MindDrive: A Vision-Language-Action Model for Autonomous Driving via Online Reinforcement Learning

Haoyu Fu[1*], Diankun Zhang[2*], Zongchuang Zhao[1], Jianfeng Cui[2], Hongwei Xie[2†], Bing Wang[2], Guang Chen[2], Hangjun Ye[2], Dingkang Liang[1✉], Xiang Bai[1]
{hyfu, zcuangzhao, dkliang}@hust.edu.cn
https://xiaomi-mlab.github.io/MindDrive/

[1] Huazhong University of Science and Technology, China
[2] Xiaomi EV, China

Abstract. Current Vision-Language-Action (VLA) paradigms in autonomous driving primarily rely on Imitation Learning (IL), which introduces inherent challenges such as distribution shift and causal confusion. Online Reinforcement Learning offers a promising pathway to address these issues through trial-and-error learning. However, applying online reinforcement learning to VLA models in autonomous driving is hindered by inefficient exploration in continuous action spaces. To overcome this limitation, we propose MindDrive, a VLA framework comprising a large language model (LLM) with two distinct sets of LoRA parameters. The one LLM serves as a Decision Expert for scenario reasoning and driving decision-making, while the other acts as an Action Expert that dynamically maps linguistic decisions into feasible trajectories. By feeding trajectory-level rewards back into the reasoning space, MindDrive enables trial-and-error learning over a finite set of discrete linguistic driving decisions, instead of operating directly in a continuous action space. This approach effectively balances optimal decision-making in complex scenarios, human-like driving behavior, and efficient exploration in online reinforcement learning. Extensive experiments validate the efficacy of our online reinforcement learning framework, which outperforms state-of-the-art IL and offline Reinforcement Learning methods on the challenging Bench2Drive benchmark. To the best of our knowledge, this is the first work to demonstrate the effective application of online reinforcement learning to VLA models in autonomous driving.

1 Introduction

Autonomous driving relies on the model's ability to perceive, make decisions, and act in dynamic complex environments. Traditional end-to-end autonomous driving frameworks [16, 28, 75] integrate perception, prediction, and planning

* Equal contribution. † Project leader. ✉ Corresponding author. Work done when Haoyu Fu was an intern at Xiaomi EV.

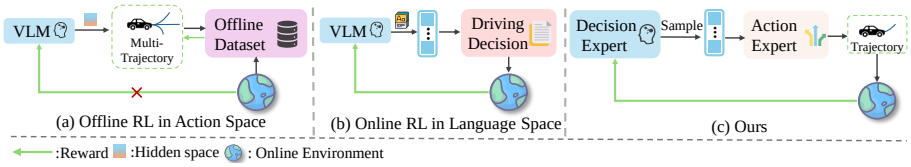

Fig. 1: The comparison of different VLA for reinforcement learning paradigms. The proposed Decision Expert and Action Expert share the same base LLM with distinct LoRA adapters.

modules, but they lack common sense and causal reasoning abilities. With the improved visual understanding and reasoning ability of Vision Language Models (VLMs) [1, 2, 51], many research [11, 45, 58] attempts to apply the Vision-Language-Action (VLA) paradigm to the field of end-to-end autonomous driving. The VLA paradigm in autonomous driving aims to translate complex traffic scene understanding into trajectories of the ego vehicle.

Current VLA models [11,20,59] are primarily trained using imitation learning (IL), which aims to fit expert behaviors from collected driving data. However, exclusive reliance on the IL paradigm renders models susceptible to causal confusion and distributional shift [7,46], resulting in irreversible error accumulation in closed-loop driving scenarios. Reinforcement learning offers a perspective for addressing these challenges through trial-and-error and has achieved notable success in enhancing the causal reasoning capabilities of VLMs [14, 41, 44].

Unlike reinforcement learning in VLMs' discrete language space, the action space of autonomous driving is a continuous trajectory space. Applications of reinforcement learning in the current VLA domain for autonomous driving can be categorized into two main paradigms: offline reinforcement learning for the action space and online reinforcement learning for the language space. Offline reinforcement learning is typically trained on fixed datasets composed of expert demonstrations, as shown in Fig. 1 (a). These methods [27, 33, 64, 74] employ offline reinforcement learning with distinct reward functions [6] to generate more feasible trajectories in the action space. Although they made great progress, offline reinforcement learning limits the VLA model's ability to explore the environment through interaction. Additionally, trajectory optimization in reinforcement learning cannot effectively improve VLMs' reasoning ability. To overcome these limitations, some other methods [19, 29, 68] attempt to employ online reinforcement learning in the language space, as shown in Fig. 1 (b). These methods formulate autonomous driving tasks as VQA tasks and treat driving decisions as actions, thereby facilitating deeper causal reasoning via online reinforcement learning. However, they struggle to map driving decisions to specific, human-like driving trajectories. Therefore, utilizing online reinforcement learning to enhance the performance of autonomous driving VLA requires further exploration.

To address these challenges, we propose a novel architecture, MindDrive, a VLA model for autonomous driving via online reinforcement learning, as shown in Fig. 1 (c). MindDrive transforms the action space from continuous trajecto-

ries into discrete language-based decisions by dynamically mapping, significantly improving exploration efficiency while using trajectory rewards to reinforce the model's reasoning in online reinforcement learning. Specifically, MindDrive consists of two experts initialized from the same LLM and differing only in their respective LoRA adapters [15]. One LLM acts as a Decision Expert, responsible for making reasonable decisions based on the current scenario, while the other serves as an Action Expert, establishing a dynamic mapping from the reasoning result to continuous trajectories. MindDrive first leverages IL to establish a one-to-one correspondence between the meta-actions inferred by the Decision Expert and the multi-modal trajectories output by the Action Expert. The high-quality driving trajectories generated by the Action Expert provide reasonable, human-style candidates for online reinforcement learning. Subsequently, we leverage action trajectory feedback within an online reinforcement learning framework to refine the Decision Expert, allowing it to learn optimal policies by sampling diverse trajectories and receiving corresponding rewards from the interactive environment.

Meanwhile, to enable model exploration and training in a dynamic, interactive environment, we introduce an online closed-loop reinforcement learning framework for autonomous driving VLA models, built upon the CARLA simulator [8]. We define explicit signals for task success and failure, and partition the online reinforcement learning process into data-collection and training phases. During collection, we compute and cache scene tokens for each frame, which serve as compact state representations. This pre-computation step reduces memory buffer overhead, enables large-batch training, and allows the entire process to be formulated as a standard Markov Decision Process [43].

We evaluate the driving ability of MindDrive on the comprehensive and challenging closed-loop benchmark, Bench2Drive [25]. Extensive experiments demonstrate that our framework leads to more effective driving behavior in complex driving scenarios. Employing a lightweight 0.5B LLM [60], MindDrive outperforms the strong baseline [11] with the same parameter size by 5.15% in Driving Score (DS) and 9.26% in Success Rate (SR), respectively. Furthermore, with the 3B LLM [61], MindDrive achieves 80.59 DS and 58.26% SR, significantly boosting performance and setting a new state-of-the-art (SOTA) on the benchmark.

