DriveFine: Refining-Augmented Masked Diffusion VLA for Efficient and Robust Driving

Chenxu Dang[1,2,3⋆], Sining Ang[3], Yongkang Li[1], Haochen Tian[2], Jie Wang[2], Guang Li[2], Hangjun Ye[2], Jie Ma[1†], Long Chen[2†], and Yan Wang[3†]

[1] Huazhong University of Science and Technology
[2] Xiaomi EV
[3] Institute for AI Industry Research (AIR), Tsinghua University

Abstract. End-to-end autonomous driving increasingly relies on generative planners, broadly classified into continuous diffusion policies and discrete token-based vision-language-action models (VLAs). While diffusion policies exploit parallel iterative noise prediction to efficiently generate accurate trajectories, they suffer from cross-modal misalignment, limited training efficiency, and poor generalization. Conversely, token-based VLAs generate discrete trajectory tokens autoregressively, offering better generalization but being constrained by inefficient causal reasoning, cumulative error, and irreversible decoding. To overcome these limitations, we propose **DriveFine**, a vision-language-action model that explores masked diffusion LLMs for trajectory planning, which significantly improves the efficiency, flexibility, and adaptability of trajectory generation. To address the inherent irreversibility of token decoding and improve the trajectory quality, we introduce a plug-and-play block Mixture-of-Expert (**block MoE**) module, which seamlessly injects the refinement capability into the dLLM at minimal cost. Through explicit expert routing and gradient isolation, DriveFine decouples generation and refinement, preventing cross-task interference. We further devise a tailored hybrid reinforcement learning strategy to facilitate the effective exploration of the refinement expert and further explore the performance ceiling. Extensive experiments on the NAVSIM v1, v2, and Navhard benchmarks demonstrate that DriveFine exhibits strong performance, robustness and efficiency. The code will be released.

Keywords: Masked Diffusion LLM · Reinforcement Learning · Block MoE · Autonomous Driving

1 Introduction

End-to-end autonomous driving systems (E2E-AD) are required to integrate sensor observations and textual instructions while generating driving actions or trajectories with a planner. Early deterministic planners (e.g., MLP-based [13,16, 55] or anchor-based classifiers [4,26]) tended to imitate a single expert trajectory,

⋆ Completed during the internship at Xiaomi EV and AIR. † Corresponding authors.

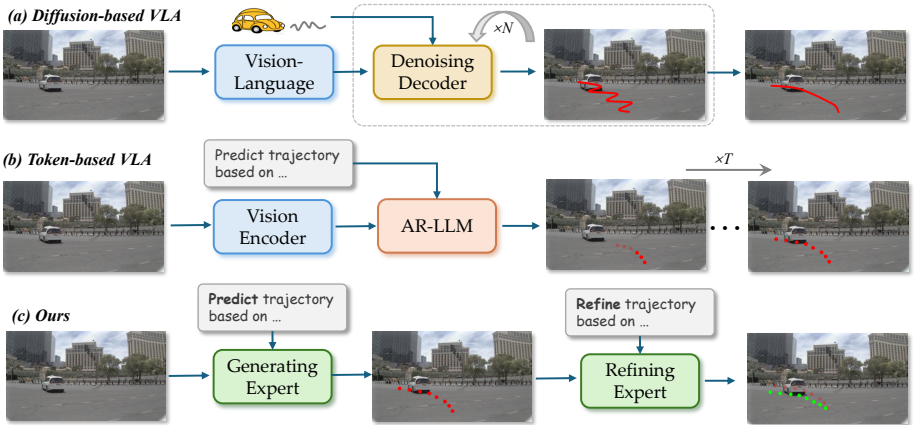

Fig. 1: Comparison of generative planners. (a) Parallel refinement with multi-step denoising. (b) Token-by-token decoding. (c) Generation then refinement.

which struggle with distribution shift and cannot capture the inherently multi-modal nature of driving.

Recently, non-deterministic generative planners [25, 30, 31, 54] have emerged as the dominant paradigms in E2E-AD. They predict driving trajectories as probability distributions, which effectively captures the multimodal nature of driving behavior. Moreover, their inherent sampling capability facilitates active exploration and allows the integration with rule-driven reinforcement learning strategies, such as GRPO [35], to guide the driving policy toward human-preferred behaviors. Current state-of-the-art generative planners can be categorized into diffusion-based policies [19, 25, 30, 56] with continuous action denoising and token-based vision-language-action models (VLAs) [31, 53, 54] with discrete action decoding.

