Learning Agent-Centric Vision-Language-Action World Models for Autonomous Driving

Guoqing Wang Pin Tang Xiangxuan Ren Chao Ma*
MoE Key Lab of Artificial Intelligence, Institute of AI
Shanghai Jiao Tong University, Shanghai, China
{guoqing.wang,pin.tang,bunny_renxiangxuan,chaoma}@sjtu.edu.cn

Project page: https://AgentWorld-AD.github.io

Abstract

Recent progress in integrating vision-language-action (VLA) models with world modeling has significantly advanced end-to-end autonomous driving by allowing systems to reason about imagined futures instead of merely reacting to current observations. However, most world VLA models generate future frames at the pixel level with plenty of dense visual tokens. Although such generated futures can be visually plausible, they preserve many driving-irrelevant appearance details, i.e., only a small portion of the tokens directly encode critical planning-relevant factors, such as agent geometry, motion dynamics, and interactions. This introduces a significant performance bottleneck. An important question thus arises: *what world state should a world VLA model reason over to support planning beyond pixel-level prediction?* We argue that planning-oriented world modeling should focus on a compact and structured agent-centric latent state that captures the interactions between objects in the driving scene over time. As such, we introduce **AgentWorld**, an agent-centric VLA world model that uses discrete agent tokens jointly for reasoning and trajectory planning. Each token is dynamically grounded to a surrounding agent associated with its object's state, such as semantic identity, 3D geometry, motion, and interaction context. AgentWorld learns these tokens through three key processes: explicit geometric grounding, reasoning-oriented supervised fine-tuning, and agent-oriented reinforcement learning. Extensive open-loop and closed-loop experiments under diverse driving scenarios show that AgentWorld enhances planning capabilities, fully exploits visual evidence and agent dynamics, and yields interpretable agent-centric world states.

1 Introduction

The advancements in Vision-Language-Action (VLA) [1–10] and world models [11–17] have significantly propelled end-to-end autonomous driving [18–22]. VLA models inherit the instruction-following and reasoning capabilities of Multimodal Large Language Models (MLLMs) [23–29], effectively connecting perception, reasoning, and action generation within a unified framework. World models complement this ability by predicting how the driving scene will evolve before an action is taken. Recent efforts aim to integrate these two approaches into VLA world models [30–34], enabling VLA systems to consider imagined futures to produce more reasonable actions.

Despite this promising paradigm shown in Fig. 1, most prior VLA world models imagine the future frames through pixel-level generation with autoregressive or multi-step diffusion. Although such generated future frames can be visually plausible, they retain numerous task-irrelevant visual details, i.e., texture, lighting, background, and visual continuity. Only a small portion of tokens explicitly encode

*Corresponding author.

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

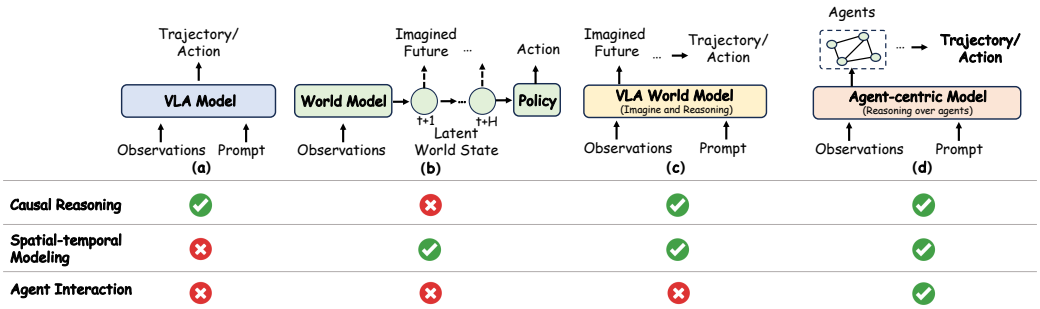

Figure 1: (a) VLA models output actions from observations and prompts, lacking explicit spatiotemporal modeling and interaction awareness. (b) World models learn predictive latent futures, yet their imagined states fail to align with causal reasoning. (c) VLA World models integrate imagination and language reasoning, but reason over generated frames rather than interactive agents for driving decisions. (d) By contrast, AgentWorld models futures via discrete latent agent states, enabling VLA reasoning over physically grounded 3D geometry, motion dynamics and agent interactions.

planning-relevant cues, including agent geometry, motion dynamics, and interactive behaviors. This introduces a significant bottleneck in VLA world models. As illustrated in Fig. 2, existing VLA world models demand far more tokens than the fine-tuned base model, i.e., Qwen2-VL [35], yet bring only slight planning improvement. Moreover, the visual grounding ablation reveals that removing image cues only slightly degrades existing world VLA models, even with their heavy use of dense visual tokens. This suggests that the visual inputs in these models are not yet fully transformed into planning-relevant reasoning, with many of the visual tokens instead being used for image generation. As such, the core challenge is not simply how to generate visually plausible future frames, but how to integrate world modeling with VLA through a compact, structured, and task-aligned state. An important question thus arises: *what world state should a VLA world model reason over to support planning beyond pixel-level prediction?*

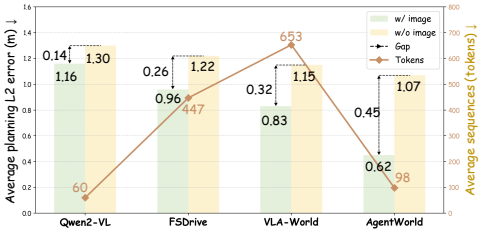

Figure 2: Visual grounding results (UniAD metric) and token efficiency of VLA models.

Motivated by JEPA-style paradigms [36–38], which prioritize predictive abstraction over pixel reconstruction, we argue that planning-oriented world modeling should emphasize a latent world state that captures the temporal interactions among objects in driving scenes. Rather than focusing on reconstructing dense visual futures, this state abstracts the scene into decision-relevant entities and their evolving relationships. We implement this idea by breaking down the scene into temporally evolving agents, each defined by semantic identity, 3D geometry, motion state, and interaction context. This structured abstraction enables the policy to recognize relevant agents, predict their motion evolution, and reason about how their future behaviors will influence the ego's trajectory. In this way, we align the latent world state with the causal structure of driving, shifting world modeling from future frame generation to structured reasoning about interacting agents.

In this paper, we introduce AgentWorld, an agent-based world VLA model designed for fully autonomous driving, as illustrated in Fig. 1. AgentWorld integrates causal reasoning, predictive world modeling, and agent interactions into a cohesive framework. In AgentWorld, the driving environment is represented by a set of discrete agent tokens, labeled as `<agent_i>`. These tokens are utilized for both reasoning and trajectory planning. This structure creates a well-defined interface among perception, reasoning, and planning processes. To make these latent agents physically meaningful, AgentWorld anchors them to 4D geometric features and 2D visual cues, and updates their hidden states through the MLLM into contextual representations of semantic identity, 3D geometry, motion, and interaction. AgentWorld learns these tokens via explicit geometric grounding, reasoning-oriented supervised fine-tuning (SFT), and agent-oriented reinforcement learning (RL) with group relative policy optimization [39]. The resulting agent tokens deliver a compact, physically grounded, and

interaction-aware representation of the world state, facilitating interpretable reasoning and planning regarding surrounding agents in driving scenarios.

Extensive open-loop and closed-loop experiments demonstrate the effectiveness of AgentWorld across diverse driving scenarios. By reasoning over interpretable agent-centric latent states, AgentWorld improves planning performance and better leverages visual evidence. These results validate agent-centric world states as an effective intermediate representation for VLA-based autonomous driving. Our contributions are summarized as follows:

- We reveal that the bottleneck of existing world VLA models stems from the fact that a considerably large number of visual tokens for future generation are irrelevant to planning, which motivates our design of compact, structured, planning-oriented latent world modeling.
- We introduce an *agent-centric latent state* for planning-oriented world modeling, which represents the driving scene as temporally evolving agents grounded in 3D geometry, motion, semantics, and interaction context.
- We propose **AgentWorld**, an agent-centric world VLA model that instantiates this state with discrete agent tokens. Experiments on open-loop and closed-loop benchmarks show consistently improved planning performance and more interpretable world states.

2 Related Work

2.1 Vision–Language–Action World Models

VLA models [1, 2, 40–43, 32, 44–52] and world models [12, 11, 53–68] have become two important directions for embodied intelligence and autonomous driving. Early driving VLMs [69, 4, 70, 40, 71, 72] demonstrate that language improves instruction following, scene understanding, and explicit reasoning in complex scenarios. However, language is often used as an auxiliary interface for explanation or high-level guidance, rather than being directly coupled with action generation. Recent VLA methods [2, 1, 3] address this limitation by unifying multimodal perception, language reasoning, and trajectory prediction within an end-to-end policy. In parallel, driving world models aim to improve planning through predictive scene evolution. Early works [17, 12, 15, 11] generate future driving scenes conditioned on videos, actions, or text prompts, while subsequent studies [73, 74, 56, 62, 75, 76] extend this paradigm with multi-view generation, autoregressive spatiotemporal modeling, and 4D occupancy prediction. These methods suggest that autonomous driving can benefit from reasoning over possible future states instead of reacting only to current observations. Recently, VLA world models have emerged by integrating VLA reasoning with world-model imagination. Representative approaches [30, 31, 33, 77–82] introduce future frames, latent world states, dynamic tokens, or spatiotemporal reasoning modules into action inference. Despite their progress, most existing methods still leave a fundamental question underexplored: *what world state should a VLA world reason over to support planning beyond pixel-level prediction?* We argue that this question is central to driving, where decisions are governed by physically grounded multi-agent interactions.

2.2 World State Representations for Autonomous Driving

The above question is closely related to how world states are represented for planning. Dense representations such as BEV [71, 83] and occupancy [84–86] preserve 3D spatial structure and geometric details, but they remain low-level and often entangle traffic participants with background elements, making them inefficient interfaces for high-level policy reasoning. Compact scene-level latents [34, 30, 77, 87] provide more efficient abstractions for future imagination, yet they typically compress multiple agents and interactions into global tokens, limiting structural interpretability and agent-level attribution. Language-aligned world states [1, 45, 42] naturally interface with MLLM reasoning, but are weakly grounded in physical quantities such as 3D geometry, motion, and inter-agent dynamics. As a result, policies can still rely on ego-status extrapolation, language priors, or dataset-specific correlations instead of modeling the physical evolution of surrounding agents. Consequently, existing representations either retain excessive low-level details, obscure agent-level causal factors, or lack explicit physical grounding, motivating a compact, interpretable, and agent-centric world state for driving-oriented reasoning.