The main contributions of this paper are as follows:

- We propose MindDrive, an online reinforcement learning framework for VLA autonomous driving models. By introducing a dynamic language-action mapping, MindDrive significantly improves exploration efficiency and enables trajectory-level action rewards to facilitate reasoning optimization.
- We introduce an online reinforcement learning scheme. To the best of our knowledge, MindDrive is the first VLA-based autonomous driving model trained with online reinforcement learning in a simulator, aiming to bring new inspiration to the autonomous driving community.
- Extensive experiments demonstrate the effectiveness of MindDrive, achieving 80.59 DS and 58.26% SR on Bench2Drive benchmark, significantly surpassing SOTA IL and reinforcement learning baselines at the same scale.

2 Related work

2.1 End-to-End Autonomous Driving (AD)

End-to-end (E2E) autonomous driving (AD) based on imitation learning [16, 23, 28, 42] integrates perception, prediction, and planning into a differentiable framework [5, 37, 40, 65]. These models directly map sensor inputs to trajectories or control commands from expert data. Representative methods such as UniAD [16] and VAD [28] adopt dense bird's-eye view (BEV) representations to unify multiple tasks. Generative approaches, including GenAD [70] and DiffusionDrive [35], employ diffusion-based planning for multi-modal trajectory generation. DiffAD [53] rasterizes heterogeneous driving targets into a unified BEV space for joint optimization. SparseAD [75] and DriveTransformer [26] improve efficiency with sparse queries.

Despite strong open-loop results, E2E models often fail in closed-loop due to distribution shift and causal confusion [4, 25]. To overcome these limitations, recent studies have explored integrating vision-language models (VLMs) for AD, leveraging their strong generalization, reasoning, and world knowledge [13, 55]. HERMES [71, 72] proposes a unified model of the AD world that integrates 3D scene understanding and future generation for the first time. DriveGPT4 [59], ELM [73], and DriveMM [18] incorporate VLMs for perception, prediction, and decision-making. OmniDrive [52], EMMA [20], and OpenEMMA [56] represent trajectories as text with chain-of-thought reasoning [54]. AutoVLA [74] discretizes actions for alignment with language models but struggles with continuous motion. SimLingo [45] proposes the action dreaming task to achieve alignment between language instructions and driving actions.

Vision-language-action (VLA) models from embodied AI, such as OpenVLA [30] and the π series [3,21], integrate VLMs into continuous action spaces [9]. Inspired by these advancements, ORION [11] bridges the language between action spaces via a generative planner for continuous trajectories, improving reasoning and generalization in autonomous driving scenarios. DrivePI [38] and DriveMonkey [69] use VLMs that serves as a unified VLA framework to jointly perform spatial understanding, 3D perception, prediction, and planning for AD. However, imitation learning is still limited in effectively propagating trajectory-level supervision into the high-level reasoning space of language models. MindDrive advances this line by integrating online reinforcement learning for adaptive interaction and robust decision-making.

2.2 Reinforcement Learning for AD

Reinforcement learning (RL) has achieved remarkable success across diverse domains, ranging from large language models (LLMs) [14] to complex games [34], and has recently been increasingly explored in AD [39]. CarPlanner [66] leverages expert-guided reward to stabilize training, while RAD [12] and R2SE [36] enhance closed-loop training and mitigate catastrophic forgetting. Raw2Drive [63] further introduces a dual-stream model-based RL framework to align raw and

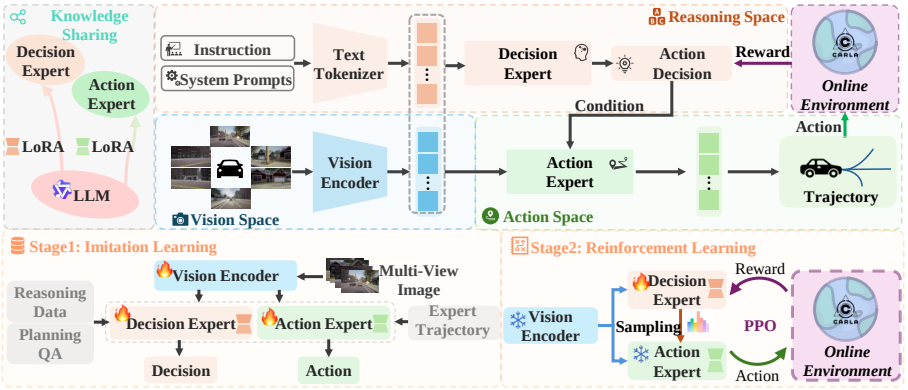

Fig. 2: Overview of the MindDrive framework. The system features two experts (Decision and Action) that share a base LLM but employ distinct LoRA adapters. The Decision Expert infers meta-actions from scene, while the Action Expert maps these actions to concrete trajectories. MindDrive is initialized via imitation learning and refined by online closed-loop reinforcement learning to enhance reasoning capabilities.

privileged world models. Recent works such as AlphaDrive [29], AutoVLA [74], and RecogDrive [33] employ GRPO-based reinforcement learning [14] to refine planning policies through carefully designed reward mechanisms. However, these methods primarily rely on offline expert datasets and handcrafted rewards, lacking actual interaction with dynamic environment. In contrast, online RL enables adaptive decision-making via direct environment interaction and trial-and-error learning but faces slow convergence and inefficient exploration. To address these challenges, we propose an adaptive online RL framework that improves closed-loop robustness and enhances decision-making quality.

3 Method

In this section, we present MindDrive in detail. As illustrated in Fig. 2, MindDrive comprises two main components: a Decision Expert and an Action Expert, which share a common Vision Encoder and Text Tokenizer but differ only in their respective LoRA parameters [15]. The Decision Expert performs high-level reasoning from navigation instructions and multi-view visual inputs, generating abstract driving decisions in the form of meta-actions. The Action Expert translates these meta-actions into concrete action trajectories, conditioned on the scene information and instruction. This design enables flexible and interpretable action generation, bridging high-level reasoning with low-level control.

Our training process includes two stages: 1) Imitation learning (IL) establishes a mapping between language and action space (Sec. 3.2), providing high-quality candidate trajectories for online Reinforcement learning (RL) and effectively reducing its exploration space. 2) Online RL further enhances the model's comprehension ability via action rewards in an online environment (Sec. 3.3).