(1) **Diffusion Policies**, as shown in Fig. 1(a), iteratively refine trajectories by constructing a Markov chain and predicting the continuous noise distributions in parallel, which typically facilitates the efficient and accurate trajectory generation.

However, the standalone diffusion planner head hinders the cross-modal alignment, leading to inefficient training that typically requires hundreds of epochs [25, 30]. Even more concerning, diffusion policies typically exhibit limited generalization and robustness. As evidenced empirically in Fig. 2: upon undergoing reinforcement fine-tuning utilizing PDMS as the reward signal, two representative diffusion policies, ReCogDrive [25] and DiffusionDrive [30], both exhibit a sig-

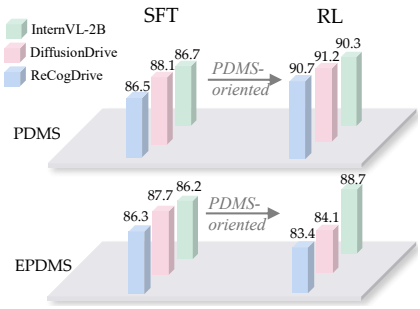

Fig. 2: PDMS-oriented RFT.

nificant decline in EPDMS. Here, PDMS and EPDMS are metrics provided by NAVSIM [10], see 4.1 for details. We attribute this degradation to the weak coupling between diffusion heads and the upstream feature encoders (VLMs or ViTs), which induces reward hacking and the loss of pretrained knowledge and substantially limits their practical applicability.

(2) **Token-based VLAs**, as illustrated in Fig. 1(b), decode actions into tokens autoregressively within a predefined vocabulary, thus achieving a unified representation across vision, language, and action. As shown in Fig. 2, the PDMS-oriented RFT for InternVL [6] leads to simultaneous improvements in both PDMS and EPDMS, exhibiting stronger generalization and robustness.

Despite favorable properties, token-based VLAs [31, 54] generally lag behind methods with diffusion policies [25, 56] in both performance and efficiency across standard benchmarks [10]. This limitation mainly stems from their causal attention and autoregressive decoding mechanisms, which introduce additional inference latency and inherently suffer from limited context modeling and cumulative errors. Moreover, token-based VLAs suffer from irreversible decoding: once trajectory tokens are generated, they cannot be modified. As a result, they inherently lack explicit self-refinement mechanism, leading to insufficient reasoning and suboptimal trajectory quality.

Obviously, both dominant planning paradigms exhibit complementary strengths and weaknesses, which motivates the exploration of integrating the their advantages. Recently, masked diffusion LLMs (dLLMs) have emerged as a highly promising generative paradigm. By parallel unmasking of discrete masked tokens and selectively remasking, they generate content efficiently and have found broad applications in text [34, 48] and multimodal domains [21, 49].

To harness these properties for trajectory planning, we propose **DriveFine**, a vision-language-action (VLA) model built upon dLLMs, motivated by their distinctive advantages over autoregressive approaches:

- Parallel decoding to improve inference efficiency.
- Bidirectional attention facilitates comprehensive context extraction.
- Flexible, adaptive decoding mechanism to prevent error accumulation.

In particular, we employ a pretrained dLLM (LLaDA) as the base trajectory generator. Continuous visual features are extracted via a vision tower [40], and textual instructions are employed to guide the iterative decoding of trajectory tokens from fully masked token sequences.

However, decoding in dLLMs is also inherently irreversible. As illustrated in Fig. 3, the lack of self-refinement prevents already decoded low-quality trajectory points (e.g., collisions or outliers) from being corrected, which may result in overall trajectory failure. In contrast, diffusion policies such as ReCogDrive [25] naturally support iterative refinement, repeatedly revising and refining trajectories to ensure high-quality outputs. This raises the challenge of endowing dLLMs with analogous self-refinement capabilities to enable iterative trajectory refinement and improvement.