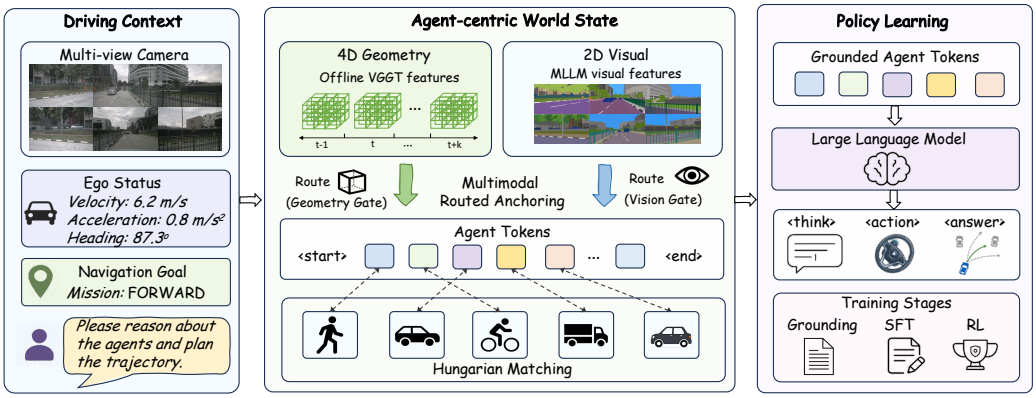

Figure 3: Overview of AgentWorld. Given multi-view observations, ego-status, navigation goals, and language prompts, AgentWorld constructs an agent-centric world state by grounding discrete agent tokens with 4D geometry and 2D visual features through multimodal routed anchoring. The grounded agent tokens are then used by the policy model for reasoning, action, and planning through grounding, supervised fine-tuning, and reinforcement learning stages.

3 Method

As illustrated in Fig. 3, AgentWorld builds an agent-centric VLA world model for end-to-end autonomous driving. Instead of treating future images or global latent features as the reasoning state, AgentWorld represents the driving scene with discrete agent tokens that are grounded in 4D geometry, 2D visual features, and multimodal context. These grounded tokens act as structured world states, exposing surrounding participants and their interactions to the policy for reasoning and planning. The method is organized into three components. Sec. 3.1 motivates agent-centric world states for VLA world models. Sec. 3.2 details how agent tokens are physically grounded through multimodal routed anchoring. Sec. 3.3 describes the policy learning with grounded agent tokens.

3.1 Rethinking World States for VLA World Models

Let $\mathcal{H}_t = \{o_{1:t}, g\}$ denote the driving context at time t, where $o_{1:t}$ contains historical multi-view observations and ego-status, and g is the navigation goal. The goal of end-to-end autonomous driving is to predict a future ego trajectory $\tau_{t:t+H}$. Following the VLA world model paradigm [30, 34, 77], planning can be viewed as decision-making augmented by predictive imagination:

$$p(\tau_{t:t+H}, \hat{\mathcal{S}} \mid \mathcal{H}_t) = p_w(\hat{\mathcal{S}} \mid \mathcal{H}_t)\, p_\pi(\tau_{t:t+H} \mid \mathcal{H}_t, \hat{\mathcal{S}}). \tag{1}$$

Here, p_w denotes a world model that constructs an imagined world state $\hat{\mathcal{S}}$ from the driving context, while p_π denotes a VLA that predicts the ego trajectory conditioned on both the current context and the imagined state. This abstraction reveals that the world state is not merely an auxiliary prediction target, but the information interface through which predictive imagination influences decision-making.

Therefore, the effectiveness of a VLA world model depends critically on the form of world state $\mathcal{S}$ over which the policy reasons. Pixel-level future observations provide an intuitive form of imagination, but require the policy to process many dense visual tokens, most of which are devoted to image generation rather than planning-relevant factors. As a result, agent geometry, motion dynamics, and interactions are only sparsely and implicitly encoded in pixels, leading to high computational costs and limited task alignment. Compact global latents [77] improve efficiency, but often entangle heterogeneous scene factors and obscure the agent-level causal structure needed for planning.

This motivates an agent-centric world state. For driving, the planning-relevant future is largely shaped by the evolution of surrounding participants: their 3D geometry, motion states, and interactions with the ego vehicle. Rather than representing the world state as dense pixels, language-only descriptions, or a compact scene latent, AgentWorld organizes the scene around potential agents. Specifically, we introduce a fixed set of special agent tokens into the MLLM sequence,

$$\mathcal{T}_{\text{agent}} = \{\langle \text{agent_1} \rangle, \langle \text{agent_2} \rangle, \cdots, \langle \text{agent_}N \rangle\}, \tag{2}$$

where N is the maximum number of agent tokens. After being grounded with multimodal evidence and updated by the MLLM, the hidden states at these token positions define the agent-centric latent state:

$$\mathcal{S}_t^{\text{agent}} = \{\mathbf{h}_t^1, \mathbf{h}_t^2, \cdots, \mathbf{h}_t^N\}, \quad \mathbf{h}_t^i \in \mathbb{R}^{D_{\text{LLM}}}, \tag{3}$$

where $\mathbf{h}_t^i$ is the MLLM hidden state at the position of $\langle \text{agent_}i \rangle$. Each token acts as an addressable agent slot that can be dynamically grounded to a surrounding participant or assigned to a "no-object" state. Through explicit physical grounding, active tokens are encouraged to encode semantic identity, 3D geometry, motion, and interaction context.

Under this formulation, AgentWorld performs planning by reasoning over grounded agent tokens:

$$p(\tau_{t:t+H} \mid \mathcal{H}_t) = p_\pi \left(\tau_{t:t+H} \mid \mathcal{H}_t, \mathcal{S}_t^{\text{agent}}\right). \tag{4}$$

The role of the world model is to construct $\mathcal{S}_t^{\text{agent}}$ as a compact, physically grounded, and interaction-aware state, rather than reconstructing all future visual details. These tokens are trained with semantic identity, 3D geometry, motion, and interaction context, encouraging the policy to infer how surrounding participants are likely to evolve and how their dynamics may influence the ego vehicle. This shifts VLA world models from future frame imagination toward agent-oriented reasoning.

3.2 Grounding Agent Tokens as World States

Given the agent-centric world state defined in Sec. 3.1, we now describe how each agent token is grounded to a physically verifiable participant. The grounding process consists of three steps: agent token initialization, multimodal routed anchoring, and physical grounding objective.

Agent token initialization. We initialize agent queries from the learnable embeddings of the special agent tokens, rather than introducing a separate object-query set. Let

$$\mathbf{E}_{\text{agent}} = \text{Emb}(\mathcal{T}_{\text{agent}}) \in \mathbb{R}^{N \times D_{\text{LLM}}}, \tag{5}$$

where each slot corresponds to one agent token. The initial agent queries are obtained by

$$\mathbf{Q}_t^0 = \mathbf{E}_{\text{agent}} \mathbf{W}_q = [\mathbf{q}_t^{1,0}; \cdots; \mathbf{q}_t^{N,0}] \in \mathbb{R}^{N \times d}. \tag{6}$$

Thus, each special token is itself a learnable agent query. At initialization, these queries are ungrounded latent slots that define where agent-centric world states will be formed in the MLLM sequence, without assuming any fixed correspondence to physical agents.

Multimodal routed anchoring. The architecture of the multimodal routed anchoring module is shown in Fig. 4. To ground the initialized agent queries, AgentWorld routes each slot to two complementary features: 4D geometry $\mathbf{G}_t^0$ from offline VGGT-style features following [88–90] and 2D visual $\mathbf{V}_t^0$ from the MLLM visual encoder. The geometry branch provides spatial layout, temporal consistency, and motion-related cues, while the vision branch provides appearance and semantic context. Both features are projected into a shared routing space:

$$\mathbf{G}_t = \phi_g(\mathbf{G}_t^0), \quad \mathbf{V}_t = \phi_v(\mathbf{V}_t^0). \tag{7}$$

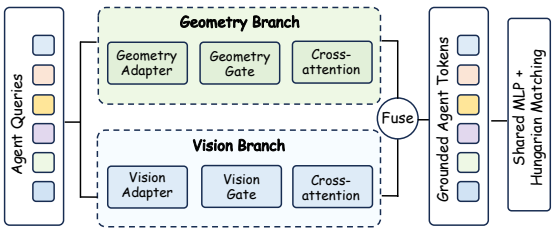

Figure 4: The pipeline of the multimodal routed anchoring. Agent queries are routed to geometry and vision branches, and fused through cross-attention to produce grounded agent tokens.

For each routing layer, an agent query is processed by two parallel branches. Each branch first adapts the query to its modality, then uses a token-wise gate to modulate the query before cross-attention:

$$\mathbf{R}_m^\ell = \text{CrossAttn}_m \left(\boldsymbol{\gamma}_m^\ell \odot \psi_m(\mathbf{Q}_t^\ell), \phi_m(\mathbf{M}_m), \phi_m(\mathbf{M}_m)\right), \quad \boldsymbol{\gamma}_m^\ell = \sigma(f_m(\mathbf{Q}_t^\ell)), \tag{8}$$

where $m \in \{g, v\}$, $\mathbf{M}_g = \mathbf{G}_t$, and $\mathbf{M}_v = \mathbf{V}_t$. The independent gates allow each token to decide how much geometry and vision to use, without forcing the two modalities to compete.

The geometry and vision responses are fused to update the agent queries:

$$\mathbf{Q}_t^{\ell+1} = \mathrm{Fuse}\left(\mathbf{Q}_t^{\ell}, \mathbf{R}_g^{\ell}, \mathbf{R}_v^{\ell}\right). \tag{9}$$

After L routing layers, the resulting grounded queries $\tilde{\mathbf{Q}}_t$ are projected to the MLLM hidden space and inserted into the corresponding $\langle \mathrm{agent_}i \rangle$ positions. The MLLM then contextualizes these tokens, and the hidden states at the corresponding positions serve as the grounded latent states $\{\mathbf{h}_t^i\}_{i=1}^N$. These states are subsequently supervised by the shared agent head and Hungarian matching.

Physical grounding objective. We ground each agent token by supervising its MLLM hidden state $\mathbf{h}_t^i$ with a lightweight training-time head:

$$h_{\mathrm{agent}}(\mathbf{h}_t^i) = (\hat{\mathbf{p}}_t^i, \hat{\mathbf{b}}_t^i, \hat{\boldsymbol{\tau}}_t^i), \tag{10}$$

which predicts category/existence, 3D bbox, and future trajectory. Because token order is unordered, predicted slots are matched to ground-truth agents with DETR-style bipartite matching [91]; matched slots receive classification, box, and trajectory losses, while unmatched slots are trained as no-object. This training-time head makes active tokens physically verifiable, leaving the grounded hidden states as agent-centric world states.

3.3 Policy Learning with Agent Tokens

Grounded agent tokens serve as the world-state interface for policy learning. AgentWorld exposes the learned slots through the `<perception>` field,

$$\texttt{<perception>}\langle \mathrm{agent_}1 \rangle \cdots \langle \mathrm{agent_}N \rangle\texttt{</perception>},$$

where the state of each participant is carried by the token hidden state rather than an explicit text description. The policy is trained to output a structured response $y = (\xi, a, \tau)$, consisting of a concise agent-grounded rationale `<think>`, a high-level driving action `<action>`, and future ego waypoints `<answer>`. This format forces planning to be expressed through addressable participants instead of dense perception text or image-level future prediction.

