NAVSIM, we use `navtrain` for training and `navtest` for

Table 1: **Comparison on NAVSIM Benchmark.** 1x Cam (single front-view camera), Nx Cam (surround-view cameras), L (LiDAR). Best and second-best results within each method category are highlighted in **bold** and underline, respectively.

Method	Venue	Sensors	NC ↑	DAC ↑	TTC ↑	C. ↑	EP ↑	PDMS ↑
Human	-	-	100	100	100	99.9	87.5	94.8
VLA-based Methods								
AutoVLA (Zhou et al., 2025b)	NeurIPS'25	3x Cam	98.4	95.6	**98.0**	99.9	81.9	89.1
ReCogDrive (Xiong et al., 2026)	ICLR'26	3x Cam	97.9	97.3	94.9	100.0	87.3	90.8
SpanVLA (Zhou et al., 2026b)	arXiv'26	3x Cam	**99.1**	97.1	95.2	100.0	86.3	90.3
LatentVLA (Xie et al., 2026)	arXiv'26	Cam	98.9	98.2	95.2	100.0	88.2	92.4
SGDrive (Li et al., 2026a)	CVPR'26	1× Cam	98.6	97.8	96.2	100.0	85.8	91.1
ChainFlow-VLA (Wang et al., 2026c)	arXiv'26	Cam	98.8	**98.6**	96.1	100.0	**90.8**	**93.6**
World-Model-based Methods								
LAW (Li et al., 2025a)	ICLR'25	1× Cam	96.4	95.4	88.7	99.9	81.7	84.6
WoTE (Li et al., 2025b)	ICCV'25	3× Cam + L	98.5	96.8	94.9	99.9	81.9	88.3
PWM (Zhao et al., 2025)	NeurIPS'25	1× Cam	98.6	95.9	95.4	100.0	81.8	88.1
DriveWAM (Shi et al., 2026)	arXiv'26	1× Cam	98.3	98.1	95.2	100.0	84.3	90.1
DriveVA (Liu et al., 2026b)	ECCV'26	1× Cam	**99.0**	97.3	**98.4**	100.0	83.2	90.5
SimWAM (Zhao et al., 2026)	arXiv'26	1× Cam	98.4	**98.7**	95.5	100.0	**86.4**	**91.5**
World-Model-Augmented VLA Methods								
FSDrive (Zeng et al., 2025)	NeurIPS'25	1x Cam	98.2	93.8	93.3	99.9	80.1	85.1
DriveVLA-W0 (Li et al., 2026b)	ICLR'26	1x Cam	98.7	**99.1**	95.3	99.3	83.3	90.2
ExploreVLA (Sheng et al., 2026)	ECCV'26	1× Cam	98.8	98.4	96.5	99.9	83.5	90.4
HyWorldVLA (Yu et al., 2026)	arXiv'26	1× Cam	98.6	98.9	95.1	99.3	84.6	90.6
DriveWorld-VLA (Liu et al., 2026a)	ICML'26	3x Cam	99.1	98.2	96.1	100.0	85.9	91.3
DynVLA (Shang et al., 2026)	ICML'26	1x Cam	98.6	98.7	95.5	100.0	86.8	91.7
LWDrive (Yang et al., 2026)	arXiv'26	4× Cam	98.8	98.4	96.2	99.8	87.3	92.0
WCog-VLA (Yan et al., 2026b)	arXiv'26	6x Cam	**99.4**	98.8	98.5	100.0	87.1	92.9
InteractVLA (Ours)	-	1x Cam	99.3	99.0	**98.7**	100.0	**87.5**	**93.2**

evaluation. Conventional planning performance is measured by the Predictive Driver Model Score (PDMS) under the standard 4-second non-reactive simulation protocol. PDMS jointly assesses no at-fault collisions (NC), drivable-area compliance (DAC), time-to-collision (TTC), comfort (C.), and ego progress (EP). To assess performance across imagined interactions, we additionally conduct two successive decisions at $\Delta = 1$ second intervals, each predicting a 4-second trajectory. Concatenating the executed segment with the second planned action yields a 5-second trajectory, which is evaluated using horizon-matched replay simulation and reported separately from standard 4-second PDMS. We further evaluate closed-loop driving on Bench2Drive using Driving Score, Success Rate, Efficiency, and Comfort, following the standard protocol; detailed settings are provided in the supplementary material. Ablation studies are conducted on Navsim.

Training details. InteractVLA is initialized from the Emu3-based DriveVLA-W0 (Li et al., 2026b) and trained on 32 NVIDIA A100 40GB GPUs using AdamW and bfloat16 mixed precision. Actions are represented with the FAST tokenizer (Pertsch et al., 2025), while each 1-second observation transition is encoded by eight discrete tokens from a 128-entry codebook. The pretrained transition tokenizer is kept frozen throughout the three-stage training pipeline.

In Stage I, the action and transition branches are jointly optimized with learning rates of 10^{-4} and 2×10^{-4}, respectively, and a transition-loss weight of $\lambda_T = 0.5$. In Stage II, the learned feedback model is implemented as a two-layer temporal Transformer with a hidden dimension of 512. We collect eight imagined interaction chains per scene from 10,000 training scenes and fit the feedback model for three epochs with a learning rate of 10^{-4}.

In Stage III, group-relative policy optimization uses $K = 8$ interaction chains per initial observation, sampled with temperature 1.0 and top-$p = 0.95$. The action branch is optimized with a learning rate of 5×10^{-6} and a cosine schedule with 500 warm-up steps. For the interaction-horizon curriculum, the 6-second return term is initially disabled for 500 steps and then linearly increased to its full weight over the next 500 steps. During conventional evaluation, only the action branch is retained; the transition branch is used for continued planning in imagination.

4.2 Main Results

Navsim. Table 1 compares InteractVLA with recent VLA-, world-model-, and world-model-augmented VLA methods on NAVSIM v1. InteractVLA achieves 93.2 PDMS, outperforming

Table 2: **Comparison on Bench2Drive (CARLA).** Best and second-best results are highlighted in **bold** and underline, respectively.