3.1 Problem Formulation

In the task of end-to-end autonomous driving, we aim to generate a diverse trajectory set A and determine an optimal trajectory a^* based on surrounding visual information V and language instructions L:

$$\begin{aligned} A &= \{a_i\}_{i=1}^{N} \quad \text{, where} \quad a_i \sim \pi_g(a \mid V, L), \\ a^* &= \arg\max_{a \in A} \pi_g(a \mid V, L), \end{aligned} \tag{1}$$

where a_i represents the trajectory in the multimodal trajectory set A and $\pi_g(a \mid V, L)$ is the trajectory generation policy function. Current methods often select the optimal trajectory based on the score-based selection policy π_c:

$$a^* = \arg\max_{a \in A} \pi_c(a). \tag{2}$$

Unlike the score-based selection policy, to fully leverage the potential of the VLA model, we model the selection task as an action decision process and introduce $\pi_d(a|V, L)$ as the selection policy function:

$$\begin{aligned} a^* &= \arg\max_{a \in A} \pi_c(a) \\ &= \arg\max_{a \in A} \underbrace{\pi_d(a \mid V, L)}_{selection} \cdot \underbrace{\pi_g(a \mid V, L)}_{generation}. \end{aligned} \tag{3}$$

From Eq. 3, we can clearly identify that the generation of the optimal trajectory depends on two core policy functions: π_d and π_g. Previous methods fail to establish a connection between these two policy spaces in online reinforcement learning. To address this challenge, we establish a mapping relationship from linguistic meta-actions to trajectories, and then leverage the trajectory feedback to optimize the reasoning of π_d through online reinforcement learning.

Online RL enables the model to continuously learn and optimize its policy through dynamic interaction with the environment, which is crucial for enhancing the model's understanding of causal relationships. We model our trajectory decision process tasks as a Markov Decision Process (MDP) [63, 66] for online reinforcement learning. The MDP is structured as a tuple $\langle S, A, P, R, \gamma \rangle$. The state $s_t \in S$ represents all necessary information for the agent to decide at step t. The model selects an action from the action space A according to the policy (π_d, π_g). Following an action, the system transitions to a new state s_{t+1} according to dynamics implicitly defined by closed-loop simulation environment. The reward $r_t \in R(s_t)$ is a scalar feedback signal that evaluates the quality of executing action in state s_t. Our objective is to learn a decision-making policy π_d that maximizes the expected cumulative discounted reward in the collected data τ, guided by the discount factor γ. This objective is formulated as:

$$J(\theta) = \mathbb{E}_\tau \sim \pi_d \left[R(\tau)\right] = \mathbb{E}_\tau \sim \pi_d \left[\sum_{t=0}^{T-1} \gamma^t r_t\right]. \tag{4}$$

3.2 Language-Action Mapping

To enhance the synergy between our decision-making π_d and trajectory generation policy π_g, we decouple a single LLM into two specialized experts with distinct LoRA parameters. One LLM serves as the Decision Expert, implementing the policy π_d, while the other acts as the Action Expert, responsible for the policy π_g. This architecture ensures they operate from a shared foundation of world knowledge while performing their distinct functions. We first leverage IL to create a mapping between the Decision Expert and the Action Expert, establishing a connection between language and action and enhancing exploration efficiency in the subsequent reinforcement learning process.

Inspired by [45, 58], we decouple the control into longitudinal and lateral control to improve planning flexibility and design the corresponding meta-action in planning QA. We generate the planning QA pairs using LLMs and refine them through manual filtering to ensure a one-to-one correspondence between language and actions (see the Appendix for details). The model is then trained on both the reasoning data and planning QA to learn the mapping from language to action. The loss function can be expressed as:

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\log P(y_{i,t}|x_i, y_{i,<t}). \tag{5}$$

where x denotes the input sequence, N denotes the batch size, and y denotes the target sequence, with $y_{i,<t}$ representing historical tokens preceding time step t. Then, we map the meta-actions to the temporal speed trajectory for longitudinal control and the geometric path trajectory for lateral control in Action Expert. Specifically, we leverage the autoregressive nature of the Action Expert to encode visual and language information into a hidden state h and introduce two special tokens, `<speed_waypoints>` and `<path_waypoints>`, to extract the logits from the Action Expert's output:

$$P_\theta(h) = \prod_{k=1}^{K} P_\theta(x_k \mid x_{<k}). \tag{6}$$

Finally, we utilize a Variational Autoencoder (VAE) [31] with a GRU-based decoder to align our language and action spaces, directly translating visual-language representations into the final action trajectory:

$$p(\boldsymbol{z}|h) = N(\mu_h, \sigma_h^2), \pi_g(a|V, L) = gru(\boldsymbol{z}), \tag{7}$$

where $\boldsymbol{z}$ is gaussian variables in the latent space.

We utilize the common detection loss $\mathcal{L}_{det}$ for auxiliary supervision [11]. The VAE is trained by a Kullback-Leibler divergence loss under the supervision of expert trajectories. We employ L1 loss as the Behavior Cloning (BC) loss for speed and path waypoint regression. The total loss is:

$$\mathcal{L}_{\pi}^{\text{il}}(\theta) = \mathcal{L}_{BC} + \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{vae}} + \mathcal{L}_{\text{det}}. \tag{8}$$

3.3 Online RL for Action-Reasoning

IL generates human-like trajectories but often suffers from causal confusion. To overcome this, we leverage online RL within the CARLA simulator [8]. As shown in Fig. 3, this online approach enables the agent to explore the environment through trial and error, learning from direct interaction and consequences, thereby bolstering the model's driving performance in complex scenarios. To leverage prior knowledge from IL, the value network shares the same weights as the LLM, with the only difference being that its final layer is replaced by a Multi-Layer Perceptron (MLP) to predict the state value.

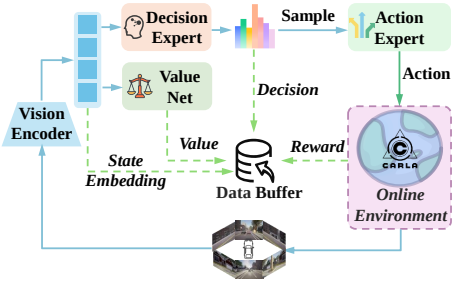

Fig. 3: Online reinforcement learning (RL) framework. The RL pipeline stores state embeddings, value estimates, decisions, and action rewards into the data buffer.

To achieve an efficient rollout process, we deployed N parallel CARLA collectors, focusing on routes across different scenarios that the model failed to complete after IL. At each step, we use a vision encoder to process the scene's visual information and produce state embeddings. We query the Decision Expert with question and sample from its output logits about meta-action tokens. The sampling meta-actions further map to precise trajectories in the action space by the Action Expert. Meanwhile, the value network estimates the value of the current state at each decision step.

Given that MindDrive has already acquired basic driving skills through IL pre-training, we employ a sparse reward function to guide the optimization of its high-level reasoning space. Specifically, a reward of $+1$ is assigned when the vehicle successfully reaches its destination, while a reward of -1 is incurred when a predefined penalty event is triggered. For all other normal driving scenarios, the reward is 0. The reward function r_t is defined as follows:

$$r_t = \begin{cases} +1 & \text{if the destination is reached} \\ -1 & \text{if a penalty event is triggered .} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

We use the official CARLA [8] leaderboard metrics as our penalty event. The penalty events include major infractions, such as collisions with other vehicles and running a red light (see more details in Exp. 4.3). The rollout process will be terminated once any penalty event is triggered.