Reasoning-oriented SFT. After the physical grounding stage, we train the MLLM to use agent tokens as the interface for planning. Each expert response is organized as $y^\star = (\xi^\star, a^\star, \tau^\star)$, where $\xi^\star$ is a concise rationale over relevant agent tokens, $a^\star$ is a high-level driving action, and $\tau^\star$ is the expert future trajectory. We optimize the standard autoregressive objective:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_k \log p_\theta\left(y_k^\star \mid y_{<k}^\star, \mathcal{H}_t, \mathcal{S}_t^{\mathrm{agent}}\right), \quad \mathcal{S}_t^{\mathrm{agent}} = \{\mathbf{h}_t^i\}_{i=1}^N. \tag{11}$$

The rationale $\xi^\star$ is deliberately compact: it does not verbalize all perception results, but only identifies planning-relevant agent tokens, summarizes their interaction risk, and explains the resulting maneuver. For example, the model may state that `<agent_3>` is crossing the ego path and therefore the ego vehicle should decelerate and keep lane. This supervision encourages the MLLM to bind its reasoning and action generation to the grounded latent agent-centric world states rather than relying solely on ego-status history or generic driving priors. To bridge training and inference, the final SFT phase adopts the same MLLM-only input-output format used in RL, where the auxiliary grounding heads and routed anchoring modules are no longer required in inference.

Agent-oriented RL. We further optimize AgentWorld with GRPO [39]. Starting from the reasoning-oriented SFT checkpoint, the model samples multiple structured responses for each scene, each containing a concise rationale, a high-level action, and future ego waypoints. These candidates are evaluated by a composite reward that measures format, perception, action, trajectory, and interaction context. The trajectory reward encourages smoothness and imitation of expert behavior following [30], while the interaction reward explicitly evaluates the ego rollout against surrounding agents and penalizes collisions, insufficient clearance, and short time-to-collision. This reward design makes the RL stage agent-oriented: improving the final trajectory requires the policy to account for the dynamic agents represented by the learned `<agent_i>` tokens.

GRPO ranks multiple responses sampled from the same scene by their relative rewards, using the SFT model as a reference to stabilize updates. During RL, only the MLLM policy is updated. The

Table 1: End-to-end trajectory planning results on nuScenes [92]. We evaluate L2 error and collision rate following the respective protocols of ST-P3 [93] and UniAD [18]. The $*$ indicates the use of additional ego-state information.

Method	ST-P3 metrics								UniAD metrics								VLM
	L2 (m) ↓				Collision (%) ↓				L2 (m) ↓				Collision (%) ↓				
	1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.	
ELM [94]	-	-	-	-	-	-	-	-	0.34	1.23	2.57	1.38	0.12	0.50	2.36	0.99	BLIP2-2.7B
FeD* [95]	-	-	-	-	-	-	-	-	0.27	0.53	0.94	0.58	**0.00**	**0.04**	0.52	0.19	LLaVA-7B
OccWorld [56]	0.39	0.73	1.18	0.77	0.11	0.19	0.67	0.32	0.52	1.27	2.41	1.40	0.12	0.40	2.08	0.87	GPT3-like
Doe-1 [55]	0.37	0.67	1.07	0.70	0.02	0.14	0.47	0.21	0.50	1.18	2.11	1.26	0.04	0.37	1.19	0.53	Lumina-mGPT-7B
RDA-Driver* [96]	0.17	0.37	0.69	0.40	0.01	0.05	0.26	0.10	0.23	0.73	1.54	0.83	**0.00**	0.13	0.83	0.32	LLaVA-7B
EMMA* [97]	0.14	0.29	0.54	0.32	-	-	-	-	-	-	-	-	-	-	-	-	Gemini 1.0 Nano-1
OmniDrive* [71]	0.14	0.29	0.55	0.33	**0.00**	0.13	0.78	0.30	-	-	-	-	-	-	-	-	LLaVA-7B
FSDrive* [34]	0.14	0.25	0.46	0.28	0.03	0.06	0.21	0.10	0.18	0.39	0.77	0.45	**0.00**	0.06	0.42	0.16	Qwen2-VL-2B
VLA-World* [30]	0.10	0.24	0.45	0.26	0.02	0.05	0.18	0.08	0.10	0.35	0.80	0.42	0.01	0.05	0.30	0.12	Qwen2-VL-2B
AgentWorld	0.10	0.25	0.47	0.27	0.01	0.04	0.18	0.08	0.13	0.38	0.85	0.45	0.01	0.05	0.31	0.12	Qwen2-VL-2B
AgentWorld*	**0.09**	**0.23**	**0.44**	**0.25**	0.01	**0.03**	**0.17**	**0.07**	**0.09**	**0.34**	**0.77**	**0.40**	0.01	**0.04**	**0.28**	**0.11**	Qwen2-VL-2B

Table 2: Perception performance on nuScenes val set. We evaluate 3D object detection using both mAP and NDS metrics.

Method	mAP ↑	NDS ↑
CenterNet [99]	0.306	0.328
FCOS3D [100]	0.299	0.373
DETR3D [101]	0.303	0.374
PGD [102]	0.305	0.409
AgentWorld	**0.318**	**0.416**

Table 3: Action prediction performance on the nuScenes dataset measured by F1 score (%). † denotes models trained on nuScenes.

Method	Lateral (F1) ↑			Longitudinal (F1) ↑			
	forward	left	right	keep	acc.	dec.	stop
Qwen2-VL-2B	62.43	22.75	28.65	40.70	50.23	49.21	41.04
Qwen2-VL-2B†	92.60	61.78	66.52	56.42	74.32	76.10	74.85
VLA-World	**95.88**	74.22	75.06	60.98	81.42	80.04	81.24
AgentWorld	95.14	**77.63**	**78.41**	**63.85**	**84.17**	**82.93**	**84.56**

auxiliary agent head is retained solely as a frozen reward generator for agent-token consistency and interaction evaluation. At inference time, the reward-side agent head is not required and can be discarded. Thus, RL improves driving behavior while preserving the agent-token reasoning interface learned during SFT. Detailed rewards are provided in the appendix.

4 Experiments

4.1 Experimental Setups

Datasets and metrics. We evaluate AgentWorld on nuScenes [92] and Bench2Drive [98], covering both open-loop planning and closed-loop driving. For nuScenes, we report standard trajectory L2 error and collision rate. For Bench2Drive, we follow the official protocol and report Driving Score, Success Rate, Efficiency, Comfortness when available. More details of metrics can be found in the appendix.

Implementation details. AgentWorld is initialized from Qwen2-VL and Qwen2.5-VL series models [35, 25], with multi-view images as visual inputs and frozen VGGT-style features as offline 4D geometry features. We train the model in three stages: physical grounding SFT, reasoning-oriented SFT followed by agent-oriented RL. The SFT stages jointly supervise grounded agent tokens, high-level actions, reasoning traces, and future waypoints, while the RL stage optimizes sampled responses using format, perception, action, trajectory, and agent-interaction rewards. All experiments are implemented in PyTorch on 4×80GB NVIDIA A800 GPUs. Detailed hyperparameters, prompts, reward designs, and training schedules are provided in the appendix.

4.2 Main Results

Open-loop planning. Tab. 1 shows that AgentWorld achieves highly competitive results without additional ego-state input, outperforming most prior VLA and VLA world baselines in both L2 error and collision rate. When ego-status is further incorporated, AgentWorld obtains the best overall performance, reaching the lowest average L2 error and collision rate under both evaluation protocols.

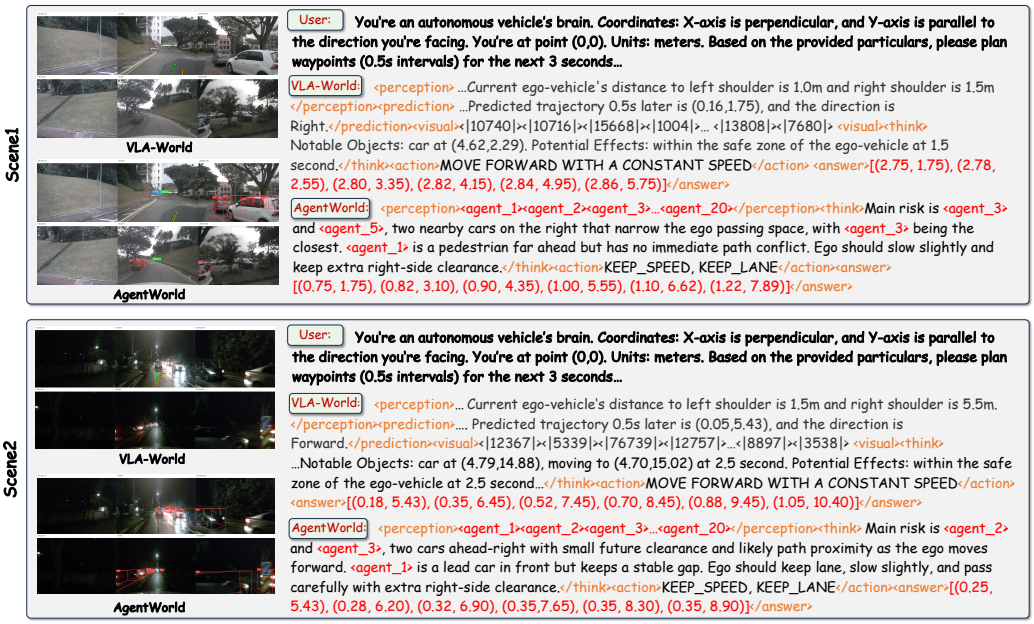

Figure 5: Qualitative comparison between VLA-World [30] and AgentWorld. For each scene, the left panels visualize the future ego trajectories of VLA-World (top) and AgentWorld (bottom), with AgentWorld additionally showing the traffic participants aligned with the special tokens $\langle \text{agent}_i \rangle$. The right panels present the corresponding perception, reasoning, and planning results.

Compared with VLA-World, the improvement is especially clear in long-horizon prediction and safety-critical collision metrics, suggesting that the agent-centric world state provides more reliable cues for future motion reasoning. These results demonstrate that explicitly modeling surrounding agents yields stronger and safer planning than relying on future imagination or generic VLA reasoning alone. The qualitative results are shown in Fig. 5. The visualized trajectories further show that AgentWorld produces more reliable waypoints in complex scenes with fewer tokens.

Perception results. As shown in Tab. 2, AgentWorld achieves the best performance on mAP and NDS. This consistent improvement suggests that the proposed agent-centric latent states are not merely useful for downstream planning but also retain physically grounded 3D information required for perception. More importantly, AgentWorld improves detection while using agent tokens as the intermediate world state. This indicates that explicitly grounding each latent agent to physical participants provides a structured representation that benefits spatial understanding.

Meta-action results. Tab. 3 reports action prediction performance on nuScenes. AgentWorld achieves the best overall results across most lateral and longitudinal action categories. The gains are especially pronounced for turning, lane keeping, acceleration, deceleration, and stopping, indicating that agent-centric world states provide more informative cues for both directional and speed-control decisions. Although VLA-World obtains a slightly higher score on the relatively saturated *forward* category, AgentWorld delivers more balanced improvements across interaction-sensitive actions. These results show that explicitly grounding and reasoning over surrounding agents leads to stronger and more reliable policy prediction.

Closed-loop planning. Tab. 4 reports closed-loop results on Bench2Drive. AgentWorld achieves the highest driving score and success rate among all compared methods, demonstrating stronger robustness in interactive simulation. It also obtains the best comfort score, suggesting that agent-centric reasoning improves not only task completion but also trajectory smoothness. Although VAD has a slightly higher efficiency score, AgentWorld provides a better overall balance across safety, success, and comfort, validating the effectiveness of agent-centric world states for closed-loop driving.

Table 4: Closed-loop evaluation results on Bench2Drive (CARLA) with SOTA methods.