Method	Venue	Driving Score ↑	Success Rate (%) ↑	Efficiency ↑	Comfort ↑
Orion (Fu et al., 2025)	ICCV'25	77.74	54.62	151.48	17.38
AutoVLA (Zhou et al., 2025b)	NeurIPS'25	78.84	57.73	146.93	**39.33**
SimLingo (Renz et al., 2025)	CVPR'25	85.07	67.27	**259.23**	33.67
PixelPilot (Tang et al., 2026)	ECCV'26	79.14	58.87	153.26	38.01
AutoMoT (Huang et al., 2026)	ICML'26	89.42	74.09	–	–
AnchorVLA (Liu et al., 2026c)	arXiv'26	89.92	**77.28**	251.14	28.94
BLUE (Ling et al., 2026)	EMNLP'26	90.58	76.18	–	–
LinkVLA (Wang et al., 2026b)	CVPR'26	91.01	74.55	255.84	34.62
InteractVLA	-	**91.13**	76.27	254.60	37.02

Table 3: **Planning via imagined interaction.**

Method	$PDMS_{5s}$ ↑
Human	90.6 (↓4.2)
DynVLA	79.9 (↓11.8)
InteractVLA	**88.3 (↓4.9)**

Table 4: **Ablation of the main components of InteractVLA.**

Variant	$PDMS_{4s}$ ↑	$PDMS_{5s}$ ↑
w/o Obs. Adapt.	93.0 (↓0.2)	87.6 (↓0.7)
w/o Int.-Aware Fdbk.	90.2 (↓3.0)	81.4 (↓6.9)
w/o Return-to-Go	91.9 (↓1.3)	83.1 (↓5.2)

Table 5: **Ablation of feedback.**

Feedback	$PDMS_{5s}$ ↑
Replay only	81.4 (↓6.9)
Learned only	84.9 (↓3.4)
Fixed fusion	86.5 (↓1.8)

DynVLA, LWDrive, and DriveWorld-VLA by 1.5, 1.2, and 1.9 points, respectively, and improving over WCog-VLA despite using only a single front-view camera. It also achieves strong safety and progress metrics, including 98.7 TTC and 87.5 EP. These results show that imagined interaction improves planning quality while preserving effective driving progress.

Bench2Drive. As shown in Table 2, InteractVLA achieves the highest Driving Score of 91.13 and a Success Rate of 76.27%, while maintaining competitive efficiency and comfort. The gains under closed-loop evaluation indicate that interaction experience generated during training transfers effectively to sequential decision-making, helping the policy adapt to states induced by its own actions and the evolving traffic environment.

4.3 Planning through Continued Imagined Interaction

Table 3 evaluates planning after continued imagined interaction. While all methods experience some degradation relative to standard planning, InteractVLA remains substantially more robust than DynVLA. Specifically, InteractVLA achieves a 5-s PDMS of 88.3, outperforming DynVLA by 8.4 points, while maintaining substantially higher NC (98.5 vs. 95.7) and DAC (96.6 vs. 90.9). More importantly, its PDMS decreases by only 4.9 points under continued planning, compared with a 11.8-point drop for DynVLA and close to the 4.2-point degradation of the human reference. These results indicate that InteractVLA better preserves planning quality when subsequent decisions are conditioned on observations induced by its own preceding trajectories, supporting the effectiveness of imagined interaction for continued planning.

4.4 Ablation Studies

Components Analysis. Table 4 validates the contribution of each component. Removing interaction-aware feedback causes the largest degradation, especially under continued planning (88.3→81.4), showing that conventional replay-based evaluation limits learning once imagined scenes diverge from logged traffic. Without return-to-go, the model still retains relatively strong 4-s performance (91.9) but drops markedly at 5 s (83.1), indicating that step-wise optimization improves short-horizon planning but fails to account for downstream consequences. Observation adaptation provides a smaller yet consistent gain, mainly improving robustness to reconstructed observations.

Interaction-Aware Feedback. Table 5 compares different feedback strategies for continued planning. Replay-based feedback performs worst, reflecting its limited reliability once imagined observations diverge from logged traffic. Learned feedback improves the 5-s PDMS, while combining the two further improves performance. The proposed adaptive fusion achieves the best result, supporting the use of observation correspondence to balance replay-based and learned feedback.



Figure 4: **Qualitative comparison between DynVLA and InteractVLA.** Representative driving scenes and the corresponding predicted ego trajectories are visualized in BEV. InteractVLA exhibits more consistent trajectory prediction across diverse interactive driving scenarios.

Effect of Interaction Horizon. As shown in Fig. 5, increasing the number of imagined interaction steps during training consistently improves the final 4-s planning performance. PDMS rises from 90.8 with one-step interaction to 93.2 with three steps, while extending to four steps yields only a marginal gain to 93.4. Meanwhile, the training time increases substantially from about 3 days for three-step interaction to about 10 days for four steps. This indicates that longer imagined interaction provides useful downstream experience, but the performance gain largely saturates beyond three steps while the training cost grows considerably. We therefore use three interaction steps as the default setting.

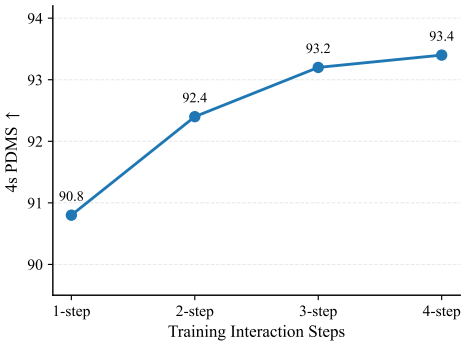

Figure 5: **Training interaction horizon.**

4.5 Qualitative Visualization

We further visualize representative interaction-intensive scenarios in Fig. 4. Compared with DynVLA, InteractVLA produces more consistent trajectories when responding to surrounding traffic and road geometry, such as at intersections, in dense traffic, and on curved roads. These scenarios require the policy to continually adapt its decisions to an evolving environment rather than relying only on the current observation. By learning from action-conditioned scene evolution, InteractVLA is exposed to states and consequences induced by its own behavior, leading to more reliable planning under interactive conditions. These qualitative results further support imagined interaction as an effective mechanism for policy self-improvement beyond fixed offline demonstrations.

5 Conclusion

We presented **InteractVLA**, a framework for self-improving autonomous driving through imagined interaction. InteractVLA converts action-conditioned future prediction into policy-dependent training experience, supported by efficient transition modeling, interaction-aware feedback, and multi-step policy optimization. Experiments on NAVSIM and Bench2Drive demonstrate strong standard planning performance and improved continued planning under policy-induced future states. These results highlight imagined interaction as a promising direction for enabling driving policies to learn from the consequences of their own behavior.

AI USE STATEMENT

We used generative AI for literature summarization, manuscript polishing, and code debugging. All AI-assisted content, including technical claims, experimental interpretations, and code suggestions, was reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Yuntao Chen, Yuqi Wang, and Zhaoxiang Zhang. Drivinggpt: Unifying driving world modeling and planning with multi-modal autoregressive transformers. In *ICCV*, pp. 26890–26900. IEEE, 2025.

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. *arXiv preprint arXiv:2406.15349*, 2024.

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. In *ICCV*, pp. 24823–24834. IEEE, 2025.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *NeurIPS*, 37:91560–91596, 2024.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023a.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *CVPR*, 2023b.

Wenhui Huang, Songyan Zhang, Qihang Huang, Zhidong Wang, Zhiqi Mao, Collister Chua, Zhan Chen, Long Chen, and Chen Lv. Automot: A unified vision-language-action model with asynchronous mixture-of-transformers for end-to-end autonomous driving. In *ICML*, 2026.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

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.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *NeurIPS*, 37:819–844, 2024.