After collecting a complete route, the δ value is calculated by the Temporal-Difference method:

$$\delta_t = r_t + \gamma \cdot V'(s_{t+1}) - V'(s_t), \tag{10}$$

where V' is the value network function, and γ is the discount factor to ensure future rewards are appropriately weighted in the decision-making process. Then we calculate GAE (Generalized Advantage Estimation):

$$\hat{G}_t = \delta_t + \gamma\lambda \cdot \hat{G}_{t+1}, \tag{11}$$

where λ is the trace-decay parameter that controls the bias-variance trade-off in advantage estimation. Consequently, the return $\hat{R}_t$ is computed as the sum of the advantage and the value estimate:

$$\hat{R}_t = \hat{G}_t + V'(s_t). \tag{12}$$

Instead of directly using multi-images for RL training, we use the state embedding extracted by the vision encoder to represent the current state. This approach can simultaneously integrate temporal and visual information and improve computational efficiency by avoiding repeated computation. We store the value, decision-action, and reward of each step into a data buffer.

Upon collecting all routes into the rollout buffer, we optimize the policy π_d using the Proximal Policy Optimization (PPO) algorithm [47]:

$$\mathcal{L}_{\text{ppo}} = -\mathbb{E}_t \left[\min \left(\frac{\pi_d}{\pi_{d_{\text{old}}}} \hat{G}_t, \text{clip} \left(\frac{\pi_d}{\pi_{d_{\text{old}}}}, 1-\epsilon, 1+\epsilon \right) \hat{G}_t \right) \right], \tag{13}$$

where ϵ is the clipping parameter in PPO. Benefiting from the proposed RL process, MindDrive is able to be trained in a large batch for stable optimization.

Meanwhile, to mitigate the problem of catastrophic forgetting during the RL fine-tuning phase, we introduce a Kullback-Leibler divergence loss as a regularization term. This loss function is designed to constrain the output distribution of the meta-actions of the Decision Expert. Its formula can be expressed as:

$$\mathcal{L}_{KL} = D_{KL}(P_{\text{ref}}(\cdot|s) \parallel P_\theta(\cdot|s)). \tag{14}$$

During training, only the parameters of the MLP head within the value net are updated. The optimization is performed by minimizing the Mean Squared Error (MSE) loss, formulated as:

$$\mathcal{L}_{V'}(\theta) = \frac{1}{N} \sum_{t=1}^{N} \left(V'_\theta(s_t) - \hat{R}_t \right)^2. \tag{15}$$

Finally, the online reinforcement policy learning loss is:

$$\mathcal{L}_\pi^{\text{rl}}(\theta) = \mathcal{L}_{ppo} + \mathcal{L}_{V'} + \beta\mathcal{L}_{KL}, \tag{16}$$

where β is a coefficient that controls the strength of the KL regularization.

4 Experiments

4.1 Experimental Settings

Dataset. We train and evaluate MindDrive on the Bench2Drive dataset [25], a comprehensive closed-loop benchmark for autonomous driving (AD) based on

the CARLA simulator [8]. During imitation learning (IL), we adopt the official base set of 1000 clips, with 950 used for training and 50 for open-loop validation. We perform closed-loop evaluation on the official Bench2Drive benchmark, including 220 short routes across 44 interactive scenarios. For the online reinforcement learning (RL) stage, we utilize a set of 44 rollout routes, where the model successfully reaches the destination during the action sampling phase (see the Appendix for details).

Metrics. For closed-loop evaluation, we adopt three key metrics: Driving Score (DS), Success Rate (SR), and Multi-Ability. Following the CARLA benchmark [8], the SR measures the proportion of routes completed within the specified time limit, while DS reflects route completion penalized by traffic infractions. Multi-Ability independently evaluates five advanced urban driving skills.

Model Setting. In imitation learning, we set the number of speed waypoints $\mathbf{w_s} \in \mathbb{R}^{N_s \times 2}$ to $N_s = 6$, representing a future 3-second trajectory sampled at 2Hz. For the path waypoints $\mathbf{w_p} \in \mathbb{R}^{N_p \times 2}$, $N_p = 20$ points are used to define a 20-meter path with a 1-meter interval between consecutive points. The decision space for the Decision VLM includes 7 speed meta-actions and 6 path meta-actions, aligning with the downstream planning trajectory. The detailed definitions of the meta-actions are provided in the Appendix.

Training Process. All experiments are conducted on 32 NVIDIA A800 GPUs with 80 GB of memory. We employ EVA-02-L [10] as the vision encoder. We use the lightweight Qwen2-0.5B [60] and Qwen2.5-3B [61] as our base LLM. We use Chat-B2D [11] as our reasoning data and jointly train with planning QA pairs. Both experts are fine-tuned with LoRA [15], the rank dimension and alpha set to 16. In the online RL stage, data is collected in parallel using 24 Carla simulations. The weights for all penalty components are fixed to unity, and no specialized fine-tuning procedure is adopted. We also propose detailed pseudocode for our online RL algorithm in the Appendix.

4.2 Main Results

We conduct a comprehensive comparison between MindDrive and representative methods from both the traditional end-to-end (E2E) and vision-language-action (VLA) paradigms on the Bench2Drive benchmark. Tab. 1 lists the detailed results, and we find that:

1) MindDrive delivers promising performance with a lightweight model. Compared with the traditional E2E methods, MindDrive surpasses the latest state-of-the-art (SOTA) IL model PGS [17] by 6.45% SR, and the online RL-based method Raw2Drive [63] by 6.68 DS and 4.85% SR. With a 3B parameter LLM, MindDrive-L outperforms the PGS [17] method by 2.51 in DS and delivers a 9.62% gain in SR. Within the VLA paradigm, MindDrive achieves comparable performance to the IL-based ORION [11] and outperforms DriveMoE [62] by 3.82 DS and 6.45% SR. In open-loop metrics, MindDrive exhibits a performance gap compared to DriveMoE. We attribute this to differing optimization objectives which DriveMoE employs a skill-specialized action MoE

Table 1: Closed-loop and Multi-Ability Results of E2E-AD Methods in Bench2Drive under **base** training set. C/L refers to camera/LiDAR. * and † denote expert feature distillation and imitation learning, respectively. "-L" represents larger model. DS: Driving Score, SR: Success Rate, M: Merging, O: Overtaking, EB: Emergency Brake, GW: Give Way, TS: Traffic Sign. Avg. L2 is averaged over the predictions in 2 seconds under 2Hz, similar to UniAD.