Method	Driving Score ↑	Success Rate (%) ↑	Efficiency ↑	Comfortness ↑
AD-MLP [103]	18.05	0.00	48.45	22.63
UniAD-Base [18]	45.81	16.36	129.21	43.58
VAD [19]	42.35	15.00	**157.94**	46.01
TCP-traj [104]	59.90	30.00	76.54	18.08
DriveAdapter [105]	64.22	33.08	70.22	16.01
Orion [106]	77.74	54.62	151.48	17.38
AutoVLA [1]	78.84	57.73	146.93	39.33
AgentWorld	**81.17**	**61.36**	147.82	**54.39**

Table 5: Ablation on multi-modal routed anchoring.

Method	Avg. L2 (m) ↓
w/o 4D geometry	0.30
w/o 2D visual	0.28
AgentWorld	0.27

Table 6: Ablation on different learning strategies.

Method	Avg. L2 (m) ↓
Qwen2-VL-2B	1.37
Qwen2-VL-2B + SFT	0.32
Qwen2-VL-2B + SFT + RL	0.27

Table 7: Latency of different VLA world models w/o ego.

Method	Avg. L2 (m) ↓	Latency (s) ↓
FSDrive	0.53	0.36
VLA-World	0.30	0.40
AgentWorld	0.27	0.31

4.3 Ablation Study

Multimodal routed anchoring. Tab. 5 ablates the grounding components in AgentWorld. Removing either 4D geometry or 2D visual anchoring degrades planning performance, indicating that both geometric structure and visual evidence are necessary for reliable agent grounding. The full model achieves the lowest L2 error, showing that multimodal routed anchoring provides a more accurate and physically grounded agentic world state for planning. We also found that removing the 4D geometry features still yielded results comparable to the previous SOTA method [30], which proves the effectiveness of the agent-centric world state.

Policy learning strategies. Tab. 6 ablates policy learning strategies of AgentWorld. Directly using Qwen2-VL-2B leads to poor planning performance, while SFT (AgentWorld w/o RL) substantially reduces the L2 error. Further applying RL (AgentWorld) improves performance, showing that agent-centric policy optimization helps align the learned world state with planning objectives. These results validate the necessity of both supervised alignment and RL-based refinement in AgentWorld.

Efficiency of VLA world models. Tab. 7 compares the efficiency of different VLA world models. AgentWorld achieves the lowest L2 error while also having the shortest latency, outperforming both FSDrive and VLA-World. This shows that reasoning over compact agent-centric world states is not only more accurate but also more efficient than image-level future prediction. The result highlights the advantage of AgentWorld in balancing planning performance and inference efficiency, with potential benefits for practical autonomous driving.

Agent-centric world state. Fig. 6 validates the effectiveness of agent-centric world states. Increasing the number of agent tokens consistently improves perception and reduces planning errors before saturation, indicating that AgentWorld benefits from modeling key surrounding agents. Moreover, removing ego-status causes only a small degradation compared to prior VLA world models, suggesting reduced reliance on ego-status shortcuts and stronger reasoning over grounded agent interactions when predicting the ego vehicle's future trajectory.

4.4 Further Discussion

Controlled representation comparisons. Tab. 8 compares three representations under the same backbone, agent information, supervision, and SFT+RL training. Agent tokens achieve the lowest planning L2 error and collision rate, substantially outperforming both global latents and structured text. These gains across both metrics suggest that organizing world states around individual agents supports more accurate trajectory prediction and safer planning under the same training conditions. Structured text yields fewer collisions than global latents but incurs higher L2 error, whereas agent

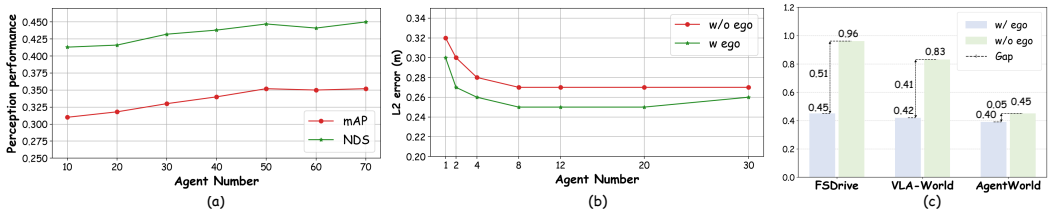

Figure 6: Ablation studies on agent-centric world states. (a) Perception and (b) planning performance with varying numbers of agent tokens. (c) Performance comparisons of w/ and w/o ego-status.

Table 8: Representation comparisons on nuScenes with the same backbone, agent information, supervision, and SFT+RL training.

Representation	L2 (m)↓	Col. (%)↓
Global latent	0.49	0.23
Structured text	0.58	0.21
Agent tokens	**0.27**	**0.08**

Table 9: Agent-token interventions on nuScenes. Consistent permutations preserve token–agent correspondence; mismatched permutations disrupt it.

Intervention	L2 (m)↓	Col. (%)↓
None	0.27	0.08
Consistent perm.	0.27 ± 0.01	0.08 ± 0.03
Mismatched perm.	0.63 ± 0.04	0.74 ± 0.09

tokens improve both metrics. These results suggest that explicitly organizing world states around individual agents provides a more effective representation for planning.

Agent-token interventions. Tab. 9 examines whether planning relies on the correspondence between agent tokens and their associated physical states. Consistent permutations preserve this correspondence and leave planning performance largely unchanged, whereas mismatched permutations substantially increase both trajectory error and collision rate. This contrast suggests that the planner relies on grounded token–agent associations rather than arbitrary slot indices. The pronounced rise in collisions further suggests that preserving these associations is important for reasoning about interactions with surrounding agents and producing safe trajectories.

5 Conclusion

We presented **AgentWorld**, an agent-centric VLA world model for end-to-end autonomous driving. Instead of relying on visually plausible future imagination alone, AgentWorld represents the driving scene through discrete agent tokens that are dynamically grounded to surrounding participants. These tokens encode semantic identity, 3D geometry, motion, and interaction, providing a compact and planning-oriented latent world state beyond images. Through explicit geometric grounding, reasoning-oriented supervised fine-tuning, and agent-oriented reinforcement learning, AgentWorld learns to reason over agents and output future trajectories. Extensive open-loop and closed-loop experiments demonstrate that this agent-centric formulation improves planning performance, reduces shortcut reliance on ego-status extrapolation and language priors, and enables more interpretable world states. Despite these promising results, AgentWorld still relies on explicit geometric supervision to ground $\langle \text{agent_}i \rangle$ tokens to physical participants, rather than allowing agent tokens to emerge purely from end-to-end planning supervision. Future work will explore more scalable self-supervised grounding with stronger visual-semantic binding, aiming to maintain consistent agent identities, physical states, and interaction semantics over time. We hope AgentWorld provides a step toward physically grounded and interaction-aware world modeling for autonomous driving.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62322113, 62376156), as well as the Shanghai Municipal Special Program for Basic Research on General AI Foundation Models (Grant No. 2025SHZDZX025G15).

References

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

[2] 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*, 2025.

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

[4] Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. In *ECCV*, pages 403–420, 2024.

[5] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *CVPR*, pages 15459–15469, 2024.

[6] Boyi Li, Yue Wang, Jiageng Mao, Boris Ivanovic, Sushant Veer, Karen Leung, and Marco Pavone. Driving everywhere with large language model policy adaptation. In *CVPR*, pages 14948–14957, 2024.

[7] Wencheng Han, Dongqian Guo, Cheng-Zhong Xu, and Jianbing Shen. Dme-driver: Integrating human decision logic and 3d scene perception in autonomous driving. In *AAAI*, pages 3347–3355, 2025.

[8] Haohan Chi, Huan-ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, et al. Impromptu vla: Open weights and open data for driving vision-language-action models. *arXiv preprint arXiv:2505.23757*, 2025.

[9] Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *WACV*, pages 1933–1943, 2025.

[10] 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*, pages 465–484, 2026.

[11] 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*, pages 14749–14759, 2024.

[12] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv:2309.09777*, 2023.

[13] Xiaofeng Wang, Zheng Zhu, Guan Huang, Boyuan Wang, Xinze Chen, and Jiwen Lu. Worlddreamer: Towards general world models for video generation via predicting masked tokens. *arXiv:2401.09985*, 2024.

[14] Jiachen Lu, Ze Huang, Zeyu Yang, Jiahui Zhang, and Li Zhang. Wovogen: World volume-aware diffusion for controllable multi-camera driving scene generation. In *ECCV*, pages 329–345, 2024.

[15] Fan Jia, Weixin Mao, Yingfei Liu, Yucheng Zhao, Yuqing Wen, Chi Zhang, Xiangyu Zhang, and Tiancai Wang. Adriver-i: A general world model for autonomous driving. *arXiv:2311.13549*, 2023.

[16] Chen Min, Dawei Zhao, Liang Xiao, Jian Zhao, Xinli Xu, Zheng Zhu, Lei Jin, Jianshu Li, Yulan Guo, Junliang Xing, et al. Driveworld: 4d pre-trained scene understanding via world models for autonomous driving. In *CVPR*, pages 15522–15533, 2024.

[17] 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*, 2023.

[18] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pages 17853–17862, 2023.

[19] 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*, pages 8340–8350, 2023.

[20] 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. *arXiv preprint arXiv:2402.13243*, 2024.

[21] Xiang Li, Pengfei Li, Yupeng Zheng, Wei Sun, Yan Wang, and Yilun Chen. Semi-supervised vision-centric 3d occupancy world model for autonomous driving. *arXiv preprint arXiv:2502.07309*, 2025.

[22] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *CVPR*, pages 14864–14873, 2024.

[23] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[24] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

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

[26] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, pages 23716–23736, 2022.

[27] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, pages 34892–34916, 2023.

[28] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pages 26296–26306, 2024.

[29] Xiangxuan Ren, Zhongdao Wang, Liping Hou, Pin Tang, Guoqing Wang, and Chao Ma. Grounding everything in tokens for multimodal large language models. In *CVPR*, pages 41171–41181, 2026.

[30] Guoqing Wang, Pin Tang, Xiangxuan Ren, Guodongfang Zhao, Bailan Feng, and Chao Ma. Learning vision-language-action world models for autonomous driving. *arXiv preprint arXiv:2604.09059*, 2026.

[31] Hao Lu, Ziyang Liu, Guangfeng Jiang, Yuanfei Luo, Sheng Chen, Yangang Zhang, and Ying-Cong Chen. Uniugp: Unifying understanding, generation, and planing for end-to-end autonomous driving. *arXiv preprint arXiv:2512.09864*, 2025.

[32] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

[33] 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*, 2026.

[34] 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. *arXiv preprint arXiv:2505.17685*, 2025.

[35] 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.

[36] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, pages 1–62, 2022.

[37] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, pages 15619–15629, 2023.

[38] Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

[39] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

[41] Weicheng Zheng, Xiaofei Mao, Nanfei Ye, Pengxiang Li, Kun Zhan, Xianpeng Lang, and Hang Zhao. Driveagent-r1: Advancing vlm-based autonomous driving with hybrid thinking and active perception. *arXiv preprint arXiv:2507.20879*, 2025.

[42] Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. *arXiv preprint arXiv:2311.10813*, 2023.

[43] 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.