Inspired by the success of Mixture-of-Transformers [20, 29], we design a block-wise Mixture-of-Experts (MoE) architecture. Specifically, given a pretrained

dLLM composed of stacked Transformer blocks, we duplicate a small number of the final blocks as expert blocks, thereby partitioning the model into two independent experts responsible for trajectory generation and refinement, respectively. The remaining blocks are shared across experts. The generation expert preserves the behavior and knowledge of the pre-trained dLLM, while the refinement expert is plug-and-play, greatly enhancing its practicality. As shown in Fig. 1(c), task-specific experts are proactively routed during inference. During training, the gradients of the refinement expert are isolated from those of the generation expert, enabling the cross-task decoupling while avoiding cross-expert interference.

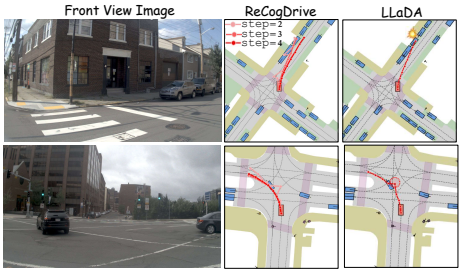

Fig. 3: Failure cases caused by irreversible decoding.

To further unlock the performance potential of DriveFine, we design a tailored hybrid reinforcement learning strategy. Specifically, we employ GRPO to reinforce the generation expert, which generates a group of trajectories and computes group-wise relative advantages. The generated trajectories are pairwise combined to construct online anchor–target trajectory pairs. In parallel, the refinement expert actively refines the generated trajectories, yielding online trajectory pairs, which, together with offline pairs, jointly supervise the refinement expert. Experimental results demonstrate that the block-wise MOE significantly enhances trajectory quality with limited parameter increase and slight inference overhead, raising the performance ceiling of DriveFine. We summarize our core contributions as follows:

- We analyze the limitations of the two dominant generative planners and propose DriveFine, which leverages masked diffusion LLMs (dLLMs) as a unified solution for trajectory planning
- We propose a plug-and-play block-MoE module that injects the self-refinement capability into dLLMs at minimal cost.
- We design a tailored hybrid reinforcement learning strategy to further elevate DriveFine's performance ceiling.
- Extensive experiments demonstrate that DriveFine consistently achieves state-of-the-art performance on NavSim v1, v2, and Navhard benchmarks.

2 Related Work

2.1 End to end Autonomous Driving

Early end-to-end planners predominantly relied on deterministic modeling, such as MLP-based regression [7, 13, 16, 36] and anchor-based classification [4], imitating a single expert trajectory. GenAD [51] imposes a GRU-based generator.

To align the trajectory diversity, diffusion policies were introduced to recover trajectories from random noise, thereby better aligning with the inherent uncertainty of autonomous driving. Diffusion planners [37, 52] leverage diffusion policies to jointly perform trajectory prediction and planning. GoalFlow [45] adopts goal-conditioned flow matching to generate trajectories, while DiffusionDrive [30] applies truncated diffusion over multiple anchor modes.

2.2 VLAs for Autonomous Driving

Conventional end-to-end driving models are often regarded as black boxes. To enhance interpretability, understanding, and reasoning, vision–language models (VLMs) [1, 6] have been increasingly incorporated into autonomous driving systems in recent years. Early explorations primarily focused on high-level scene understanding [42] and reasoning [14, 33, 43], while the demand for action generation gave rise to a large body of Vision–Language–Action (VLA) models. Early two-stage VLAs [5, 15, 32, 39] generate meta-actions or low-frequency actions from a VLM, followed by an e2e planner for refinement. However, gradient isolation in such pipelines contradicts the principle of end-to-end learning. In contrast, recent one-stage VLAs directly output the final trajectory. For example, OpenDriveVLA [53] autoregressively decodes trajectories, while AutoVLA [54] augments the action vocabulary with clustered anchors. These approaches feature token-based decoding and are thus referred to as token-based VLAs. Other approaches incorporate a planner on top of the VLM to facilitate cross-modal alignment. Orion [12] employs a GRU-based decoder, while recent works [9,19,25] increasingly adopt diffusion policies as planners, achieving stronger performance. Concurrent work [46] explore masked diffusion LLMs but overlooks the issue of irreversible decoding.