Method	LLM	Reference	Modality	Scheme	Closed-loop		Ability (%) ↑						Open-loop
					DS↑	SR(%)↑	M	O	EB	GW	TS	**Mean**	Avg. L2 ↓
Traditional End-to-End Paradigm													
TCP-traj*	-	NeurIPS 22	C	IL	59.90	30.00	8.89	24.29	51.67	40.00	46.28	34.22	1.70
ThinkTwice* [24]	-	CVPR 23	C	IL	62.44	31.23	27.38	18.42	35.82	50.00	54.23	37.17	0.95
DriveAdapter* [22]	-	ICCV 23	C&L	IL	64.22	33.08	28.82	26.38	48.76	50.00	56.43	42.08	1.01
UniAD-Base [16]	-	CVPR 23	C	IL	45.81	16.36	14.10	17.78	21.67	10.00	14.21	15.55	0.73
VAD [28]	-	ICCV 23	C	IL	42.35	15.00	8.11	24.44	18.64	20.00	19.15	18.07	0.91
GenAD [70]	-	ECCV 24	C	IL	44.81	15.90	-	-	-	-	-	-	-
MomAD [50]	-	CVPR 25	C	IL	44.54	16.71	-	-	-	-	-	-	0.87
WoTE [32]	-	ICCV 25	C&L	IL	61.71	31.36	-	-	-	-	-	-	-
DriveTransformer-L [26]	-	ICLR 25	C	IL	63.46	35.01	17.57	35.00	48.36	40.00	52.10	38.60	0.62
DiffAD [53]	-	arXiv 25	C	IL	67.92	38.64	30.00	35.55	46.66	40.00	46.32	38.79	1.55
PGS [17]	-	ICLR 26	C	IL	78.08	48.64	35.00	73.33	55.00	60.00	43.68	53.40	0.77
DriveDPO [48]	-	NeurIPS 25	C	Offline RL	62.02	30.62	-	-	-	-	-	-	-
Raw2Drive [63]	-	NeurIPS 25	C	Online RL	71.36	50.24	**43.35**	51.11	60.00	50.00	62.26	53.34	-
Vision-Language-Action Paradigm													
Driverπ0 [62]	Paligemma-3B	arXiv 25	C	IL	60.45	30.00	29.35	36.58	48.83	40.00	54.45	41.84	0.56
DriveMoE [62]	Paligemma-3B	arXiv 25	C	IL	74.22	48.64	34.67	40.00	65.45	40.00	59.44	47.91	0.38
ORION [11]	Vicuna-7B	ICCV 25	C	IL	77.74	54.62	25.00	71.11	**78.33**	30.00	69.15	54.72	0.68
ORION [11]	Qwen2-0.5B	ICCV 25	C	IL	72.89	45.83	26.32	62.22	55.55	50.00	63.30	51.37	0.67
UniDrive-WM [57]	Vicuna-7B	arXiv 26	C	IL	79.22	56.36	29.81	74.04	79.84	40.00	71.30	59.01	0.64
RecogDrive [33]	Qwen2.5-7B	ICLR 26	C	Offline RL	71.36	45.45	29.73	20.00	69.09	20.00	**71.34**	42.03	-
MindDriver [67]	Qwen2.5-3B	CVPR 26	C	Offline RL	65.48	39.55	-	-	-	-	-	-	-
AutoVLA [74]	Qwen2.5-3B	NeurIPS 25	C	Offline RL	78.84	57.73	-	-	-	-	-	-	-
MindDrive†	Qwen2-0.5B	-	C	IL	75.85	49.30	30.26	66.67	60.00	33.33	56.91	49.44	0.69
MindDrive-L†	Qwen2.5-3B	-	C	IL	75.78	50.71	33.33	57.78	68.33	40.00	61.05	52.10	0.66
MindDrive	Qwen2-0.5B	-	C	Online RL	78.04	55.09	32.89	**75.56**	68.33	50.00	57.89	56.94	0.69
MindDrive-L	Qwen2.5-3B	-	C	Online RL	**80.59**	**58.26**	42.31	66.67	76.67	**50.00**	68.42	**60.81**	0.66

that explicitly mitigates mode averaging during trajectory fitting to directly optimize open-loop performance. Conversely, MindDrive incorporates online RL rewards designed to prioritize closed-loop driving performance. Notably, using the lightweight Qwen2-0.5B, MindDrive achieves competitive performance in contrast to the substantially larger LLMs employed by ORION and DriveMoE (*i.e.*, Vicuna1.5-7B and Paligemma-3B), highlighting our method's superior efficiency.

2) Online RL enhances MindDrive's capabilities for complex, dynamic interactions. As shown in Tab. 1, MindDrive demonstrates a clear advantage over other methods. It surpasses the offline RL-based RecogDrive [33] by 6.68 DS and 9.64% SR, and MindDrive-L outperforms the SOTA RL-based method AutoVLA [74] by 1.75 DS and 0.53% SR. Furthermore, MindDrive outperforms the IL-based MindDrive 2.19 DS and 5.79% SR, while MindDrive-L surpasses the IL-based MindDrive-L 4.81 DS and 7.55% SR, strongly validating the superiority of the proposed online RL paradigm.

The Multi Ability metric further supports this finding. Specifically, MindDrive achieves a 14.91% improvement in mean ability over RecogDrive [33] and 5.57% over the IL method ORION [11] with the same lightweight LLM. Notably, it achieves substantial gains in capabilities closely tied to meta-actions selection, with improvements of 55.56% in Overtaking and 30% in Give Way compared to the offline RL method RecogDrive. Although MindDrive performs lower than

Table 2: Ablation on different penalty events. C: Collision, TL: Traffic Light, RD: Route Deviation, S: Stop. M: Merging, O: Overtaking, EB: Emergency Brake, GW: Give Way, TS: Traffic Sign, DS/SR: Driving Score/Success Rate.

ID	Penalty Events				Closed-loop ↑		**Ability** (%) ↑					
	C	TL	RD	S	DS	SR(%)	M	O	EB	GW	TS	**Mean**
1					75.85	49.30	30.26	66.67	60.00	33.33	56.91	49.44
2	✓				75.41	50.70	28.00	71.11	60.00	50.00	56.91	53.20
3	✓	✓			76.02	51.47	31.42	57.77	68.97	50.00	58.43	53.32
4	✓	✓	✓		76.95	51.85	27.63	73.33	65.00	50.00	57.36	54.67
5	✓	✓	✓	✓	**78.04**	**55.09**	32.89	75.56	68.33	50.00	57.89	**56.94**

the SOTA VLA methods in the Emergency Brake and Traffic Sign ability, it still exhibits a noticeable advantage over our IL-based version, with improvements of 8.33% and 0.98%, respectively. Furthermore, MindDrive-L achieves new SOTA performance on the DS, SR, and mean driving ability metrics. These results confirm that online RL significantly strengthens the model's causal reasoning and decision-making robustness in complex interactive environments.

4.3 Ablation Studies

In ablation studies, we use 0.5B parameter LLM model. For each route, two online reinforcement learning epochs are performed unless otherwise specified.

Ablation on Penalty Events. We introduce four penalty events during the online RL stage: collisions with pedestrians or vehicles, running a red light, driving off-road, or deviating over 30 meters from the route, and failing to obey a stop sign (*i.e.*, Collision, Traffic Light, Route Deviation, and Stop, respectively). We assign a sparse reward of -1 to penalize our model for triggering these events. As shown in Tab. 2, the progressive incorporation of these penalties leads to consistent improvements in both the success rate and the mean driving ability score relative to the IL baseline (ID-1).

Specifically, introducing the collision penalty (ID-2) yields a 1.4% SR and 3.76% mean ability improvement over the IL baseline (ID-1), while maintaining a comparable DS. In overtaking scenarios, MindDrive demonstrates exceptional performance, outperforming the baseline (ID-1) by a significant margin of 4.44%. We attribute this improvement to the model's learned ability to adopt a more proactive strategy for collision avoidance in the continuous and interactive traffic flow. However, this strategic shift consequently impairs its merging performance.