Anqing Jiang, Yu Gao, Yiru Wang, Zhigang Sun, Shuo Wang, Yuwen Heng, Hao Sun, Shichen Tang, Lijuan Zhu, Jinhao Chai, et al. Irl-vla: Training an vision-language-action policy via reward world model. *arXiv preprint arXiv:2508.06571*, 2025.

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 *ICCV*, 2023.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *JAIR*, 4:237–285, 1996.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. In *CVPR*, pp. 4032–4042, June 2026a.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *ICLR*, volume 2025, pp. 42942–42959, 2025a.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. In *ICCV*, pp. 27137–27146. IEEE, 2025b.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. In *ICLR*, volume 2026, pp. 7890–7911, 2026b.

Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-r1: Bridging reasoning and planning in vlms for autonomous driving with reinforcement learning. In *AAAI*, volume 40, pp. 6708–6716, 2026c.

George Ling, Lijin Yang, Hao Yang, and Zhongzhan Huang. Blue: Toward better language use in efficient vision-language-action models for autonomous driving. In *EMNLP*, 2026.

Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, Long Chen, et al. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026a.

Mengmeng Liu, Diankun Zhang, Jiuming Liu, Jianfeng Cui, Hongwei Xie, Guang Chen, Hangjun Ye, Michael Ying Yang, Francesco Nex, and Hao Cheng. Driveva: Video action models are zero-shot drivers. In *ECCV*, pp. 315–335. Springer, 2026b.

Qi Liu, Yabei Li, Hongsong Wang, Heng Zhang, and Lei He. Anchorvla: Bridging discrete decisions and continuous trajectories for vision-language-action planning. *arXiv preprint arXiv:2607.03182*, 2026c.

Qiqi Liu, Huan Xu, Jingyu Li, Bin Sun, Zhihui Hao, Dangen She, Xiatian Zhu, and Li Zhang. Uni-world vla: Interleaved world modeling and planning for autonomous driving. *arXiv preprint arXiv:2603.27287*, 2026d.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *CVPR*, pp. 11993–12003, 2025.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pp. 627–635, 2011.

Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, et al. Dynvla: Learning world dynamics for action reasoning in autonomous driving. In *ICML*, 2026.

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.

Zihao Sheng, Xin Ye, Jingru Luo, Sikai Chen, and Liu Ren. Explorevla: Dense world modeling and exploration for end-to-end autonomous driving. In *ECCV*, pp. 266–284. Springer, 2026.

Chen Shi, Jinrui Xu, Shaoshuai Shi, Kehua Sheng, Bo Zhang, and Li Jiang. Drivewam: Video generative priors enable scalable world-action modeling for autonomous driving. *arXiv preprint arXiv:2605.28544*, 2026.

Bin Sun, Yaoguang Cao, Yan Wang, Rui Wang, Jiachen Shang, Xiejie Feng, Jiayi Lu, Jia Shi, Shichun Yang, Xiaoyu Yan, et al. Minddrive: An all-in-one framework bridging world models and vision-language model for end-to-end autonomous driving. *arXiv preprint arXiv:2512.04441*, 2025.

Pin Tang, Guoqing Wang, Xiangxuan Ren, Zhongdao Wang, Guodongfang Zhao, Bailan Feng, and Chao Ma. Pixelpilot: Scalable vision-language-action models for end-to-end autonomous driving. In *ECCV*, pp. 465–484. Springer, 2026.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. In *CoRL*, 2024.

Guoqing Wang, Pin Tang, Xiangxuan Ren, Guodongfang Zhao, Bailan Feng, and Chao Ma. Learning vision-language-action world models for autonomous driving. In *CVPR Findings*, 2026a.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *CVPR*, pp. 22442–22452. IEEE, 2025.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *ECCV*, pp. 55–72. Springer, 2024a.

Xinyang Wang, Qian Liu, Wenjie Ding, Zhao Yang, Wei Li, Chang Liu, Bailin Li, Kun Zhan, Xianpeng Lang, and Wei Chen. Unifying language-action understanding and generation for autonomous driving. In *CVPR*, pp. 25193–25203, June 2026b.

Xiyang Wang, Xinlin Wang, Tingguang Zhou, Gong Chen, Xingtai Gui, Zhi Xu, Xiaolei Wu, Feiyang Tan, Hangning Zhou, and Mu Yang. Chainflow-vla: Causal flow planning with vision-language models. *arXiv preprint arXiv:2605.23270*, 2026c.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *CVPR*, pp. 14749–14759. IEEE, 2024b.

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. *NeurIPS*, 2022.

Chengen Xie, Bin Sun, Tianyu Li, Junjie Wu, Zhihui Hao, XianPeng Lang, and Hongyang Li. Latentvla: Efficient vision-language models for autonomous driving via latent action prediction. *arXiv e-prints*, pp. arXiv–2601, 2026.

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *ICLR*, volume 2026, pp. 157518–157556, 2026.

Tianyi Yan, Tao Tang, Xingtai Gui, Yongkang Li, Jiasen Zheng, Weiyao Huang, Lingdong Kong, Wencheng Han, Xia Zhou, Xueyang Zhang, et al. Ad-r1: Closed-loop reinforcement learning for end-to-end autonomous driving with impartial world models. In *CVPR*, pp. 1085–1095, 2026a.

Xuerun Yan, Zhexi Lian, Nuoheng Zhang, Shiyu Fang, Haoran Wang, Chen Lv, Jia Hu, and Binyang Song. Wcog-vla: A dual-level world-cognitive vision-language-action model for end-to-end autonomous driving. *arXiv preprint arXiv:2607.08375*, 2026b.

Chen Yang, Yuhao Wei, Ze Xu, Ziheng Zou, Shuang Liang, Delin Ouyang, Lingfeng Qi, Jie Li, and Guofa Li. Lwdrive: Layer-wise world-model-guided vision-language model planning for autonomous driving. *arXiv preprint arXiv:2606.29879*, 2026.

Quanfu Yu, Xian Wu, Hao Xu, and Liulong Ma. Hyworldvla: A vision-language-action model with hybrid world modeling for autonomous driving. *arXiv preprint arXiv:2607.20988*, 2026.

Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *NeurIPS*, 38:67299–67318, 2025.

Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.

Bing Zhan, Shuyao Shang, Jiahao Gu, Shuo Lu, Yuan Xu, Zhao Wang, Yida Wang, Xueyang Zhang, Kun Zhan, Lue Fan, et al. Brainwam: Action-space coordination of semantic priors and predictive dynamics for autonomous driving. *arXiv preprint arXiv:2608.12854*, 2026.

Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From forecasting to planning: Policy world model for collaborative state-action prediction. *NeurIPS*, 38:134585–134611, 2025.

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Honglin Li, Dingkang Liang, and Xiang Bai. Simwam: A simple world action model for end-to-end autonomous driving. *arXiv preprint arXiv:2608.07468*, 2026.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *ECCV*, pp. 55–72. Springer, 2024.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, et al. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *ICCV*, pp. 28632–28642. IEEE, 2025.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*, 2025a.