2.3 Reinforcement Learning for Autonomous Driving

Imitation learning lacks negative supervision from counterexamples, which leads to poor performance in out-of-distribution (OOD) scenarios. To address this limitation, many studies introduce reinforcement learning to improve the generalization of planners. AlphaDrive [17] is the pioneer that introduces GRPO [35] to promote the driving policy. AdaThinkDrive [31] and AutoVLA [54] directly apply GRPO to token-based VLAs, whereas ReCogDrive [25] designs a diff-GRPO. DiffusionDrivev2 [56] further proposes intra- and inter-anchor truncated GRPO strategies to enhance the multi-modal diffusion policy [30]. More recently, a series of scoring-based reinforcement learning models [18,26,27] perform stage-wise reasoning under direct reward guidance, leading to stronger performance.

3 Method

In this section, we present a detailed introduction of DriveFine with the overview in Fig. 4. Sec. 3.1 describes the formulation of trajectory planning with the

pretrained dLLM. Sec 3.2 details the proposed Block-MoE for refinement. Sec 3.3 illustrates the hybrid reinforcement learning strategy.

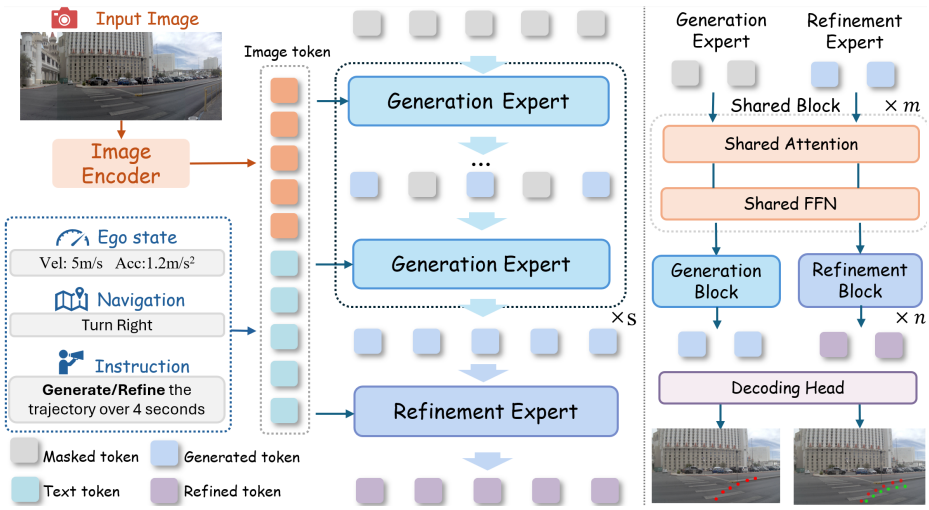

Fig. 4: Architecture of DriveFine. Visual and textual inputs are jointly aligned into a unified language space. A set of masked tokens undergoes s steps of parallel denoising followed by a single refinement step. The main difference between the generation expert and the refinement expert lies in their input tokens: the former decodes only the masked tokens, whereas the latter operates on generated tokens.

3.1 Diffusion LLM for Planning

Modality Tokenization As shown in Fig. 4, a single front-view image is processed by a pretrained vision tower (SigLIP [40]) to generate continuous image tokens, while a text tokenizer converts textual prompts into text tokens.

To preserve the continuity of trajectory and enable token-level refinement, we discretize the action space in discrete tokens. The spatial range $[-100m, +100m]$ is uniformly divided into 4,000 bins with a resolution of 0.05 m, shared across longitudinal and lateral axes. The heading angle range $[-90, +90]$ is discretized into 1800 bins with a resolution of 0.1. These bins are appended to the LLM vocabulary, enabling direct decoding of trajectory tokens and facilitating unified cross-modal alignment between language and actions.

Training and Inference. We first introduce how a standard dLLM is applied to the autonomous driving trajectory planning task. During training, clean sequences are corrupted by random masking, where tokens are replaced by a special mask token [M] with probability t, and supervised via masked cross-entropy loss:

$$\mathcal{L}_\theta = -\mathbb{E}_{t,p_0,r_0,r_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbb{I}\big[r_t^i = [\mathrm{M}]\big]\log p_\theta\big(r_0^i \mid p_0, r_t\big)\right] \tag{1}$$

Here, L, p_0, r_0 and r_t denote the sequence length, contextual token, original sequence, and the noised sequence, respectively.