Incorporating the traffic-light penalty (ID-3) brings further improvements over the IL baseline (ID-1), elevating Traffic Sign by 1.52% and Emergency Braking by 8.97%. However, conflicting reward signals within this penalty lead to a noticeable drop in overtaking performance. Introducing a route-deviation penalty (ID-4) helps achieve a better trade-off between decisiveness and caution, although the stricter constraints on exploration limit further performance gains.

Table 3: Ablation studies on policy regularization and control methods. DS and SR denote Driving Score and Success Rate, respectively.

(a) Ablation on policy regularization

Policy Regularization	DS ↑	SR(%) ↑
PPO-Vanilla	74.73	46.73
PPO-Entropy	75.71	49.24
PPO-KL	**78.04**	**55.09**

(b) Ablation on control methods

Control Method	DS ↑	SR(%) ↑
Navigation Command (IL)	68.11	41.59
Meta Action (IL)	75.85	49.30
Meta Action (RL)	**78.04**	**55.09**

Notably, adding a stop sign penalty significantly boosts overall performance, as it strongly correlates with the stop meta-action, enabling more effective policy learning. This is especially beneficial in merging scenarios that feature stop signs, yielding improvements of 5.26% in Merging ability and 3.24% in the SR compared to ID-4.

Without complex reward engineering, MindDrive can discover effective driving policies through online trial-and-error, autonomously learning from failures to progressively determine optimal actions. We argue that by building upon the foundational driving skills acquired via IL, MindDrive operates effectively in a binary outcome setting where an episode either succeeds or fails. In this setup, the sparse reward from the final outcome is directly propagated back to each decision step, providing sufficient supervision to distinguish between optimal and suboptimal actions. Furthermore, our replay buffer retains both successful and failed episodes, and the large batch size amplifies the contrast between trajectories of different outcomes and lengths, providing sufficient gradient signals for policy improvement.

Ablation on Policy Regularization. We evaluate different policy regularization methods within PPO, as reported in Tab. 3 (a). Our method outperforms vanilla PPO by 3.31 DS and 8.36% SR, demonstrating that KL divergence loss effectively stabilizes policy updates during RL training and mitigates catastrophic forgetting. Compared to entropy-based regularization (PPO-Entropy), our approach improves by 2.33 DS and 5.85% SR, indicating that while entropy regularization promotes exploration, excessive policy stochasticity is suboptimal for goal-directed driving tasks. Overall, our KL regularization method enables more efficient learning, with a faster policy optimization process and higher sample efficiency than the baselines.

Ablation on Control Method. We further investigate the effects of different control methods by comparing two high-level instruction approaches: navigation commands and LLM-generated meta-actions. As detailed in Tab. 3 (b), the VLM-guided meta-actions IL model achieves 7.74 DS and 7.71% SR improvements over the navigation-command baseline, indicating that VLM-derived meta-actions enable more effective reasoning in complex traffic scenarios. Meanwhile, employing online RL further improves meta-action selection, yielding additional gains of 2.19 DS and 5.79% SR, which demonstrates that real-time interactive reinforcement learning can dynamically optimize the reasoning logic of

vision-language meta-actions and better adapt to the constantly changing traffic environments.

Ablation on the Number of Training Epochs. We finally analyze MindDrive's sensitivity to the number of training epochs during online RL. As shown in Fig. 4, when only one epoch is trained, the value network's inaccurate estimation leads to wrong advantage estimates for each action, resulting in a performance drop compared to the baseline. After two training epochs, our model significantly outperforms the baseline, improving DS by 2.19 and SR by 5.79%. However, further increasing the epochs substantially degrades performance, with DS dropping from 78.04 to 73.69 and SR declining from 55.09% to 45.12%. We attribute this degradation to catastrophic forgetting, where excessive training causes the policy to overfit recent experiences and forget previously learned scene understanding (See the Appendix for full analysis). MindDrive further enhances the model's driving capabilities via online RL, achieving optimal performance in just two online training epochs, thereby demonstrating the high efficiency of our online RL framework. Therefore, we set the default number of training epochs to 2, balancing sample efficiency and training stability.

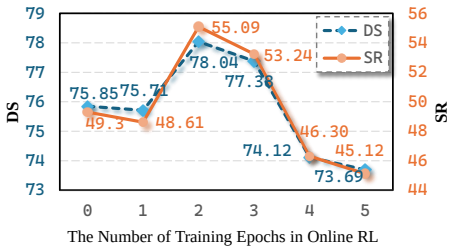

Fig. 4: Ablation on the Training Epochs for Online RL. DS and SR denote Driving Score and Success Rate, respectively.

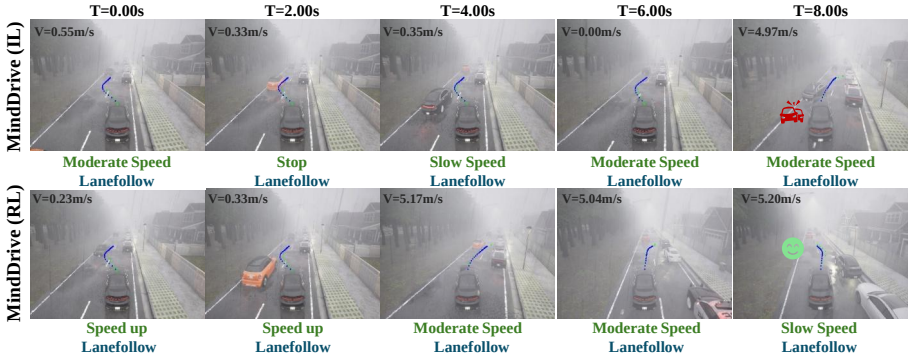

Fig. 5: Qualitative results of MindDrive after IL and RL training on the closed-loop evaluation of the Bench2Drive lanefollow scenario. The green and blue refer to the prediction speed and path meta-actions, respectively. Red denote a collision has occurred.

4.4 Qualitative Results

We present a qualitative comparison between the IL and online RL versions of MindDrive in the Bench2Drive lane-follow scenario, as shown in Fig. 5. The IL paradigm exhibits strong task-specific competence, such as issuing timely stop commands for early braking. However, IL-only MindDrive struggles in dynamic and interactive scenarios, especially in those requiring complex decision-making (*e.g.*, determining the optimal timing for lane changes). After Online RL training, MindDrive selects more robust meta-actions in challenging scenarios, resulting in safer and more decisive lane-change behaviors.

These qualitative results demonstrate that the online RL stage substantially enhances the VLM's high-level reasoning and decision-making capabilities, enabling it to better handle complex and uncertain traffic environments.

5 Conclusion

In this paper, we introduce **MindDrive**, a novel autonomous driving framework that leverages language as an interface for online reinforcement learning (RL). MindDrive maps language instructions to actions, transforming the exploration space into a discrete language space to reduce the cost of online RL. It further enables the Large Language Model to refine its reasoning through action feedback in a closed-loop simulator. Extensive experiments within our proposed online RL training framework demonstrate that MindDrive achieves state-of-the-art performance with a lightweight model. To the best of our knowledge, this is the first work to successfully train a Vision-Language-Action model for autonomous driving in an interactive simulator. We expect this work will provide valuable insights for the autonomous driving community.