Yang Zhou, Xiaofeng Wang, Hao Shao, Letian Wang, Guosheng Zhao, Jiangnan Shao, Jiagang Zhu, Tingdong Yu, Zheng Zhu, Guan Huang, et al. Drivedreamer-policy: A geometry-grounded world-action model for unified generation and planning. *arXiv preprint arXiv:2604.01765*, 2026a.

Zewei Zhou, Tianhui Cai, Seth 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. *NeurIPS*, 38:27920–27956, 2025b.

Zewei Zhou, Ruining Yang, Yiluan Guo, Sherry X Chen, Tao Feng, Kateryna Pistunova, Yishan Shen, Lili Su, Jiaqi Ma, et al. Spanvla: Efficient action bridging and learning from negative-recovery samples for vision-language-action model. *arXiv preprint arXiv:2604.19710*, 2026b.

APPENDIX

A Overall **14**

B Datasets and Evaluation Metrics **14**

B.1 NAVSIM v1 14

B.2 Bench2Drive 15

C More Implementation Details **16**

C.1 Transition Token Extraction 16

C.2 Interaction-Aware Feedback Modeling 16

C.3 GRPO Optimization 18

D More Experiments **20**

D.1 Inference Efficiency. 20

D.2 Ablation on Compact Transition Tokens 20

E More Visualizations **21**

E.1 Failure Cases 21

E.2 Imagined Interaction 21

F Additional Related Work **21**

A OVERALL

This appendix provides additional details, analyses, and qualitative results to complement the main paper. §B describes the datasets and evaluation protocols used on NAVSIM v1 and Bench2Drive. §C presents further implementation details, including transition-token extraction, interaction-aware feedback modeling, and GRPO optimization over multi-step imagined interactions. §D reports additional experiments on inference efficiency and compact transition modeling. §E provides further qualitative analyses, including representative failure cases and visualizations of imagined interaction rollouts. Finally, §F extends the discussion of related work on conventional end-to-end driving and recent Vision-Language-Action models.

B DATASETS AND EVALUATION METRICS

We evaluate InteractVLA on NAVSIM v1 (Dauner et al., 2024) and Bench2Drive (Jia et al., 2024), which provide complementary non-reactive and closed-loop evaluation protocols.

B.1 NAVSIM v1

NAVSIM v1 (Dauner et al., 2024) is built on nuPlan driving data and evaluates planning through non-reactive simulation. Following the standard protocol, we report results under the 4-s planning setting using the Predictive Driver Model Score (PDMS).

PDMS consists of five components: No At-Fault Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time-to-Collision (TTC), and Comfort (C). The final score is computed as

$$\mathrm{PDMS} = \mathrm{NC} \cdot \mathrm{DAC} \cdot \frac{5\,\mathrm{EP} + 5\,\mathrm{TTC} + 2\,\mathrm{C}}{12}. \tag{12}$$

No At-Fault Collision (NC). NC measures whether the simulated ego trajectory causes an at-fault collision. Collisions with active road users receive the strongest penalty, collisions with static objects receive a softer penalty, and collision-free trajectories obtain the maximum score.

Drivable Area Compliance (DAC). DAC measures whether the ego vehicle remains within the drivable area during simulation. A trajectory receives full credit if the ego footprint stays inside the drivable region and zero otherwise.

Ego Progress (EP). EP measures how much progress the ego vehicle makes along the intended route. The achieved progress is normalized by the safe progress estimated by the PDM-Closed planner, rewarding trajectories that advance effectively without sacrificing safety.

Time-to-Collision (TTC). TTC evaluates whether the planned ego motion enters an unsafe configuration with surrounding traffic. The ego state is projected forward and checked for potential collisions within a predefined temporal horizon. A trajectory receives a penalty if the predicted time-to-collision falls below the safety threshold.

Comfort (C). Comfort evaluates trajectory smoothness using vehicle dynamics including longitudinal and lateral acceleration, yaw rate, yaw acceleration, and jerk. A trajectory receives full credit when these quantities remain within the benchmark-defined comfort bounds.

All component scores are normalized to [0,1] when computing PDMS; for consistency with the benchmark convention, we report the scores multiplied by 100 in the tables.

In addition to the standard 4-s evaluation, we also report a 5-s continuation setting. The policy continues planning from a state induced by its preceding decision, and the resulting trajectory is evaluated using the same PDMS metrics. This setting is used to assess planning quality under successive policy-induced states.

B.2 BENCH2DRIVE

Bench2Drive (Jia et al., 2024) is a CARLA-based closed-loop benchmark covering diverse interactive driving scenarios. We follow the official evaluation protocol and report Driving Score (DS), Success Rate (SR), Efficiency, and Comfort.

Driving Score (DS). Driving Score combines route completion with penalties for driving infractions. For each route, the completed route percentage is multiplied by penalties associated with events such as collisions, red-light violations, and other traffic-rule violations. The final score is averaged over all evaluation routes.

Success Rate (SR). Success Rate measures the percentage of routes completed successfully without major infractions:

$$\mathrm{SR} = \frac{N_{\mathrm{success}}}{N_{\mathrm{total}}} \times 100\%. \tag{13}$$

Efficiency. Efficiency measures whether the ego vehicle maintains reasonable driving progress relative to surrounding traffic. The benchmark compares ego speed with the average speed of nearby vehicles at multiple checkpoints along the route and averages the resulting ratios.

Comfort. Comfort evaluates driving smoothness from vehicle dynamics such as acceleration, yaw motion, and jerk. The trajectory is divided into temporal segments, and a segment is considered comfortable when the corresponding dynamic quantities remain within the benchmark-defined bounds. The final score is the percentage of comfortable segments.

NAVSIM provides scalable evaluation of trajectory safety and progress under logged traffic, while Bench2Drive evaluates the policy in a closed-loop environment where ego actions affect subsequent states. We therefore use the two benchmarks together to evaluate both standard planning performance and the ability to adapt under sequential interaction.

C More Implementation Details

C.1 Transition Token Extraction

Following DynVLA (Shang et al., 2026), we pretrain a discrete transition tokenizer to extract compact representations of scene changes between consecutive observations. Given two adjacent observations O_t and O_{t+1}, we first patchify them into visual sequences and feed them into a Transformer-based transition encoder. Since visual changes in driving arise from both ego motion and the motion of surrounding traffic, we decouple the transition representation into *ego-centric* and *environment-centric* components.

Specifically, two sets of learnable queries, Q_{ego} and Q_{env}, aggregate the encoded features into continuous transition representations,

$$(E_t^{\text{ego}}, E_t^{\text{env}}) = \mathcal{E}_T\left(O_t, O_{t+1}; Q_{\text{ego}}, Q_{\text{env}}\right). \tag{14}$$

We maintain separate vector-quantization codebooks $\mathcal{C}_{\text{ego}}$ and $\mathcal{C}_{\text{env}}$ for the two branches. Each continuous representation is assigned to its nearest codebook entry, producing discrete ego-centric and environment-centric transition tokens. The final transition-token target is obtained by concatenating the two sets,

$$T_t^* = [T_t^{\text{ego}}, T_t^{\text{env}}]. \tag{15}$$

This discrete bottleneck provides a compact representation of the scene evolution from O_t to O_{t+1}.