During the inference stage, the policy model is encouraged to condition on the given inputs (visions and instructions) and progressively reconstruct a feasible trajectory from fully masked trajectory via several iterative unmasking steps. Specifically, at denoising step t, all masked tokens of noised trajectory r_t are decoded in parallel. A subset of mask tokens is then decoded to sequence r_{t-1} for the next iteration.

3.2 Block-level MoE for Refinement

As our analysis in sec. 1, the inherent irreversible decoding of dLLMs prevents them from self-refining trajectories like diffusion policies. Insufficient reasoning leads to suboptimal trajectory quality, significantly limiting the planner's performance potential. How can an explicit refinement mechanism be injected into the dLLM?

A straightforward approach is to introduce additional MoE layers to enable adaptive learning, and compute loss directly on the unmasked tokens for supervision. However, this departs from the training and inference behaviors of pretrained dLLMs, undermines their foundational capabilities, as they learn only to decode the mask tokens. Moreover, the deep coupling between generation and refinement introduces mutual interference and hinders task-specific tuning and optimization.

In reality, despite their different objectives, generation and refinement share a high similarity in contextual representation. Motivated by this insight and get inspired by Mixture-of-Transformers [20,29], we propose our block-level Mixture-of-Experts (block-MoE) together with a carefully designed training–inference pipeline.

As illustrated in Fig. 4 (right), a diffusion language model (LLaDA) consists of several stacked transformer blocks. We duplicate a small number of the final n blocks as expert blocks, thereby partitioning the model into two experts responsible for generation and refinement, respectively. The preceding m blocks and visual tower are shared across both experts.

Training and Inference. During inference, the model is explicitly prompted to perform a specific task. The shared blocks extract common contextual representations, following which the corresponding expert blocks are manually activated to execute either generation or refinement.

During training, the generation branch computes the loss only on masked tokens as in Eq. 1, while the refinement branch computes the loss over all tokens, with gradient flow confined to the refinement blocks. Here, the refinement expert is simply warm-started for basic decoding.

Clearly, our block-MoE completely decouples generation and refinement during training and inference. This preserves the foundational knowledge of the pretrained dLLM. Moreover, the refinement expert is plug-and-play and learns on top of the generative expert. The entire model can be trained jointly, highlighting its flexibility and transferability."

3.3 Reinforcement Finetuning

Recent studies have demonstrated the critical role of reinforcement learning in autonomous driving [25, 31, 54] both theoretically and empirically. We next provide a detailed description of our RFT pipeline. As shown in Fig. 5, we optimize the generative expert with GRPO. For the refinement expert, we specially designed an online–offline hybrid reinforcement learning approach.

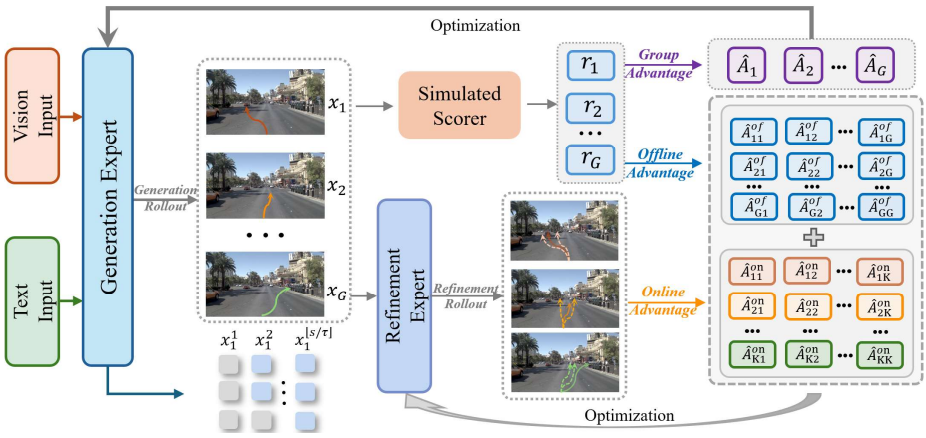

Fig. 5: Reinforcement Fine-tuning Pipeline. The offline and online advantages are combined to form a hybrid advantage that supervises the refinement expert.