[44] Kangan Qian, Sicong Jiang, Yang Zhong, Ziang Luo, Zilin Huang, Tianze Zhu, Kun Jiang, Mengmeng Yang, Zheng Fu, Jinyu Miao, et al. Agentthink: A unified framework for tool-augmented chain-of-thought reasoning in vision-language models for autonomous driving. *arXiv preprint arXiv:2505.15298*, 2025.

[45] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *ECCV*, pages 292–308, 2024.

[46] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[47] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, pages 2165–2183, 2023.

[48] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

[49] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[50] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. In *ICML*, 2023.

[51] Jie Wang, Guang Li, Zhijian Huang, Chenxu Dang, Hangjun Ye, Yahong Han, and Long Chen. Vggdrive: Empowering vision-language models with cross-view geometric grounding for autonomous driving. *arXiv preprint arXiv:2602.20794*, 2026.

[52] 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 preprint arXiv:2601.05611*, 2026.

[53] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[54] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. *arXiv preprint arXiv:2402.11502*, 2024.

[55] Wenzhao Zheng, Zetian Xia, Yuanhui Huang, Sicheng Zuo, Jie Zhou, and Jiwen Lu. Doe-1: Closed-loop autonomous driving with large world model. *arXiv preprint arXiv:2412.09627*, 2024.

[56] 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*, pages 55–72, 2024.

[57] 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*, pages 28632–28642, 2025.

[58] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, et al. Generalized predictive model for autonomous driving. In *CVPR*, pages 14662–14672, 2024.

[59] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CICAI*, pages 3–15, 2023.

[60] Tianshuo Xu, Hao Lu, Xu Yan, Yingjie Cai, Bingbing Liu, and Yingcong Chen. Occ-llm: Enhancing autonomous driving with occupancy-based large language models. *arXiv:2502.06419*, 2025.

[61] Jiazhi Yang, Kashyap Chitta, Shenyuan Gao, Long Chen, Yuqian Shao, Xiaosong Jia, Hongyang Li, Andreas Geiger, Xiangyu Yue, and Li Chen. Resim: Reliable world simulation for autonomous driving. *arXiv preprint arXiv:2506.09981*, 2025.

[62] Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *AAAI*, pages 9327–9335, 2025.

[63] Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. *arXiv preprint arXiv:2412.05435*, 2024.

[64] Bohan Li, Zhuang Ma, Dalong Du, Baorui Peng, Zhujin Liang, Zhenqiang Liu, Chao Ma, Yueming Jin, Hao Zhao, Wenjun Zeng, et al. Omninwm: Omniscient driving navigation world models. *arXiv preprint arXiv:2510.18313*, 2025.

[65] 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. In *NeurIPS*, pages 91560–91596, 2024.

[66] Xin Zhou, Dingkang Liang, Sifan Tu, Xiwu Chen, Yikang Ding, Dingyuan Zhang, Feiyang Tan, Hengshuang Zhao, and Xiang Bai. Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. In *ICCV*, pages 27817–27827, 2025.

[67] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *CVPR*, pages 15791–15801, 2025.

[68] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[69] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *CVPR*, pages 15120–15130, 2024.

[70] 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 *ECCV*, pages 256–274, 2024.

[71] 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*, pages 22442–22452, 2025.

[72] Tianqi Wang, Enze Xie, Ruihang Chu, Zhenguo Li, and Ping Luo. Drivecot: Integrating chain-of-thought reasoning with end-to-end driving. *arXiv preprint arXiv:2403.16996*, 2024.

[73] Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. In *AAAI*, pages 10412–10420, 2025.

[74] Xiaotao Hu, Wei Yin, Mingkai Jia, Junyuan Deng, Xiaoyang Guo, Qian Zhang, Xiaoxiao Long, and Ping Tan. Drivingworld: Constructing world model for autonomous driving via video gpt. *arXiv preprint arXiv:2412.19505*, 2024.

[75] Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, Pengxuan Yang, et al. Latent-wam: Latent world action modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2603.24581*, 2026.

[76] 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. *arXiv preprint arXiv:2604.04198*, 2026.

[77] 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. *arXiv preprint arXiv:2603.11041*, 2026.

[78] 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. *arXiv preprint arXiv:2603.01063*, 2026.

[79] 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. *arXiv preprint arXiv:2510.12796*, 2025.

[80] 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*, 2026.

[81] Shuhan Tan, Kashyap Chitta, Yuxiao Chen, Ran Tian, Yurong You, Yan Wang, Wenjie Luo, Yulong Cao, Philipp Krahenbuhl, Marco Pavone, et al. Latent chain-of-thought world modeling for end-to-end driving. *arXiv preprint arXiv:2512.10226*, 2025.

[82] 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.

[83] 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*, pages 27137–27146, 2025.

[84] Ruixun Liu, Lingyu Kong, Derun Li, and Hang Zhao. Occvla: Vision-language-action model with implicit 3d occupancy supervision. *arXiv preprint arXiv:2509.05578*, 2025.

[85] Chenxu Dang, Jie Wang, Guang Li, Zhiwen Hou, Zihan You, Hangjun Ye, Jie Ma, Long Chen, and Yan Wang. Sparseoccvla: Bridging occupancy and vision-language models via sparse queries for unified 4d scene understanding and planning. *arXiv preprint arXiv:2601.06474*, 2026.

[86] Zhe Liu, Runhui Huang, Rui Yang, Siming Yan, Zining Wang, Lu Hou, Di Lin, Xiang Bai, and Hengshuang Zhao. Drivepi: Spatial-aware 4d mllm for unified autonomous driving understanding, perception, prediction and planning. *arXiv preprint arXiv:2512.12799*, 2025.

[87] Hongbin Lin, Yiming Yang, Yifan Zhang, Chaoda Zheng, Jie Feng, Sheng Wang, Zhennan Wang, Shijia Chen, Boyang Wang, Yu Zhang, et al. Futurex: Enhance end-to-end autonomous driving via latent chain-of-thought world model. *arXiv preprint arXiv:2512.11226*, 2025.

[88] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025.

[89] Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Xiang An, Yan Feng, Peng Pei, Xunliang Cai, et al. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. *arXiv preprint arXiv:2510.18632*, 2025.

[90] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *arXiv preprint arXiv:2505.24625*, 2025.

[91] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[92] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020.

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

[94] Yunsong Zhou, Linyan Huang, Qingwen Bu, Jia Zeng, Tianyu Li, Hang Qiu, Hongzi Zhu, Minyi Guo, Yu Qiao, and Hongyang Li. Embodied understanding of driving scenarios. In *ECCV*, pages 129–148, 2024.

[95] Jimuyang Zhang, Zanming Huang, Arijit Ray, and Eshed Ohn-Bar. Feedback-guided autonomous driving. In *CVPR*, pages 15000–15011, 2024.

[96] Zhijian Huang, Tao Tang, Shaoxiang Chen, Sihao Lin, Zequn Jie, Lin Ma, Guangrun Wang, and Xiaodan Liang. Making large language models better planners with reasoning-decision alignment. In *ECCV*, pages 73–90, 2024.

[97] 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.

[98] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *arXiv preprint arXiv:2406.03877*, 2024.

[99] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[100] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *ICCV*, pages 913–922, 2021.

[101] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, pages 180–191, 2022.

[102] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *CoRL*, pages 1475–1485, 2022.

[103] 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.

[104] 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. In *NeurIPS*, pages 6119–6132, 2022.

[105] 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*, pages 7953–7963, 2023.

[106] 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*, pages 24823–24834, 2025.

A Appendix

A.1 Data Curation

This part describes the data curation pipeline used to train AgentWorld. The pipeline is designed around three principles. First, physical agent states are used only as supervision or reward signals, and are never exposed as textual input to the policy. Second, the `<perception>` field only exposes special agent tokens, while the state of each participant is carried by the token hidden state. Third, the final SFT phase and the RL phase share the same MLLM-only input-output template.

Influential agent selection. Since AgentWorld uses a fixed number of agent tokens, we select the most planning-relevant participants for grounding and reasoning. For each sample, we rank annotated objects by an influence score that considers both their current state and future interaction with the ego vehicle. The selected agents are used as grounding targets in Stage 1 and as the source of agent-token references in Stage 2.

Let $\mathcal{T}_i = \{\mathbf{p}_i^{t+\Delta_m}\}_{m=1}^{M_H}$ denote the future trajectory of object i within a horizon of $H = 3$s, and let $\mathcal{T}_{\text{ego}} = \{\mathbf{p}_{\text{ego}}^{t+\Delta_m}\}_{m=1}^{M_H}$ denote the expert ego trajectory. We define the relative future trajectory as

$$\mathbf{r}_i^{t+\Delta_m} = \mathbf{p}_i^{t+\Delta_m} - \mathbf{p}_{\text{ego}}^{t+\Delta_m}. \tag{12}$$

The current influence score measures the relevance of an object based on its current position, lane relation, distance, and closing speed:

$$S_i^{\text{cur}} = w_d s_i^{\text{dist}} + w_l s_i^{\text{lane}} + w_c s_i^{\text{close}}, \tag{13}$$

where s_i^{dist} increases for nearby objects, s_i^{lane} indicates whether the object is in or close to the ego lane, and s_i^{close} measures relative closing speed.

The future influence score measures whether the object is likely to affect the ego plan:

$$S_i^{\text{future}} = w_{\min} s_i^{\min} + w_{\text{ttc}} s_i^{\text{ttc}} + w_{\text{corr}} s_i^{\text{corr}}, \tag{14}$$

where $s_i^{\min}$ is based on the minimum distance between $\mathcal{T}_i$ and $\mathcal{T}_{\text{ego}}$, s_i^{ttc} measures time-to-collision risk, and s_i^{corr} indicates whether the future object trajectory enters the ego corridor.

The final influence score is

$$S_i = \alpha_i \left(\lambda S_i^{\text{cur}} + (1-\lambda) S_i^{\text{future}}\right), \tag{15}$$

where α_i is a category-dependent weight. λ is set to 0.45. We use larger weights for vulnerable road users such as pedestrians and cyclists, since they are safety-critical even when their current distance is not the smallest.

The algorithm is shown in Alg. 1 In our implementation, $K \leq N$, where N is the number of available agent tokens. If fewer than K valid objects exist, the remaining slots are treated as no-object. This selection is used to prioritize agents that are most relevant to planning, rather than exhaustively supervising all annotated objects in crowded scenes.

Stage 1: grounding data. The goal of Stage 1 is to turn the special tokens $\{\langle \text{agent_}i \rangle\}_{i=1}^{N}$ into physically grounded world-state slots. For each scene, we construct a grounding sample with the following visible input shown in Fig. 7:

$$\texttt{<perception>}\langle \text{agent_}1 \rangle \cdots \langle \text{agent_}N \rangle\texttt{</perception>}. \tag{16}$$

No textual description of object category, location, or motion is provided. The model must encode this information in the contextual hidden states of the agent tokens.

We compute influence scores using Alg. 1. Only top-K agents are used. The training targets are the selected influential agents $\mathcal{A}_K$. For each selected agent, we store its category, 3D bounding box, velocity or motion state, and future trajectory. During training, the predictions from the agent-token hidden states are matched to these targets using DETR-style bipartite matching. Matched tokens are supervised with classification, 3D box, and trajectory losses, while unmatched tokens are supervised as no-object. The offline VGGT features are provided as geometry memory for routed anchoring, but the VGGT encoder itself is not trained.