To ensure that the learned tokens capture scene dynamics rather than static appearance, the tokenizer is trained by reconstructing the future state conditioned on the current observation and the quantized transition representation. Following DynVLA (Shang et al., 2026), we additionally regularize the ego-centric tokens with the relative ego motion between consecutive frames, encouraging them to explicitly encode viewpoint changes caused by the ego vehicle. The same transition representation is also used to reconstruct the future image and BEV representation, providing cross-view consistency supervision for environment dynamics.

The overall tokenizer objective is

$$\mathcal{L}_{\text{tok}} = \mathcal{L}_{\text{rec}}^{\text{img}} + \lambda_{\text{bev}}\mathcal{L}_{\text{rec}}^{\text{bev}} + \lambda_{\text{vq}}\mathcal{L}_{\text{VQ}} + \lambda_{\text{ego}}\mathcal{L}_{\text{ego}}, \tag{16}$$

where $\mathcal{L}_{\text{rec}}^{\text{img}}$ and $\mathcal{L}_{\text{rec}}^{\text{bev}}$ denote future image and BEV reconstruction losses, $\mathcal{L}_{\text{VQ}}$ is the standard vector-quantization objective, and $\mathcal{L}_{\text{ego}}$ supervises the ego-centric representation with relative ego motion.

After pretraining, the transition tokenizer and decoder are kept fixed. For each pair of consecutive logged observations, the tokenizer extracts T_t^* as the supervision target for the transition pathway described in the main paper. During imagined interaction, the transition pathway predicts these discrete tokens directly from the current visual context and action representations, without access to the future observation.

C.2 Interaction-Aware Feedback Modeling

We provide additional implementation details of the Reward Model introduced in Sec. 3.3.2, including supervision construction, transition-feature aggregation, model architecture, training objectives, and visual correspondence.

Supervision Construction. We construct Reward-Model training data using model-generated rollouts produced by a fixed action-transition model. Each rollout starts from a logged observation and contains three successive planning steps. At step t, the VLA samples action tokens $\hat{A}_t$, which are decoded by $\mathcal{D}_A$ into a 4-s trajectory $\hat{\tau}_t$ represented by eight relative motion increments at 0.5-s intervals. The transition pathway predicts the corresponding transition tokens, which are decoded deterministically to obtain the visual context for the next planning step.

We execute the first $\Delta = 1\,\mathrm{s}$ of each preceding trajectory and compose it with the current 4-s prediction. This yields continuation trajectories with horizons of 4, 5, and 6 seconds:

$$\begin{aligned}\bar{\tau}_1 &= \hat{\tau}_1,\\ \bar{\tau}_2 &= \hat{\tau}_1^{[0:\Delta]} \oplus \hat{\tau}_2,\\ \bar{\tau}_3 &= \hat{\tau}_1^{[0:\Delta]} \oplus \hat{\tau}_2^{[0:\Delta]} \oplus \hat{\tau}_3,\end{aligned} \tag{17}$$

where $\oplus$ transforms and concatenates trajectory segments in the coordinate frame of the initial observation. The three trajectories are evaluated using the NAVSIM replay evaluator with a simulation interval of 0.1 s and horizon-matched metric caches. All evaluations share the same initial scene anchor.

For each valid rollout, we retain the transition representations together with the replay-derived NC and TTC scores for the three horizons. We process up to $10{,}000$ eligible training scenes and sample eight rollouts per scene. Rollouts with incomplete predictions or missing metric-cache coverage are discarded.

Trajectory-Aligned Transition Representations. The Reward Model operates on the final normalized hidden states of the transition pathway before projection to transition-token logits. At each planning step, these states contain four groups of $N = 8$ transition tokens corresponding to four successive one-second intervals. We average the tokens within each interval,

$$\mathbf{z}_{t,j} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{h}_{t,j,n}, \qquad j \in \{1,2,3,4\}, \tag{18}$$

where $\mathbf{h}_{t,j,n} \in \mathbb{R}^D$ denotes the n-th transition hidden state of interval j at planning step t.

To align the transition representations with the continuation trajectories in Eq. 17, we construct

$$\begin{aligned}\mathbf{Z}_4 &= [\mathbf{z}_{1,1:4}],\\ \mathbf{Z}_5 &= [\mathbf{z}_{1,1}, \mathbf{z}_{2,1:4}],\\ \mathbf{Z}_6 &= [\mathbf{z}_{1,1}, \mathbf{z}_{2,1}, \mathbf{z}_{3,1:4}].\end{aligned} \tag{19}$$

Each resulting sequence therefore contains one transition feature per second of the evaluated trajectory.

Reward Model Architecture. Each transition feature is projected to a 512-dimensional embedding and augmented with a sinusoidal encoding of its interval-end time. The resulting sequence is processed by a two-layer Transformer encoder with eight attention heads, a feed-forward dimension of 1536, GELU activations, and a dropout rate of 0.1. We use pre-layer normalization and allow full attention across all valid trajectory intervals.

The final valid representation is passed through layer normalization and a two-layer MLP with dimensions $512 \to 256 \to 2$ and a SiLU activation. The two outputs estimate the NC and TTC component scores:

$$\mathbf{q}_{H,\mathrm{rm}} = \sigma(f_\phi(\mathbf{Z}_H, \mathbf{t}_H)), \qquad H \in \{4,5,6\}, \tag{20}$$

where $\mathbf{t}_H = (1,\ldots,H)$ denotes the interval-end times. Both outputs lie in $[0,1]$; the TTC output represents compliance with the TTC criterion rather than a collision time in seconds.

Reward Model Training. Let ℓ_H^m denote the predicted logit for component m at horizon H, and y_H^m its replay-derived target. We optimize

$$\begin{aligned}\mathcal{L}_{\mathrm{RM}} = \mathbb{E}_{\mathcal{D}_{\mathrm{RM}}}\Bigg[\frac{1}{3}\sum_{H\in\{4,5,6\}} \Big(&\mathcal{L}_{\mathrm{BCE}}(\ell_H^{\mathrm{NC}}, y_H^{\mathrm{NC}})\\ &+\mathcal{L}_{\mathrm{SL1}}(\sigma(\ell_H^{\mathrm{TTC}}), y_H^{\mathrm{TTC}})\Big)\Bigg].\end{aligned} \tag{21}$$

Binary cross-entropy is applied directly to the numerical NC targets returned by the evaluator, while Smooth-L_1 with transition parameter 1 is used for TTC. All horizons and both components are weighted equally.

Only the Reward Model parameters ϕ are optimized in this stage; the action-transition model remains fixed and its transition features are cached before training. We train the Reward Model for three epochs using a learning rate of 10^{-4}, weight decay of 0.01, a batch size of 64 rollouts per device, and gradient accumulation over two steps.