GRPO for Generation Expert. For the generation expert, we employ a rule-based online reinforcement learning strategy, Group Relative Policy Optimization (GRPO). Specifically, given an arbitrary scenario p, the generation expert samples in parallel a group of G candidate trajectories $\{x_i\}_{i=1}^{G}$. Following [44], we progressively sample for s steps to ensure consistency between training and inference, and aggregate every τ neighboring steps to balance the consistency and efficiency. For each trajectory x_i, we retain $\lfloor \frac{s}{\tau} \rfloor$ sampled paths $\{x_i^j\}_{j=1}^{\lfloor s/\tau \rfloor}$. The associated rewards $\{r_i\}_{i=1}^{G}$ are then evaluated in a simulator. Subsequently, without normalization as in [50], the group-relative advantages are computed as follows:

$$\hat{A}_i = r_i - \text{mean}(\{r_i\}_{i=1}^{G}) \tag{2}$$

Subsequently, the sampled tokens are processed sequentially, and their losses are computed using the standard GRPO objective:

$$\mathcal{L}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D} \\ o_i\sim\pi_\theta(\cdot|q)}} \Bigg[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{k=1}^{|o_i|} \min\Big(\rho_{i,k}^{j}\,\hat{A}_i^k,\ \mathrm{clip}\big(\rho_{i,k}^{j},\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_i^k\Big) - \beta\, D_{\mathrm{KL}}\Big(\pi_\theta(\cdot \mid p) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid p)\Big)\Bigg], \tag{3}$$

where o_i denotes the length of the sequence, ϵ controls the clipping range, β balances the KL divergence penalty, and:

$$\rho_{i,k}^{j} = \frac{\pi_\theta\Big(o_{i,k}^{j} \mid q, x_i^{j}\Big)}{\pi_{\theta_{\mathrm{old}}}\Big(o_{i,k}^{j} \mid q, x_i^{j}\Big)} \;\Bigg|\; x_{i,k}^{j} = [\mathrm{M}],\ x_{i,k}^{j+1} \neq [\mathrm{M}]. \tag{4}$$

Hybrid RFT for Refinement Expert. The purpose of the refinement expert is to fine-tune the generated trajectories to improve their quality. Compared to the anchor trajectories to be refined, corrective trajectories that lead to score improvements should be encouraged, while those that degrade the score should be penalized, regardless of the anchor itself. Therefore, the generated trajectories naturally serve as a reference for advantage computation, eliminating the need for traditional baseline estimation in reinforcement learning (e.g., the value network in PPO or the group-average reward in GRPO). Notably, the sampling of trajectories ensures diversity, allowing them to be directly used for training the refinement expert without further modification. Specifically, for sampled trajectories $\{x_i\}_{i=1}^{G}$, we compute pairwise reward differences to obtain a relative advantage matrix of size $G \times G$. Since the sampled trajectories are produced by the generation expert, they constitute offline data for the refinement expert; hence, we define this as an offline reward matrix $\hat{\mathbf{A}}^{\mathrm{of}} \in \mathbb{R}^{G\times G}$:

$$\hat{A}_{ij}^{\mathrm{of}} = r_i - r_j, \quad \forall i, j \in G \tag{5}$$

The offline advantage matrix offers several benefits: (1) it has zero mean, simultaneously encouraging improvements and penalizing degradations; (2) the squared advantages provide denser reward signals compared to GRPO, enhancing training stability; and (3) it requires no additional sampling, making the computation simple and efficient.

Despite its many advantages, the refiner is inherently limited by the generation expert. To encourage the refinement expert to explore freely, for each generated trajectory x_i, we allow it to sample K refined trajectories $\{\hat{x}_{ik}\}_{k=1}^{K}$ online and compute their corresponding rewards $\{\hat{r}_{ik}\}_{k=1}^{K}$ simultaneously. The online reward matrix $\hat{\mathbf{A}}^{on} \in \mathbb{R}^{G\times K}$ is then calculated as follows:

$$\hat{A}_{ik}^{\mathrm{on}} = \hat{r}_{ik} - r_i, \quad \forall i \in G, k \in K \tag{6}$$

Finally, a hybrid loss is computed to optimize the refinement expert:

$$\mathcal{L}_{\text{hybrid}}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D} \\ o_i\sim\pi_\theta(\cdot|q)}} \frac{1}{G}\sum_{i=1}^{G}\left[\frac{1}{G}\sum_{j=1}^{G}\frac{1}{|o_{ij}|}\sum_{o=1}^{|o_{ij}|}\rho_{i,o}^{j}\hat{A}_{ij}^{of} + \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|o_{ik}|}\sum_{o=1}^{|o_{ik}|}\rho_{i,o}^{k}\hat{A}_{ik}^{on}\right], \tag{7}$$

In our implementation, both experts are trained synchronously: trajectories are sampled online by the generation expert and then fed to the refinement expert for refining, thereby enabling their collaborative learning and improvement.