User: You're an autonomous vehicle's brain. Coordinates: X-axis is perpendicular, and Y-axis is parallel to the direction you're facing. You're at point (0,0). Units: meters. Based on the provided multi-view driving scene, initialize agent-centric world states for surrounding traffic participants. Use the following latent agent slots to represent possible agents in the scene. Each active slot should correspond to one physical traffic participant, and unused slots should be treated as no-object.

AgentWorld: <perception><agent_1><agent_2><agent_3><agent_4><agent_5><agent_6><agent_7><agent_8><agent_9><agent_10><agent_11><agent_12><agent_13><agent_14><agent_15><agent_16><agent_17><agent_18><agent_19><agent_20></perception>

Figure 7: Example of the Stage-1 agent grounding QA template. Given multi-view driving observations and a grounding instruction, AgentWorld outputs a fixed set of `<agent_i>` tokens inside the `<perception>` field. These tokens act as latent agent slots: active slots are supervised to align with physical traffic participants through 3D box and future trajectory targets, while unused slots are trained as no-object.

```
PROMPT_TEMPLATE = """
You are the decision-making AI for an autonomous vehicle. You are analyzing a composite multi-view image created by
stitching front-facing camera views with vertically-flipped rear-facing camera views.

Current Speed: {speed} m/s

The top-{top_k} traffic participants affecting ego driving have been detected and ranked by importance:
{agent_list}

Each participant is identified as <agent_i> by its matched agent-slot token. Each entry may include object category, 3D
bbox in ego coordinates, current distance, velocity, future trajectory, ego future trajectory, relative future trajectory,
minimum future distance, and TTC.

Your Task:
Generate a concise real-time driving-risk rationale inside <think></think>. Identify only the key agents that directly
affect ego driving. Focus on path conflict, lane occupancy, lateral clearance, lead-vehicle distance, pedestrians near the
road, and future trajectory proximity.

Important constraints:
1. Output only one <think>...</think> block.
2. Mention only the most relevant agents, usually 1-12 agents.
3. Use special tokens such as <agent_1>, <agent_2>, etc. Do not replace them with object names only.
4. Clearly identify the main risk if one exists.
5. If no critical risk exists, state that no immediate path conflict is predicted.
6. Keep the explanation concise, preferably 1-3 sentences.
7. Do not restate the target action label or say "the decision is to {action}".
8. The response should sound like a brief driving-risk thought, not a full report.

Output format:
<think>...</think>
"""
```

Figure 8: Prompt design for reasoning with agent data generation.

Stage 2: Reasoning-oriented SFT data. Stage 2 teaches the MLLM to use grounded agent tokens for traffic reasoning and planning. Each response follows a compact structured format:

$$y = (\xi, a, \tau), \tag{17}$$

where ξ is a short rationale in `<think>`, a is a high-level action in `<action>`, and τ is the future ego trajectory in `<answer>`.

As shown in Fig. 9, the SFT prompt uses the same `<perception>` field as Stage 1:

$$\texttt{<perception>}\langle \text{agent_1} \rangle \cdots \langle \text{agent_}N \rangle\texttt{</perception>}. \tag{18}$$

To generate the rationale, we use the slot-to-agent assignment from the Stage-1 grounding model. Specifically, we run the grounding model on the training set, match predicted slots to ground-truth agents, and obtain a mapping from $\langle \text{agent_}i \rangle$ to either a physical agent or no-object.

The high-level action is derived from the expert ego trajectory. We classify longitudinal behavior into `ACCELERATE`, `KEEP_SPEED`, `DECELERATE`, or `STOP`, and lateral behavior into `KEEP_LANE`, `CHANGE_LEFT`, or `CHANGE_RIGHT`. The `<answer>` field contains the expert future waypoints.

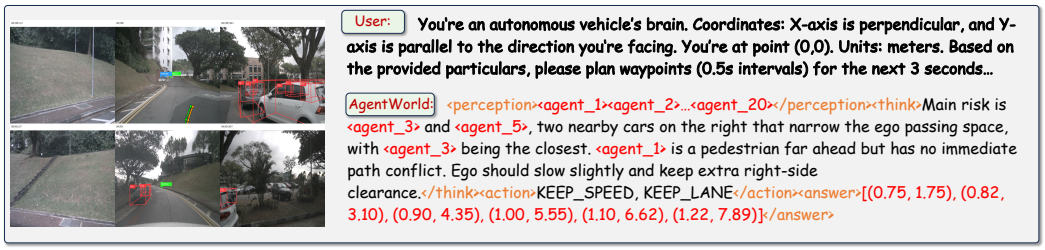

Figure 9: Example of the Stage-2 reasoning-oriented SFT template. Given multi-view driving observations, ego-state instructions, and the fixed `<agent_i>` slots in the `<perception>` field, AgentWorld learns to generate a structured response consisting of concise agent-grounded reasoning in `<think>`, a high-level driving command in `<action>`, and future ego waypoints in `<answer>`. The rationale explicitly references decision-relevant agent tokens, encouraging the policy to plan based on surrounding-agent interactions rather than shortcut cues.

We optionally use strong MLLMs to generate concise, agent-grounded driving-risk rationales. The prompt template is shown in Fig. 8. The annotator model receives multi-view driving images, the current ego speed, and structured information about relevant traffic participants identified by their matched agent-slot tokens, and generates a brief rationale in <think> focusing on the key agents and their implications for driving risk.

Stage 3: RL Data The RL stage uses the same prompt template as the final SFT phase, but removes the supervised assistant response. The model observes images, ego status, navigation command, and the `<perception>` field containing the agent tokens. It then samples multiple structured responses:

$$y_k = (\xi_k, a_k, \tau_k), \quad k = 1, \ldots, K_{\mathrm{RL}}. \tag{19}$$

We use a composite reward that measures format, perception, action, trajectory, and interaction context. This makes the RL stage agent-oriented: to improve reward, the model must generate trajectories that are safe with respect to the dynamic agents represented by the learned tokens.

A.2 Rewards of RL

Format reward. This reward R_{fmt} enforces the hierarchical output protocol of AgentWorld. A structural reward of 1.0 is assigned if the response strictly follows `<perception>...</perception>` `<think>...</think>` `<action>...</action>` `<answer>...</answer>`, and 0 otherwise. The `<perception>` field must contain only valid special agent tokens $\{\langle \mathrm{agent_}i \rangle\}_{i=1}^{N}$, without free-form object descriptions or invalid tokens. The `<action>` field must be parsable into a valid high-level maneuver, and the `<answer>` field must contain a complete sequence of future waypoints with the required horizon. We also encourage the `<think>` tag to contain at least one agent token.

Perception reward. AgentWorld does not require the policy to explicitly output bounding boxes or textual perception descriptions. Instead, perception is exposed through a fixed set of latent agent slots in the `<perception>` field. Therefore, we define a lightweight slot-alignment reward that checks whether the generated perception field contains the correct number of valid agent tokens. Given the predefined token set $\mathcal{T}_{\mathrm{agent}} = \{\langle \mathrm{agent}_1 \rangle, \cdots, \langle \mathrm{agent}_N \rangle\}$ and the parsed token set $\hat{\mathcal{T}}_{\mathrm{agent}}$ from the response, the reward is defined as

$$R_{\mathrm{percep}} = \mathbb{I}\left[|\hat{\mathcal{T}}_{\mathrm{agent}}| = N\right] \cdot \mathbb{I}\left[\hat{\mathcal{T}}_{\mathrm{agent}} = \mathcal{T}_{\mathrm{agent}}\right], \tag{20}$$

where duplicate, missing, or invalid tokens receive zero reward. We additionally require the `<perception>` field to contain only agent tokens and no free-form object descriptions. This reward ensures that the policy preserves the agent-token interface learned during grounding, while leaving the physical content of each slot to be represented by its hidden state rather than by generated text.

Meta-Action reward. We evaluate the high-level driving command predicted in the `<action>` field. The action is represented as a set of discrete maneuver labels, such as `KEEP_LANE`, `CHANGE_LEFT`, `DECELERATE`, or `STOP`. Given the reference action set $\mathcal{M}^{\star}$ derived from the expert trajectory and the

predicted action set $\hat{\mathcal{M}}$, we compute an F1-based reward:

$$R_{\text{action}} = \frac{2 \cdot \text{Prec}(\hat{\mathcal{M}}, \mathcal{M}^\star) \cdot \text{Rec}(\hat{\mathcal{M}}, \mathcal{M}^\star)}{\text{Prec}(\hat{\mathcal{M}}, \mathcal{M}^\star) + \text{Rec}(\hat{\mathcal{M}}, \mathcal{M}^\star) + \epsilon}.$$

This balances precision and recall over multi-label driving intents and prevents degenerate behaviors such as always predicting conservative or straight-driving actions.

Agent interaction reward. We use the frozen agent head as a reward-side probe to decode the top-K active agent trajectories from the learned `<agent_i>` tokens. Given the predicted ego trajectory $\hat{\tau}_{\text{ego}}$ and decoded agent trajectories $\{\hat{\tau}_i\}_{i \in \mathcal{I}_K}$, we compute the minimum future distance

$$d_i^{\min} = \min_m \left\| \hat{\mathbf{p}}_{\text{ego}}^{t+\Delta_m} - \hat{\mathbf{p}}_i^{t+\Delta_m} \right\|_2, \tag{21}$$

and the time-to-collision risk

$$\text{TTC}_i = \min \left\{ \Delta_m \mid \left\| \hat{\mathbf{p}}_{\text{ego}}^{t+\Delta_m} - \hat{\mathbf{p}}_i^{t+\Delta_m} \right\|_2 < d_{\text{safe}} \right\}. \tag{22}$$

If no unsafe distance occurs within the horizon, we set $\text{TTC}_i = \infty$. The agent reward penalizes collision, insufficient clearance, and short TTC:

$$R_{\text{agent}} = -\frac{1}{K} \sum_{i \in \mathcal{I}_K} \left(\lambda_{\text{col}} \mathbb{I}[d_i^{\min} < d_{\text{col}}] + \lambda_{\text{clear}} \left[1 - \frac{d_i^{\min}}{d_{\text{safe}}} \right]_+ + \lambda_{\text{ttc}} \left[1 - \frac{\text{TTC}_i}{T_{\text{safe}}} \right]_+ \right). \tag{23}$$

Overall reward. The final reward is a weighted sum:

$$R = \lambda_{\text{fmt}} R_{\text{fmt}} + \lambda_{\text{percep}} R_{\text{percep}} + \lambda_{\text{action}} R_{\text{action}} + \lambda_{\text{traj}} R_{\text{traj}} + \lambda_{\text{agent}} R_{\text{agent}}. \tag{24}$$

R_{fmt} and R_{percep} enforce valid structured generation and correct use of the fixed `<agent_i>` slots. R_{action} and R_{traj} evaluate maneuver correctness and waypoint quality. R_{agent} further makes the reward interaction-aware by penalizing collisions, unsafe clearance, and short TTC between the predicted ego trajectory and the top-K agent trajectories decoded from the learned agent tokens by a frozen agent head. Thus, the reward encourages AgentWorld to preserve its agent-token reasoning interface while optimizing safe driving behavior.

A.3 Training Details

Our training follows three stages: agent grounding SFT, reasoning-oriented SFT, and agent-oriented reinforcement learning. The offline VGGT features are pre-extracted.