Visual Correspondence. Interaction-aware feedback combines replay-based and Reward-Model estimates according to the correspondence between the generated visual context and its logged counterpart. Let $\hat{v}_{t,i}$ and $v_{t,i}^{\log}$ denote the visual-token indices at grid position i, and let $\mathbf{e}(v)$ denote the corresponding embedding in the frozen visual-tokenizer codebook. We compute

$$s_t = \left[\mathrm{clip}_{[0,1]} \left(\frac{1}{M} \sum_{i=1}^{M} \frac{1 + \cos\Big(\mathbf{e}(\hat{v}_{t,i}), \mathbf{e}(v_{t,i}^{\log})\Big)}{2} \right) \right]^{\gamma}, \tag{22}$$

where M denotes the number of visual tokens and $\gamma = 1$ in our default configuration. The same correspondence weight is used for NC and TTC. The initial planning step uses the logged observation and therefore retains the replay score directly.

Reward Aggregation. For $m \in \{\mathrm{NC}, \mathrm{TTC}\}$, replay and Reward-Model estimates are combined using Eq. 7. DAC, EP, and Comfort retain the corresponding replay-evaluator outputs. The step-level reward is therefore

$$R_t = \tilde{q}_t^{\mathrm{NC}}\, q_{t,\mathrm{rep}}^{\mathrm{DAC}} \frac{5q_{t,\mathrm{rep}}^{\mathrm{EP}} + 5\tilde{q}_t^{\mathrm{TTC}} + 2q_{t,\mathrm{rep}}^{\mathrm{C}}}{12}. \tag{23}$$

All component scores and R_t lie in $[0, 1]$. For the initial 4-s prediction, we use the original replay reward. Interaction-aware fusion is applied to the subsequent 5-s and 6-s continuation trajectories, where the generated visual context may deviate from logged traffic.

C.3 GRPO Optimization

Algorithm 1 summarizes the complete Stage-III optimization procedure. Starting from the same logged context, we sample multiple imagined interactions, propagate feedback across interaction steps through the return-to-go, and optimize the action-token policy using group-relative advantages.

Interaction Sampling. For each logged initial context, we sample K imagined interactions from the current policy. All interactions share the same initial visual context but evolve differently through stochastic action-token sampling. At interaction step t, the k-th rollout samples action tokens $\hat{A}_{t,k}$, which are decoded into

$$\hat{\tau}_{t,k} = \mathcal{D}_A(\hat{A}_{t,k}). \tag{24}$$

The transition pathway then generates the next visual context, which conditions the subsequent VLA decision.

Each step receives the interaction-aware reward $R_{t,k}$ defined above. To account for downstream consequences, we compute the interaction return

$$G_{t,k} = \sum_{j=t}^{T} w_{j-t} R_{j,k}, \tag{25}$$

where T is the interaction horizon and w_{j-t} controls the contribution of future rewards. We set $w_0 = 0.6$, $w_1 = 0.3$, and $w_2 = 0.1$. This weighting emphasizes immediate trajectory quality while preserving credit assignment from downstream consequences, whose uncertainty increases with the imagined interaction horizon.

Group-Relative Advantage. For interaction step t, we normalize the returns across the K rollouts:

$$\hat{\mathcal{A}}_{t,k} = \frac{G_{t,k} - \mu_t}{\sigma_t + \epsilon}, \tag{26}$$

where

$$\mu_t = \frac{1}{K} \sum_{k=1}^{K} G_{t,k}, \qquad \sigma_t = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (G_{t,k} - \mu_t)^2}. \tag{27}$$

Algorithm 1: Self-Improvement through Multi-Step Imagined Interaction

Input: Logged sequence $\{\mathbf{V}_t^{\log}\}_{t=1}^{T}$; VLA π_θ; transition pathway $\mathcal{T}$; action decoder $\mathcal{D}_A$; transition decoder $\mathcal{D}_T$; interaction-aware evaluator $\mathcal{R}$; interaction horizon T; group size K; execution interval Δ

Output: Updated VLA π_θ

foreach *optimization iteration* **do**

  Freeze behavior-model snapshot $\theta_{\text{old}} \leftarrow \theta$

  // 1. Generate and evaluate multi-step imagined interactions

  for $k = 1, \ldots, K$ **do**

    Initialize $\hat{\mathbf{V}}_{1,k} \leftarrow \mathbf{V}_1^{\log}$

    for $t = 1, \ldots, T$ **do**

      Jointly generate action and transition tokens

$$(\hat{A}_{t,k}, \hat{T}_{t,k}) \leftarrow \text{InteractVLA}\left(\hat{\mathbf{V}}_{t,k}; \theta_{\text{old}}\right)$$

      Decode trajectory

$$\hat{\tau}_{t,k} \leftarrow \mathcal{D}_A(\hat{A}_{t,k})$$

      Construct continuation trajectory

$$\bar{\tau}_{t,k} \leftarrow \left(\bigoplus_{j=1}^{t-1} \hat{\tau}_{j,k}^{[0:\Delta]}\right) \oplus \hat{\tau}_{t,k}$$

      Compute interaction-aware reward

$$R_{t,k} \leftarrow \mathcal{R}\left(\hat{\mathbf{V}}_{t,k}, \mathbf{V}_t^{\log}, \bar{\tau}_{t,k}\right)$$

      if $t < T$ **then**

        Generate next imagined observation

$$\hat{\mathbf{V}}_{t+1,k} \leftarrow \mathcal{D}_T\left(\hat{\mathbf{V}}_{t,k}, \hat{T}_{t,k}^{\Delta}\right)$$

  // 2. Propagate feedback across interaction steps

  for $t = 1, \ldots, T$ **do**

    for $k = 1, \ldots, K$ **do**

      Compute interaction return

$$G_{t,k} \leftarrow \sum_{j=t}^{T} w_{j-t} R_{j,k}$$

  // 3. Compute group-relative advantages

  for $t = 1, \ldots, T$ **do**

    Normalize returns across the K interaction chains

$$\hat{\mathcal{A}}_{t,k} \leftarrow \frac{G_{t,k} - \text{mean}_k(G_{t,k})}{\text{std}_k(G_{t,k}) + \epsilon}, \quad k = 1, \ldots, K$$

  // 4. Optimize with imagined and logged experience

  Update π_θ with the clipped GRPO objective using $\{\hat{A}_{t,k}, \hat{\mathcal{A}}_{t,k}\}_{t,k}$

  Update the transition pathway and feedback evaluator with supervised objectives on logged interactions

Table 6: **Ablation on compact transition modeling.** We compare different transition representations during imagined interaction.

Variant	Interaction Time (s/sample) ↓	4s PDMS ↑
w/o Transition	0.39 (only action)	91.3
Full Prediction	1.20	92.5
Compact Transition Tokens	**0.55**	93.2

This group-relative formulation compares different behaviors generated from the same initial driving context without requiring a learned value function.