4 Experiments

4.1 Experimental Settings

Dataset. We evaluate DriveFine on the NAVSIM [10] dataset, which is built upon nuPlan [2] (a subset of OpenScene [8]) and provides surround-view images from 8 cameras along with high-quality LiDAR point clouds. The dataset is split into 1,192 (navtrain) scenes for training and 136 scenes (navtest) for testing. NAVSIM also offers a simulation environment for closed-loop evaluation. NAVSIM provides three benchmarks: NAVSIM-v1, NAVSIM-v2, and NavHard.

NAVSIM v1 adopts the Predictive Driver Model Score (PDMS) as the evaluation metric, which is a weighted aggregation of multiple driving-related criteria, including collision avoidance, drivable area compliance, progress–risk trade-off, and comfort. Building upon this metric, NAVSIM v2 introduces the extended PDM Score (EPDMS), which incorporates additional weighted factors such as traffic light compliance, lane boundary adherence, driving direction following, and extended comfort. Navhard benchmark employs Gaussian splatting to generate scenarios beyond the training data distribution and adopts a two-stage evaluation protocol.

Implementation Details. We adopt Siglip-384 [40] as the visual tower, which partitions a single front-view image into 8 patches of size 384×384. Pretrained weights from LLaDA-8B [34] are loaded directly, with the first 28 transformer blocks serving as shared blocks and the last 4 blocks designated as expert ones. DriveFine is trained in two stages. In the first stage, it performs supervised fine-tuning (SFT) using the QA pairs and textualized trajectories provided by ReCogDrive [25], without any additional pretraining. In the second stage, it undergoes reinforcement fine-tuning (RFT) in the NAVSIM simulation environment. The generation expert and the refinement expert are trained simultaneously. The complementary masking and Prefix-DLM in [21] are adopted in our implementation for sufficient training and efficiency respectively.

During the SFT stage, the model is trained for 12 epochs with a batch size of 64, optimized using AdamW with a learning rate of 4×10^{-5} and cosine learning rate decay.

Table 1: PDMS results on the **NAVSIM-v1** benchmark. † denotes best-of-6 sampling, and * indicates the appliance of score-based reinforcement training as [18, 27].

Method	Ref.	Sensors	NC↑	DAC↑	TTC↑	C.↑	EP↑	PDMS↑
Human	–	–	100	100	100	99.9	87.5	94.8
End-to-End Methods								
UniAD [13]	CVPR'23	C	97.8	91.9	92.9	**100.0**	78.8	83.4
TransFuser [7]	TPAMI'23	C+L	97.7	92.8	92.8	**100.0**	79.2	84.0
LAW [22]	arXiv'24	C	96.4	95.4	88.7	99.9	81.7	84.6
Hydra-MDP [26]	arXiv'24	C+L	98.3	96.0	94.6	**100.0**	78.7	86.5
DiffusionDrive [30]	CVPR'25	C+L	98.2	96.2	94.7	**100.0**	82.2	88.1
WoTE [24]	ICCV'25	C+L	98.5	96.8	94.4	99.9	81.9	88.3
MeanFuser [41]	CVPR'26	C	98.6	97.0	95.0	100	82.8	89.0
Vision Language Action Methods								
AutoVLA [54]	NeurIPS'25	C	98.4	95.6	**98.0**	99.9	81.9	89.1
DriveVLA-W0 [23]	ICLR'26	C	98.7	**99.1**	95.3	99.3	83.3	90.2
AdaThinkDrive [31]	ICRA'26	C	98.4	97.8	95.2	**100**	84.4	90.3
ReCogDrive [25]	ICLR'26	C	97.9	97.3	95.2	99.8	86.7	90.4
CuriousVLA [3]	CVPR'26	C	99.1	98.8	95.2	98.1	88.5	90.3
DriveFine	-	C	98.6	98.1	95.2	99.9	85.5	90.8
DriveFine*	-	C	**98.8**	98.6	96.2	**100**	**86.9**	**91.8**
AutoVLA† [54]	NeurIPS'25	C	99.1	97.1	97.1	**100.0**	87.6	92.1
AdaThinkDrive† [31]	ICRA'26	C	99.1	98.8	97.2	**100**	87.9	93.0
DriveVLA-W0†	ICLR'26	C	**99.3**	97.4	97.0	99.9	88.3	93.0
DriveFine†		C	**99.3**	**99.2**	**97.9**	**100**	**89.1**	**94.2**