Stage 1: Physical grounding SFT. We first train the agent-token grounding interface for 12 epochs. The trainable modules include the MLLM backbone, the special `<agent_i>` token embeddings, the multimodal routed anchoring module, and the auxiliary agent head. We use AdamW with a cosine decay schedule and a warmup ratio of 0.03. The learning rate is set to 1.0×10^{-4}. The per-device batch size is 2 with 8 gradient accumulation steps, giving an effective batch size of 16 per device. Input images are processed with a maximum resolution of 524288 pixels. For multimodal routed anchoring, we use $L = 3$ routing layers by default. The grounding loss supervises category/existence, 3D bounding box, and future trajectory prediction after Hungarian matching. The number of agents is set to 20.

Stage 2: Reasoning-oriented SFT. Starting from the Stage-1 checkpoint, we train the MLLM to use grounded agent tokens for reasoning and planning for 12 epochs. The multimodal routed anchoring module and auxiliary agent head are frozen. The trainable parameters are the MLLM policy parameters and the special agent-token embeddings. We use AdamW with a learning rate of 5.0×10^{-5} and a cosine decay schedule. The batch configuration follows Stage 1, with a per-device batch size of 2 and 8 gradient accumulation steps. Each sample is trained with the structured output format consisting of `<perception>`, `<think>`, `<action>`, and `<answer>`. To reduce mismatch with the RL stage, we progressively drop the training-time grounding scaffold during SFT: early batches use full grounding inputs, middle batches use partial dropout, and the final SFT phase uses the same MLLM-only format as RL.

Stage 3: Agent-oriented RL. We further optimize the Stage-2 checkpoint with GRPO for 1 epoch. During RL, only the MLLM policy parameters are updated. The auxiliary agent head is kept

frozen and used only as a reward-side generator to decode top-K agent trajectories from the learned <agent_i> tokens; it receives no gradients. The routed anchoring modules are also frozen and are not optimized during RL. We use a learning rate of 1.0×10^{-6} and a global batch size of 16. For each prompt, we sample 8 responses to estimate group-relative advantages. The KL coefficient to the SFT reference policy is set to 1.0×10^{-2}. The reward is a weighted sum of format reward, perception-slot reward, action reward, trajectory reward, and agent interaction reward, with all reward weights set to 1 unless otherwise specified.

After grounding and SFT, the agent tokens have already been aligned with physical participants and learned as the planning interface of the MLLM. Therefore, in the RL stage, we no longer optimize the auxiliary grounding modules. Starting from the SFT checkpoint, GRPO updates only the MLLM policy, which samples multiple structured responses for each scene, including a rationale, a high-level action, and future waypoints. The auxiliary agent head is kept frozen only as a reward-side decoder. It decodes the hidden states of agent tokens into physical quantities, such as agent existence, 3D boxes, and short-horizon motions, which are then used to compute candidate-dependent interaction rewards for the sampled ego trajectories. This frozen decoder receives no gradient and is not part of the policy forward path. At inference time, it is completely removed, and AgentWorld performs planning with the MLLM and the learned agent-token interface alone.

Training objectives. AgentWorld is optimized in three stages. First, we perform agent grounding by supervising the hidden state $\mathbf{h}_t^i$ at each agent-token position with a lightweight head that predicts category/existence, 3D state, and future trajectory. Because the token order is unordered, predicted slots are assigned to ground-truth agents with DETR-style bipartite matching. The grounding loss is

$$\mathcal{L}_{\text{ground}} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} + \lambda_{\text{noobj}}\mathcal{L}_{\text{noobj}}. \tag{25}$$

We set $\lambda_{\text{cls}} = 1.0$, $\lambda_{\text{box}} = 1.0$, $\lambda_{\text{traj}} = 1.0$, and $\lambda_{\text{noobj}} = 0.1$ by default. The no-object weight is kept small to avoid dominating the supervision when most agent slots are empty. Matched slots are grounded to physical agents, while unmatched slots are trained as no-object.

Second, we conduct reasoning-oriented SFT. Each expert response consists of a concise rationale, a high-level action, and future waypoints, denoted as $y^\star = (\xi^\star, a^\star, \tau^\star)$. The policy is trained with the autoregressive objective

$$\mathcal{L}_{\text{SFT}} = -\sum_k \log p_\theta \left(y_k^\star \mid y_{<k}^\star, \mathcal{H}_t, \mathcal{S}_t^{\text{agent}}\right), \quad \mathcal{S}_t^{\text{agent}} = \{\mathbf{h}_t^i\}_{i=1}^N. \tag{26}$$

This stage teaches the MLLM to use grounded agent tokens for traffic saliency analysis, causal reasoning, and trajectory planning. When auxiliary grounding modules are retained in the early SFT phase, we optionally add a small grounding regularizer:

$$\mathcal{L}_{\text{Stage2}} = \mathcal{L}_{\text{SFT}} + \eta\mathcal{L}_{\text{ground}}, \tag{27}$$

where η is linearly decayed from 0.1 to 0 as the training gradually shifts to the MLLM-only format. In the final SFT phase, $\eta = 0$. Finally, we remove the training-time grounding scaffold and optimize the MLLM-only policy with GRPO.

A.4 Comparison with Different Latent VLA World Models

Recent latent VLA world models reduce the cost of future imagination by moving from pixel-space generation to latent-space reasoning. For example, Latent-CoT-style driving models [81] replace verbose textual reasoning with latent world-model tokens, DriveWorld-VLA [33] unifies world modeling and planning in a shared latent space, and LaST-VLA [82] performs reasoning in a latent spatio-temporal space with geometric and dynamic alignment. These methods improve efficiency over explicit future-frame generation, but their world states remain largely image-, BEV-, occupancy-, or spatio-temporal-latent representations. Such latents can still preserve fine-grained scene factors that are weakly related to planning, including appearance, background layout, static context, dense occupancy patterns, and generic scene evolution, while leaving the agent-level causal variables behind driving decisions implicit. In contrast, AgentWorld changes the unit of world modeling from dense scene evolution to agent evolution. Rather than compressing the future into a holistic latent state, AgentWorld represents the driving world as a set of grounded and addressable <agent_i> tokens, where each token is supervised to correspond to a surrounding traffic participant or a no-object state and encodes semantic identity, 3D geometry, motion tendency, and interaction context. This

agent-centric world state filters out much of the planning-irrelevant nuisance information retained by dense latent world models and exposes the physical entities and interactions that directly govern ego planning.

A.5 Closed-Loop Evaluation Metrics

We evaluate closed-loop driving performance following the standard protocol in Bench2Drive [98]. Unlike open-loop trajectory evaluation, closed-loop simulation measures how sequential decisions affect future observations and safety outcomes. We report four metrics: success rate, driving score, efficiency, and comfortness.

Success Rate (SR). Success rate measures the fraction of episodes in which the ego vehicle reaches the target without safety-critical infractions:

$$\mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(S_i), \tag{28}$$

where N is the number of evaluated episodes and $\mathbb{I}(S_i)$ indicates whether episode i is successfully completed.

Driving Score (DS). Driving score combines route completion and safety penalties:

$$\mathrm{DS} = \frac{1}{N}\sum_{i=1}^{N} R_i \prod_{k} P_{i,k}, \tag{29}$$

where $R_i \in [0,1]$ is the route completion ratio, and $P_{i,k} \in (0,1]$ is the penalty multiplier for the k-th infraction, such as collision, red-light violation, or off-road driving.

Efficiency. Efficiency measures the ego vehicle's speed relative to surrounding traffic at route checkpoints. We use the official Bench2Drive [98] implementation to aggregate the speed percentages across checkpoints and routes. Higher values indicate greater efficiency, and values above 100 indicate speeds exceeding those of the reference traffic.

Comfortness. Comfortness measures the proportion of trajectory segments satisfying the benchmark's kinematic constraints on acceleration, jerk, yaw rate, and yaw acceleration. We use the official Bench2Drive [98] implementation and report the resulting score as a percentage. Higher values indicate smoother driving.

A.6 The Theoretical Motivation of AgentWorld

We provide a theoretical perspective on why agent-centric latent states are more suitable than pixel-level future frames for planning-oriented VLA world models. Let $\mathcal{H}_t$ denote the driving context and let τ be a future ego trajectory. A pixel-level world model predicts future visual observations $\hat{\mathcal{I}}_{t+1:t+H}$, whereas AgentWorld predicts an agent-centric latent state $\hat{\mathcal{A}}_{t+1:t+H}$. Each agent state encodes semantic identity, 3D geometry, motion state, and interaction context.

We view future images as a composition of two parts:

$$\mathcal{I}_{t+1:t+H} = g(\mathcal{A}_{t+1:t+H}, \mathcal{U}_{t+1:t+H}), \tag{30}$$

where $\mathcal{A}$ denotes the physical and interactive evolution of traffic participants, and $\mathcal{U}$ denotes nuisance visual factors such as texture, illumination, background, and visual continuity. For planning, the cost of an ego trajectory is determined by the evolution of traffic agents rather than by all pixel-level details. We therefore assume that

$$C(\tau, \mathcal{S}) = C(\tau, \mathcal{A}), \tag{31}$$

where $\mathcal{S}$ is the full future scene state and $\mathcal{A}$ is the agent-centric latent state.

Under this assumption, $\mathcal{A}$ serves as a sufficient statistic for planning. For any policy conditioned on future images, we can construct an agent-conditioned policy by marginalizing out nuisance visual factors:

$$\pi_{\mathcal{A}}(\tau \mid \mathcal{H}_t, \mathcal{A}) = \mathbb{E}_{\mathcal{U}|\mathcal{A}}\left[\pi_{\mathcal{I}}(\tau \mid \mathcal{H}_t, g(\mathcal{A}, \mathcal{U}))\right]. \tag{32}$$

Algorithm 1 Top-K Influential Agent Selection

Require: Current sample annotations, ego pose, future horizon $H = 3$s, number of selected agents K

Ensure: Top-K influential agents $\mathcal{A}_K$

for each object i in the current sample **do**
  Transform the object box, velocity, and category into the current ego frame.
  Extract the future object trajectory $\mathcal{T}_i$ from the annotation chain.
  Extract the future ego trajectory $\mathcal{T}_{\text{ego}}$.
  Compute the relative future trajectory $\mathbf{r}_i^{t+\Delta_m}$.
  Compute current influence score S_i^{cur}.
  Compute future trajectory influence score S_i^{future}.
  Compute final score $S_i = \alpha_i(\lambda S_i^{\text{cur}} + (1-\lambda) S_i^{\text{future}})$.
end for
Sort all objects by S_i in descending order.
Select the top-K objects as $\mathcal{A}_K$.
return $\mathcal{A}_K$

Since the planning cost depends on $\mathcal{A}$ but not on $\mathcal{U}$, conditioning the policy on nuisance visual details does not provide additional decision-relevant information. This does not imply that images are unnecessary; visual evidence remains essential for grounding and estimating $\mathcal{A}$. Rather, it shows that once the agent-centric state is available, reasoning over dense generated pixels is an indirect and inefficient interface for planning.