Limitations. Due to the absence of real-world interactive simulators for autonomous driving, our evaluation is currently limited to the CARLA simulator [8]. Additionally, the challenge of synchronizing multiple CARLA simulators prevents us from evaluating multiple candidate actions from an identical initial state across parallel environments, restricing the application of the GRPO [49] algorithm in our framework. Future work will explore real-world interactive simulation infrastructures or alternative training strategies to better support parallel policy evaluation and more efficient online reinforcement learning.

Acknowledgment. This work is supported by the NSFC (62225603, 623B2038).

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: A family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 **1** (2023)

3. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164 (2024)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020)
5. Cheng, J., Chen, Y., Mei, X., Yang, B., Li, B., Liu, M.: Rethinking imitation-based planners for autonomous driving. In: Proc. of the IEEE Int. Conf. on Robotics and Automation. pp. 14123–14130 (2024)
6. Dauner, D., Hallgarten, M., Li, T., Weng, X., Huang, Z., Yang, Z., Li, H., Gilitschenski, I., Ivanovic, B., Pavone, M., et al.: Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In: Proc. of Advances in Neural Information Processing Systems. vol. 37, pp. 28706–28719 (2024)
7. De Haan, P., Jayaraman, D., Levine, S.: Causal confusion in imitation learning. Proc. of Advances in Neural Information Processing Systems **32** (2019)
8. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Proc. of IEEE Intl. Conf. on Robot Learning. pp. 1–16 (2017)
9. Fang, H., Li, S., Wang, S., Xi, X., Liang, D., Bai, X.: Towards generalizable robotic manipulation in dynamic environments. arXiv preprint arXiv:2603.15620 (2026)
10. Fang, Y., Sun, Q., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva-02: A visual representation for neon genesis. Image and Vision Computing **149**, 105171 (2024)
11. Fu, H., Zhang, D., Zhao, Z., Cui, J., Liang, D., Zhang, C., Zhang, D., Xie, H., Wang, B., Bai, X.: Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In: Porc. of IEEE Intl. Conf. on Computer Vision (2025)
12. Gao, H., Chen, S., Jiang, B., Liao, B., Shi, Y., Guo, X., Pu, Y., Yin, H., Li, X., Zhang, X., et al.: Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. In: Proc. of Advances in Neural Information Processing Systems (2025)
13. Guan, Y., Yin, L., Liang, D., Ju, J., Luo, Z., Luan, J., Liu, Y., Bai, X.: Video streaming thinking: Videollms can watch and think simultaneously. arXiv preprint arXiv:2603.12262 (2026)
14. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
15. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: Proc. of Intl. Conf. on Learning Representations (2021)
16. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 17853–17862 (2023)
17. Huang, Y., Qu, Z., Jiang, L., Liu, B., Zhang, H.: Prioritizing perception-guided self-supervision: A new paradigm for causal modeling in end-to-end autonomous driving. arXiv preprint arXiv:2511.08214 (2025)
18. Huang, Z., Feng, C., Yan, F., Xiao, B., Jie, Z., Zhong, Y., Liang, X., Ma, L.: Robotron-drive: All-in-one large multimodal model for autonomous driving. In: Porc. of IEEE Intl. Conf. on Computer Vision. pp. 8011–8021 (2025)

19. Huang, Z., Sheng, Z., Qu, Y., You, J., Chen, S.: Vlm-rl: A unified vision language models and reinforcement learning framework for safe autonomous driving. Transportation Research Part C: Emerging Technologies **180**, 105321 (2025)
20. Hwang, J.J., Xu, R., Lin, H., Hung, W.C., Ji, J., Choi, K., Huang, D., He, T., Covington, P., Sapp, B., et al.: Emma: End-to-end multimodal model for autonomous driving. arXiv preprint arXiv:2410.23262 (2024)
21. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: $\pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
22. Jia, X., Gao, Y., Chen, L., Yan, J., Liu, P.L., Li, H.: Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In: Porc. of IEEE Intl. Conf. on Computer Vision (2023)
23. Jia, X., Shi, S., Chen, Z., Jiang, L., Liao, W., He, T., Yan, J.: Amp: Autoregressive motion prediction revisited with next token prediction for autonomous driving. arXiv preprint arXiv:2403.13331 (2024)
24. Jia, X., Wu, P., Chen, L., Xie, J., He, C., Yan, J., Li, H.: Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (2023)
25. Jia, X., Yang, Z., Li, Q., Zhang, Z., Yan, J.: Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In: Proc. of Advances in Neural Information Processing Systems (2024)
26. Jia, X., You, J., Zhang, Z., Yan, J.: Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In: Proc. of Intl. Conf. on Learning Representations (2025)
27. Jiang, A., Gao, Y., Wang, Y., Sun, Z., Wang, S., Heng, Y., Sun, H., Tang, S., Zhu, L., Chai, J., et al.: Irl-vla: Training an vision-language-action policy via reward world model. arXiv preprint arXiv:2508.06571 (2025)
28. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: Porc. of IEEE Intl. Conf. on Computer Vision. pp. 8340–8350 (2023)
29. Jiang, B., Chen, S., Zhang, Q., Liu, W., Wang, X.: Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. arXiv preprint arXiv:2503.07608 (2025)
30. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
31. Kingma, D.P.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
32. Li, Y., Wang, Y., Liu, Y., He, J., Fan, L., Zhang, Z.: End-to-end driving with online trajectory evaluation via bev world model. In: Porc. of IEEE Intl. Conf. on Computer Vision (2025)
33. Li, Y., Xiong, K., Guo, X., Li, F., Yan, S., Xu, G., Zhou, L., Chen, L., Sun, H., Wang, B., et al.: Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In: Proc. of Intl. Conf. on Learning Representations (2026)
34. Li, Z., Ji, Q., Ling, X., Liu, Q.: A comprehensive review of multi-agent reinforcement learning in video games. IEEE Transactions on Games (2025)
35. Liao, B., Chen, S., Yin, H., Jiang, B., Wang, C., Yan, S., Zhang, X., Li, X., Zhang, Y., Zhang, Q., et al.: Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 12037–12047 (2025)