Policy Objective. GRPO directly optimizes the sampled action-token sequences. Let $a_{t,k,n}$ denote the n-th token of $\hat{A}_{t,k}$. We define the token-level probability ratio

$$\rho_{t,k,n}(\theta) = \frac{\pi_\theta\left(a_{t,k,n} \mid \hat{\mathbf{V}}_{t,k}, a_{t,k,<n}\right)}{\pi_{\theta_{\text{old}}}\left(a_{t,k,n} \mid \hat{\mathbf{V}}_{t,k}, a_{t,k,<n}\right)}. \tag{28}$$

The clipped GRPO objective is

$$\begin{aligned}\mathcal{L}_{\text{GRPO}} = -\frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T}\frac{1}{|\hat{A}_{t,k}|}\sum_{n}\min\Big(&\rho_{t,k,n}(\theta)\hat{\mathcal{A}}_{t,k},\\ &\text{clip}\left(\rho_{t,k,n}(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}}\right)\hat{\mathcal{A}}_{t,k}\Big).\end{aligned} \tag{29}$$

All action tokens generated at the same interaction step share the corresponding interaction-level advantage. GRPO therefore optimizes the discrete action-token policy, while the decoded trajectory determines the resulting state transition and reward.

Optimization Schedule. We employ the horizon curriculum described in the main paper, starting from shorter imagined interactions and progressively increasing the interaction horizon. During GRPO optimization, gradients from $\mathcal{L}_{\text{GRPO}}$ are applied only to the VLA action pathway. The transition and Reward Model components provide imagined states and rewards but the policy objective does not backpropagate through these components. Their supervised updates are performed separately using logged driving data.

D More Experiments

D.1 Inference Efficiency.

We compare the inference efficiency of InteractVLA with DynVLA under the same setting, using a single NVIDIA A100 GPU with batch size 1 and BF16 precision. InteractVLA achieves an inference latency of 0.39 s per planning step, compared with 0.47 s for DynVLA. This improvement is primarily attributed to the absence of autoregressive textual or visual chain-of-thought generation at inference time. Instead, our world-modeling pathway serves as a training-time interaction engine for generating imagined experience and is removed at deployment. Consequently, InteractVLA directly predicts driving trajectories at inference, without additional reasoning rollouts.

D.2 Ablation on Compact Transition Tokens

Table 6 studies different transition modeling strategies during imagined interaction. Removing transition modeling yields the lowest interaction cost of 0.39 s/sample, but decreases the 4s PDMS to 91.3, indicating that explicitly modeling state transitions provides useful supervision for action learning. Full future prediction improves the PDMS to 92.5, but substantially increases the interaction cost to 1.20 s/sample. In contrast, our compact transition tokens achieve the best PDMS of 93.2 with only 0.55 s/sample, reducing the interaction cost by 54.2% compared with full prediction.

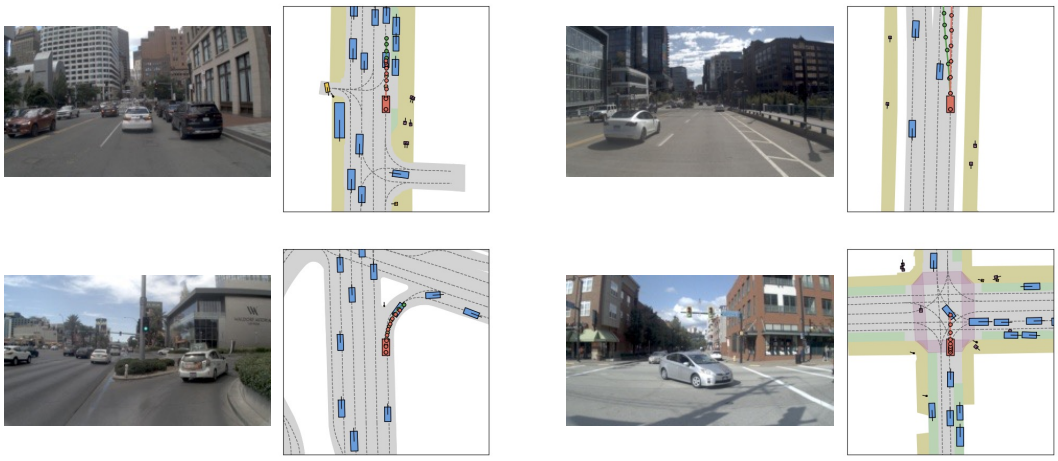

Figure 6: **Failure cases of InteractVLA.** The policy may behave conservatively in dense traffic and turning scenarios, deviate from the ground truth to avoid potential conflicts, or overlook latent risks from surrounding vehicles, highlighting the challenge of interaction-aware planning.

We attribute this improvement to two possible factors. First, predicting compact transition tokens avoids reconstructing redundant visual details, allowing the model to focus its capacity on action-relevant information and action prediction. Second, transition tokens explicitly capture scene dynamics, such as the motion and state changes of surrounding agents, which are more directly coupled with the consequences of ego actions than full observation reconstruction. Together, these results suggest that compact transition modeling provides a more effective and efficient interface between world dynamics and action learning.

E More Visualizations

E.1 Failure Cases

As shown in Figure 6, the remaining failures mainly stem from imperfect interaction forecasting in complex urban scenarios. In the two cases on the left, the ego vehicle is overly conservative in dense traffic and during a right turn, leading to slightly slower progress than the expert trajectory. In the upper-right case, the model performs a lane change to the left, deviating from the ground truth, likely because it anticipates the leading vehicle may start abruptly. In the lower-right case, the model initiates motion after observing a green light and the turning vehicle ahead nearly clearing the intersection, but overlooks the potential risk of a following vehicle behind it. These examples indicate that, while the model generally captures the high-level driving intent, it can still struggle to accurately anticipate subtle multi-agent interactions, especially when balancing safety and efficiency.

E.2 Imagined Interaction

Fig. 7 visualizes $K = 8$ imagined rollouts generated from the same logged context during training. For relatively simple scenes, the predicted futures remain visually consistent across rollouts. In contrast, intersections and dense-traffic scenarios exhibit more pronounced variations, where future observations are more strongly affected by interactions among the ego vehicle and surrounding traffic. This diversity provides richer imagined experience for learning under interactive driving scenarios.

F Additional Related Work

To complement the related work discussed in the main paper, we provide additional background on conventional end-to-end driving approaches and recent Vision-Language-Action (VLA) models considered in our closed-loop comparison on Bench2Drive.