We further show that planning performance is directly tied to the quality of the predicted agent-centric state. Assume the planning cost is L-Lipschitz with respect to the agent-centric state:

$$|C(\tau, \mathcal{A}) - C(\tau, \hat{\mathcal{A}})| \le L d(\mathcal{A}, \hat{\mathcal{A}}), \tag{33}$$

where $d(\cdot, \cdot)$ measures the prediction error of the agent-centric state. Let

$$\hat{\tau} = \arg\min_{\tau} C(\tau, \hat{\mathcal{A}}), \quad \tau^\star = \arg\min_{\tau} C(\tau, \mathcal{A}). \tag{34}$$

Then the regret of planning with the predicted agent-centric state is bounded by

$$C(\hat{\tau}, \mathcal{A}) - C(\tau^\star, \mathcal{A}) \le 2L d(\mathcal{A}, \hat{\mathcal{A}}). \tag{35}$$

This bound shows that reducing the agent-state prediction error directly reduces planning regret. In contrast, a low pixel reconstruction loss does not necessarily imply a low agent-state error, because the pixel objective also allocates representation capacity to nuisance factors $\mathcal{U}$. AgentWorld therefore provides a more task-aligned surrogate for planning by explicitly grounding latent tokens to 3D geometry, motion dynamics, and interactions.

The computational benefit also follows from the compactness of the state. If each future frame contains P visual tokens, a pixel-level world model with horizon H and K iterative generation steps processes $O(KHP)$ visual tokens. AgentWorld instead uses N agent tokens per step, resulting in $O(HN)$ world-state tokens, where $N \ll P$. When self-attention over generated world tokens dominates the computation, the complexity scales as

$$O\big(K(HP)^2\big) \quad \text{versus} \quad O\big((HN)^2\big), \tag{36}$$

leading to an approximate ratio of

$$O\left(K\left(\frac{P}{N}\right)^2\right). \tag{37}$$

Thus, AgentWorld concentrates computation on decision-relevant traffic participants rather than dense visual details, yielding a compact, structured, and planning-oriented world state for VLA-based autonomous driving.

A.7 More Experimental Results

Ablation on agent head in SFT and RL stages. Tab. 10 studies how the auxiliary agent head affects different training stages. Removing the agent head during SFT leads to the weakest performance,

Table 10: Ablation on agent head in SFT and RL stages.

Method	Avg. L2 (m) ↓
SFT w/o head	0.31
RL w head	0.26
AgentWorld	0.27

Table 11: Ablation on different model variants.

Method	Avg. L2 (m) ↓
Qwen2-VL-2B	0.27
Qwen2-VL-7B	0.25
Qwen2.5-VL-3B	0.27

Table 12: Ablation on reward of RL stage.

Method	Avg. L2 (m) ↓
w/o R_{action}	0.29
w/o R_{agent}	0.30
w/o R_{traj}	0.32

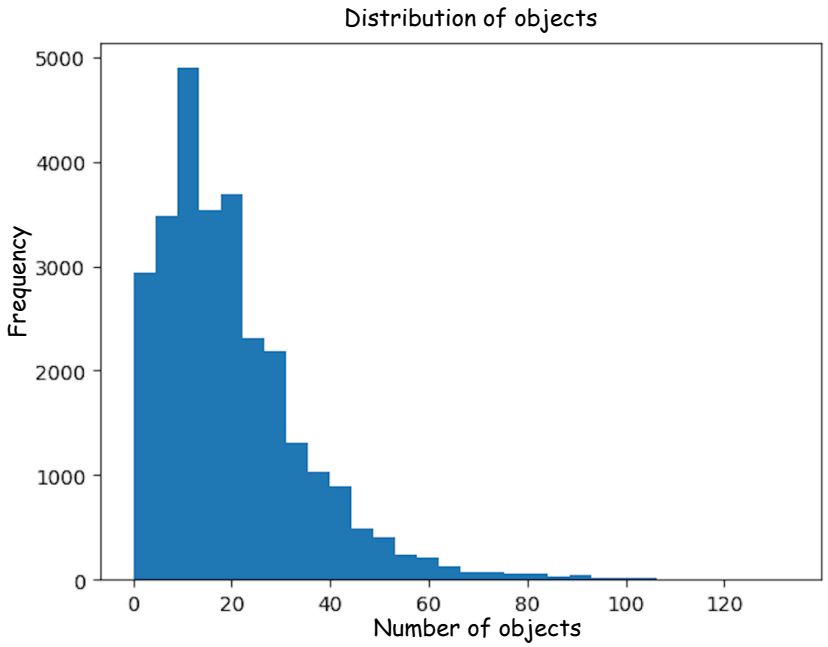

Figure 10: Distribution of object counts.

showing that explicit physical grounding is important for turning special agent tokens into usable world-state representations. Without this grounding signal, the model can still learn the response format, but the agent tokens are less reliably tied to surrounding traffic participants, which weakens downstream planning.

The *RL w head* variant achieves an Avg. L2 error of 0.26 m, compared with 0.27 m for the default AgentWorld model. In the default model, the auxiliary agent head remains frozen during RL and is used only as a reward-side decoder to compute agent-interaction rewards; only the MLLM policy parameters are updated through GRPO. At inference time, the auxiliary head is discarded, and the policy generates reasoning and ego waypoints using the learned agent-token representations. Thus, the auxiliary head supports physical grounding and reward computation during training without remaining part of the inference-time policy interface.

Ablation on different model variants. Tab. 11 compares different MLLM backbones. The 7B variant achieves the best planning accuracy, indicating that stronger language-vision reasoning capacity improves the model's ability to use grounded agent tokens for decision-making. This is consistent with the role of the MLLM in AgentWorld: after agent tokens provide a compact physical world-state interface, the backbone still needs sufficient reasoning capacity to identify salient agents, infer interactions, and translate them into safe trajectories. The smaller variants remain competitive, suggesting that AgentWorld is not tied to a specific backbone. The results indicate that the proposed agent-token interface is broadly compatible across MLLM scales, while larger models can further exploit this structured interface for better planning.

Ablation on reward in RL stage. Tab. 12 evaluates the contribution of different reward terms in the RL stage. Removing the trajectory reward causes the largest degradation, showing that waypoint-level



Figure 11: Visualization of trajectories across VLA-World and our AgentWorld on the nuScenes validation dataset. The predicted and ground-truth trajectories are depicted in red and green, respectively.

feedback is the most direct signal for improving planning accuracy. This reward guides the model toward trajectories that better match expert behavior and satisfy geometric constraints. Removing the agent-aware reward also hurts performance, confirming that interaction-level feedback is necessary for making RL consistent with the AgentWorld design. Without this term, the policy can optimize trajectory imitation while paying less attention to surrounding-agent safety, such as clearance and conflict risk. Removing the action reward produces a smaller but still clear degradation, indicating that high-level driving commands provide useful semantic guidance that stabilizes the policy before generating low-level waypoints.

Overall, the reward ablation shows that effective RL requires both trajectory-level optimization and agent-centric interaction feedback: the former improves waypoint accuracy, while the latter encourages the policy to preserve the learned agent-token reasoning interface.

Distribution of object counts. Fig. 10 motivates the choice of the maximum agent-token number N=20. The object-count distribution is highly imbalanced: most scenes contain a moderate number of objects, while only a small fraction of samples fall into the long-tail high-density regime. This suggests that using an excessively large number of agent tokens would introduce many unused or low-relevance slots in typical scenes, increasing the reasoning cost and no-object supervision burden without proportional benefit.

Choosing N=20 provides a practical trade-off between coverage and efficiency. It covers the high-frequency region of the distribution, while the remaining crowded scenes can still be handled by the Top-K influential-agent selection strategy. Importantly, AgentWorld does not aim to exhaustively represent every annotated object; instead, it prioritizes traffic participants that are most relevant to planning, such as nearby vehicles, cut-in agents, pedestrians, and agents whose future motion may conflict with the ego trajectory. Therefore, N=20 is sufficient to preserve the key planning-relevant agents in most scenarios while keeping the agent-token world state compact.

This design also supports the motivation of AgentWorld: world modeling for driving should allocate capacity to decision-relevant agents rather than dense scene reconstruction or all visible objects. The long-tail distribution further justifies using fixed agent slots with no-object handling and influential-agent selection, rather than scaling the token count with the total number of objects.

Qualitative trajectory visualization. Fig. 11 compares the predicted trajectories of VLA-World and AgentWorld against GT across representative nuScenes validation scenes. The qualitative results show that AgentWorld produces trajectories that more closely follow the ground-truth driving intent, especially in scenes requiring subtle curvature, lane following, or cautious adjustment around surrounding traffic participants. In contrast, VLA-World shows larger deviations in several cases. This suggests that visual future imagination alone may not provide a sufficiently structured interface for planning: although future frames can be visually plausible, the policy may still fail to extract the physical and interaction cues needed to align with the expert trajectory. AgentWorld alleviates this issue by grounding the scene into agent-level tokens, allowing the policy to reason over surrounding participants directly rather than relying on dense visual features or ego-motion shortcuts. These visualizations support the quantitative findings: AgentWorld improves planning not merely by adding more information, but by organizing scene information into a more actionable world state. The closer

alignment between predicted and ground-truth trajectories indicates that agent-centric representations provide a stronger interface for translating perception and prediction into driving actions.

A.8 Broader Impacts

AgentWorld aims to improve autonomous driving systems by representing dynamic scenes through compact, agent-level world states. By explicitly grounding traffic participants in 3D geometry, motion, and interaction context, the proposed framework has the potential to improve the transparency and reliability of planning-oriented reasoning, especially in complex multi-agent driving scenarios. Such representations may also facilitate diagnostic analysis, since the learned agent tokens provide an interpretable interface for examining which surrounding agents influence the ego vehicle's decisions.

At the same time, autonomous driving is a safety-critical domain, and improvements in open-loop or closed-loop benchmarks should not be interpreted as sufficient evidence for real-world deployment. Failure cases may arise from imperfect perception, long-tail traffic behaviors, distribution shifts, sensor degradation, or ambiguous human intentions. In addition, agent-centric abstraction may underrepresent rare objects, vulnerable road users, or context factors that are not well captured by the predefined agent tokens. Therefore, AgentWorld should be used as a research step toward more structured and interpretable world modeling, rather than as a standalone deployable driving system. Practical deployment would require rigorous validation under diverse geographic, environmental, and social conditions, as well as careful integration with redundancy, uncertainty estimation, rule compliance, and human oversight mechanisms.

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 claim in the abstract and introduction that this paper studies advancing world vision-language-action models via agentic world states.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are addressd in Section 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: [N/A]

Justification: This work does not include theoretical results or proofs, similar to previous VLA works.

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 provide the implementation details in the Appendix.

   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: The data and code will be publicly available.

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: We provide the training details in Section 4 and Appendix.

   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: [N/A]

   Justification: We set a fixed seed to ensure the reproduction of results.

   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: We provide the compute resources in Section 4.

   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: We conform to the NeurIPS code of Ethics in every respect.

   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: We discuss the broader impacts of this work in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [Yes]

    Justification: To enhance safety, we suggest incorporating fail-safe mechanisms to halt driving under uncertainty. Human oversight and deployment in controlled environments are also recommended to reduce potential misuse or unintended behaviors.

    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: We properly provide this information in our paper and codes.

    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: [Yes]

    Justification: The assets introduced in the paper are well organized.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor 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: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with 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: LLMs were used as the core policy for agent-grounded reasoning and trajectory planning, as described in Section 3. LLMs were also used for paper polishing.

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.