36. Liu, H., Li, T., Yang, H., Chen, L., Wang, C., Guo, K., Tian, H., Li, H., Li, H., Lv, C.: Reinforced refinement with self-aware expansion for end-to-end autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence (2026)
37. Liu, M., Cheng, H., Chen, L., Broszio, H., Li, J., Zhao, R., Sester, M., Yang, M.Y.: Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 2039–2049 (2024)
38. Liu, Z., Huang, R., Yang, R., Yan, S., Wang, Z., Hou, L., Lin, D., Bai, X., Zhao, H.: Drivepi: Spatial-aware 4d mllm for unified autonomous driving understanding, perception, prediction and planning. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (2026)
39. Luo, Y., Li, F., Xu, S., Lai, Z., Yang, L., Chen, Q., Luo, Z., Xie, Z., Jiang, S., Liu, J., et al.: Adathinkdrive: Adaptive thinking via reinforcement learning for autonomous driving. arXiv preprint arXiv:2509.13769 (2025)
40. Mao, W., Wang, T., Zhang, D., Yan, J., Yoshie, O.: Pillarnest: Embracing backbone scaling and pretraining for pillar-based 3d object detection. IEEE Transactions on Intelligent Vehicles (2024)
41. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Proc. of Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)
42. Pan, Y., Cheng, C.A., Saigol, K., Lee, K., Yan, X., Theodorou, E., Boots, B.: Agile autonomous driving using end-to-end deep imitation learning. Robotics: Science and Systems (2018)
43. Puterman, M.L.: Markov decision processes. Handbooks in operations research and management science **2**, 331–434 (1990)
44. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Proc. of Advances in Neural Information Processing Systems **36**, 53728–53741 (2023)
45. Renz, K., Chen, L., Arani, E., Sinavski, O.: Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 11993–12003 (2025)
46. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proc. of Intl. Conf. artificial intelligence and statistics. pp. 627–635 (2011)
47. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
48. Shang, S., Chen, Y., Wang, Y., Li, Y., Zhang, Z.: Drivedpo: Policy learning via safety dpo for end-to-end autonomous driving. In: Proc. of Advances in Neural Information Processing Systems (2025)
49. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
50. Song, Z., Jia, C., Liu, L., Pan, H., Zhang, Y., Wang, J., Zhang, X., Xu, S., Yang, L., Luo, Y.: Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (2025)
51. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)

52. Wang, S., Yu, Z., Jiang, X., Lan, S., Shi, M., Chang, N., Kautz, J., Li, Y., Alvarez, J.M.: Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 22442–22452 (2025)
53. Wang, T., Zhang, C., Qu, X., Li, K., Liu, W., Huang, C.: Diffad: A unified diffusion modeling approach for autonomous driving. arXiv preprint arXiv:2503.12170 (2025)
54. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Proc. of Advances in Neural Information Processing Systems **35**, 24824–24837 (2022)
55. Wu, X., Liang, D., Feng, T., Xia, K., Zhang, Y., Li, X., Tan, X., Bai, X.: Generation models know space: Unleashing implicit 3d priors for scene understanding. arXiv preprint arXiv:2603.19235 (2026)
56. Xing, S., Qian, C., Wang, Y., Hua, H., Tian, K., Zhou, Y., Tu, Z.: Openemma: Open-source multimodal model for end-to-end autonomous driving. In: Proc. of IEEE Winter Conf. on Applications of Computer Vision. pp. 1001–1009 (2025)
57. Xiong, Z., Ye, X., Yaman, B., Cheng, S., Lu, Y., Luo, J., Jacobs, N., Ren, L.: Unidrive-wm: Unified understanding, planning and generation world model for autonomous driving. arXiv preprint arXiv:2601.04453 (2026)
58. Xu, Z., Bai, Y., Zhang, Y., Li, Z., Xia, F., Wong, K.Y.K., Wang, J., Zhao, H.: Drivegpt4-v2: Harnessing large language model capabilities for enhanced closed-loop autonomous driving. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 17261–17270 (2025)
59. Xu, Z., Zhang, Y., Xie, E., Zhao, Z., Guo, Y., Wong, K.Y.K., Li, Z., Zhao, H.: Drivegpt4: Interpretable end-to-end autonomous driving via large language model. IEEE Robotics and Automation Letters (2024)
60. Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., et al.: Qwen2 technical report. arXiv preprint arXiv:2407.10671 (2024)
61. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., et al.: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)
62. Yang, Z., Chai, Y., Jia, X., Li, Q., Shao, Y., Zhu, X., Su, H., Yan, J.: Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. arXiv preprint arXiv:2505.16278 (2025)
63. Yang, Z., Jia, X., Li, Q., Yang, X., Yao, M., Yan, J.: Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). In: Proc. of Advances in Neural Information Processing Systems (2025)
64. Yuan, Z., Tang, J., Luo, J., Chen, R., Qian, C., Sun, L., Chu, X., Cai, Y., Zhang, D., Li, S.: Autodrive-r^2: Incentivizing reasoning and self-reflection capacity for vla model in autonomous driving. arXiv preprint arXiv:2509.01944 (2025)
65. Zhang, D., Zheng, Z., Niu, H., Wang, X., Liu, X.: Fully sparse transformer 3-d detector for lidar point cloud. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–12 (2023)
66. Zhang, D., Liang, J., Guo, K., Lu, S., Wang, Q., Xiong, R., Miao, Z., Wang, Y.: Carplanner: Consistent auto-regressive trajectory planning for large-scale reinforcement learning in autonomous driving. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 17239–17248 (2025)
67. Zhang, L., Yuan, Y., Wu, C., Chang, X., Cai, X., Zeng, S., Shi, L., Wang, S., Zhang, H., Xu, M.: Minddriver: Introducing progressive multimodal reasoning for autonomous driving. arXiv preprint arXiv:2602.21952 (2026)

68. Zhang, Z., Zheng, H., Wang, Y., Xu, L., Deng, T., Chen, X., Chen, Q., Zhang, B., Huang, W.: Omnidrive-r1: Reinforcement-driven interleaved multi-modal chain-of-thought for trustworthy vision-language autonomous driving. In: Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition. pp. 1106–1116 (June 2026)
69. Zhao, Z., Fu, H., Liang, D., Zhou, X., Zhang, D., Xie, H., Wang, B., Bai, X.: Extending large vision-language model for diverse interactive tasks in autonomous driving. arXiv preprint arXiv:2505.08725 (2025)
70. Zheng, W., Song, R., Guo, X., Zhang, C., Chen, L.: Genad: Generative end-to-end autonomous driving. In: Proc. of European Conference on Computer Vision. pp. 87–104 (2024)
71. Zhou, X., Liang, D., Chen, X., Tan, F., Zhang, D., Zhao, H., Bai, X.: Hermes++: Toward a unified driving world model for 3d scene understanding and generation. arXiv preprint arXiv:2604.28196 (2026)
72. Zhou, X., Liang, D., Tu, S., Chen, X., Ding, Y., Zhang, D., Tan, F., Zhao, H., Bai, X.: Hermes: A unified self-driving world model for simultaneous 3d scene understanding and generation. In: Porc. of IEEE Intl. Conf. on Computer Vision (2025)
73. Zhou, Y., Huang, L., Bu, Q., Zeng, J., Li, T., Qiu, H., Zhu, H., Guo, M., Qiao, Y., Li, H.: Embodied understanding of driving scenarios. In: Proc. of European Conference on Computer Vision. pp. 129–148 (2024)
74. Zhou, Z., Cai, T., Zhao, S.Z., Zhang, Y., Huang, Z., Zhou, B., Ma, J.: Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. In: Proc. of Advances in Neural Information Processing Systems (2025)
75. Zhu, R., Zhao, J., Zhang, D., Wang, G., Chen, X., Zhang, S., Gong, J., Zhou, Q., Zhang, W., Wang, N., et al.: Sparsead: Sparse query-centric paradigm for efficient end-to-end autonomous driving. IEEE Transactions on Artificial Intelligence (2025)