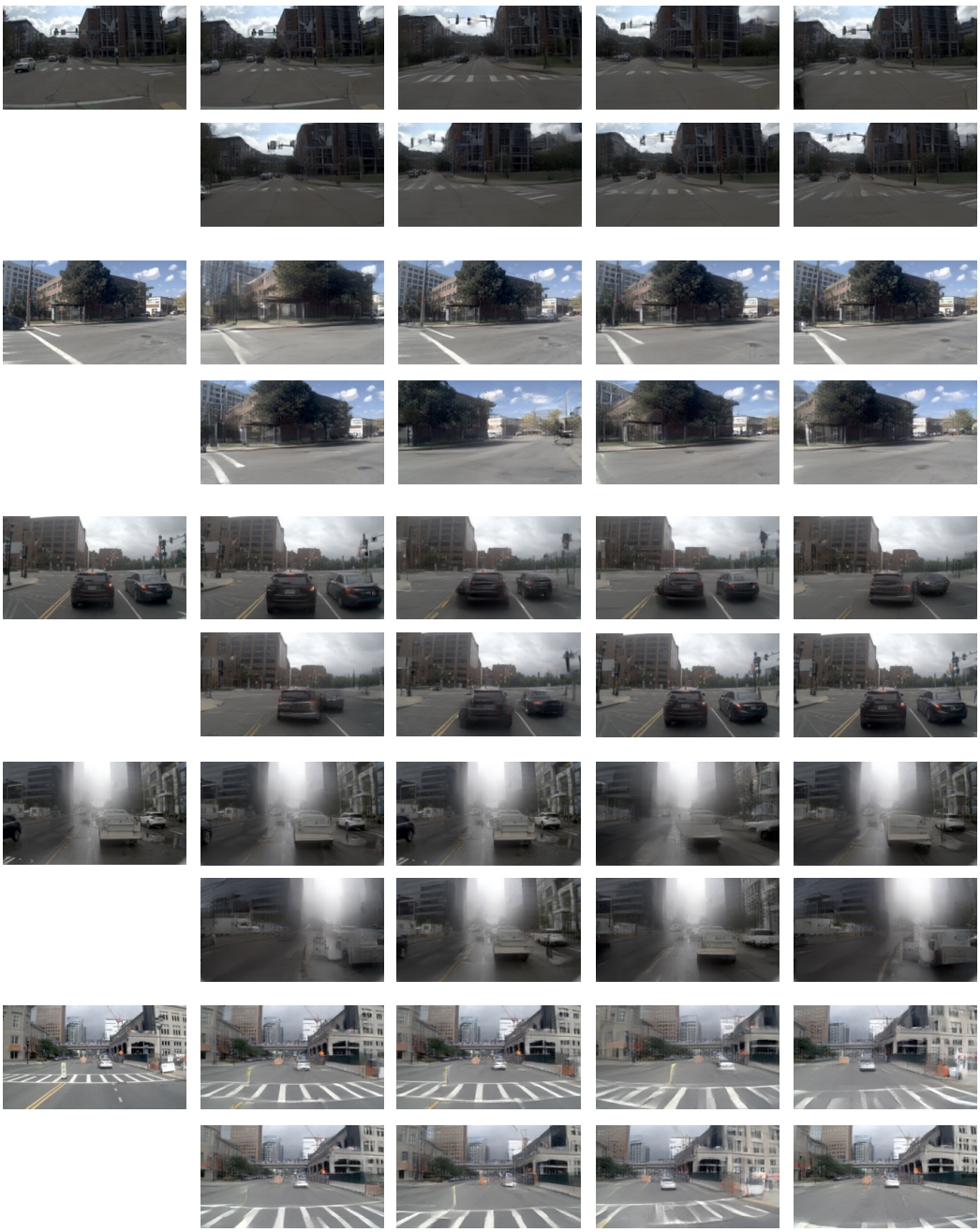

Figure 7: **Visualization of imagined interaction rollouts.** For each scene, the leftmost image shows the initial logged observation, followed by $K = 8$ imagined rollouts generated during training. The predicted futures remain relatively consistent in simple scenes, while more interactive scenarios, such as intersections and dense traffic, exhibit greater rollout diversity.

Conventional End-to-End Autonomous Driving. Before the emergence of VLA-based driving models, end-to-end autonomous driving primarily focused on directly mapping sensory observations to trajectories or low-level control commands. TCP (Wu et al., 2022) jointly predicts trajectories and vehicle controls, using trajectory prediction to facilitate multi-step control. UniAD (Hu et al., 2023b) formulates autonomous driving as a planning-oriented framework that jointly models perception, prediction, and planning through shared query representations. VAD (Jiang et al., 2023) further introduces a fully vectorized scene representation, representing surrounding agents and map elements

as structured constraints for trajectory planning. DriveAdapter (Jia et al., 2023) bridges perception and planning through feature-level adaptation between a perception student and a privileged planning teacher. In a different direction, AD-MLP (Zhai et al., 2023) demonstrates that strong open-loop planning performance can be obtained using only ego-state information, highlighting the limitations of evaluating planning solely under logged open-loop settings. Together, these methods establish strong foundations for end-to-end trajectory planning, but do not explicitly leverage the semantic reasoning and world knowledge provided by large vision-language models.

Vision-Language-Action Models for Autonomous Driving. Recent works incorporate pretrained vision-language models into end-to-end driving to enhance semantic understanding, reasoning, and action generation. ORION (Fu et al., 2025) combines long-term temporal aggregation, language-based driving reasoning, and generative trajectory prediction, with the goal of better aligning semantic reasoning with continuous driving actions. AutoVLA (Zhou et al., 2025b) unifies reasoning and trajectory generation within an autoregressive VLA and further introduces reinforcement fine-tuning based on GRPO to adaptively balance fast trajectory prediction and explicit reasoning. SimLingo (Renz et al., 2025) jointly addresses closed-loop driving, vision-language understanding, and language-action alignment through a camera-only VLA formulation. PixelPilot (Tang et al., 2026) instead performs planning in the image plane and deterministically lifts predicted trajectories into 3D, providing a sensor-agnostic interface for scalable VLA training.

More recent approaches explore specialized architectures and action representations to improve closed-loop performance and inference efficiency. AutoMoT (Huang et al., 2026) adopts an asynchronous Mixture-of-Transformers architecture, sharing attention between reasoning and action pathways while executing different tasks at different frequencies. AnchorVLA (Liu et al., 2026c) introduces trajectory-pattern anchors as an explicit interface between high-level driving decisions and continuous trajectory refinement, avoiding lengthy autoregressive coordinate generation. BLUE (Ling et al., 2026) observes that language reasoning is not equally beneficial for all driving situations and introduces a lightweight gate that selectively activates language generation when needed. LinkVLA (Wang et al., 2026b) unifies language and action tokens within a shared discrete codebook, strengthening language-action alignment while reducing the overhead of autoregressive trajectory generation.

Relation to InteractVLA. The above VLA approaches primarily improve autonomous driving through reasoning architectures, language-action alignment, action representations, or inference efficiency. InteractVLA is complementary to these directions. Rather than redesigning the reasoning or action space, we focus on how a driving VLA can acquire additional experience from its own decisions. Specifically, InteractVLA turns action-conditioned future prediction into a training-time interaction engine, enabling the VLA to generate multi-step imagined experience, obtain feedback consistent with the imagined traffic context, and learn from downstream consequences across successive planning decisions.