In the RFT stage, The group size for generation expert rollouts is set to 10, each of trajectory is further optimized online by the optimizer 6 times in parallel. The model is trained for 1 epoch with a batch size of 16 and a learning rate of 1×10^{-6}.

At inference, DriveFine executes 12 sampling steps. The number of refinement steps is empirically set to 1. We provide further analysis in the appendix.

4.2 Main Results

Results of Navtest. We first report the NAVSIM v1 results, evaluating PDMS on the navtest split, as shown in Tab. 1. DriveFine achieves state-of-the-art (SOTA) performance. In particular, compared with autoregressive token-based VLA models [31, 54], DriveFine achieves a 0.5 improvement, while being on par with diffusion-based planners [25]. When score-based reinforcement fine-tuning is applied (an additional scorer is trained), DriveFine further improves its performance to 91.8 PDMS, surpassing all existing VLA planners. Following prior works [23,31,54], we evaluate the best-of-N setting of DriveFine and obtain a PDMS of 94.2, demonstrating its higher performance upper bound.

Table 2: EPDMS results on the `NAVSIM-v2` benchmark. *indicates the reproduced results with released checkpoint.

Method	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	C.↑	EC↑	EPDMS↑
Ego-MLP [28]	93.1	77.9	92.7	99.6	86.0	91.5	89.4	98.3	85.4	64.0
TransFuser [7]	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
DriveSuprim [47]	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1
ARTEMIS [11]	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	–	83.1
ReCogDrive [25]	98.3	95.2	**99.5**	99.8	87.1	97.5	96.6	98.3	86.5	83.6
DriveVLA-W0 [23]	**99.0**	**98.4**	99.3	99.9	87.0	**98.1**	93.2	97.9	58.9	86.5
DiffusionDrive [30]*	98.2	96.1	99.5	99.8	87.5	97.3	96.9	98.3	**87.5**	88.1
DriveFine	**98.7**	97.3	**99.5**	99.8	**88.7**	97.8	**97.7**	**98.4**	83.8	**89.7**

Table 3: EPDMS results on the Navhard benchmark. * indicates values are copied from [38]; all other results are obtained from our own evaluations.

Method	Stage	NC↑	DAC↑	DDC↑	TLC↑	EP↑	TTC↑	LK↑	HC↑	EC↑	EPDMS↑
ReCogDrive [25]	S1	97.1	80.2	98.4	**100**	83.6	95.3	93.6	97.6	76.0	68.9
	S2	78.2	69.3	83.9	98.2	86.6	73.9	44.1	96.4	72.0	37.8
DiffusionDrive* [30]	S1	96.8	86.0	98.8	99.3	84.0	95.8	96.7	97.6	79.6	66.7
	S2	80.1	**72.8**	84.4	**98.4**	85.9	**76.6**	46.4	96.3	**72.8**	40.5
DriveFine(Ours)	S1	**97.6**	**90.0**	**99.1**	99.3	**84.9**	**96.7**	**97.3**	**97.6**	72.0	**74.4**
	S2	**82.1**	71.3	**84.8**	**98.4**	**88.1**	74.3	**47.2**	**96.8**	**72.8**	**41.0**

We report the comparison results on NAVSIM-v2 in Tab. 2, which provides a more comprehensive evaluation of the overall model performance. When evaluated with latest NAVSIM version, DriveFine further reaches 89.7 EPDMS, thereby establishing a new state of the art.

Results of Navhard. We further evaluate our method on the more challenging NavHard benchmark, which primarily targets difficult samples that lie beyond the training distribution. The results are reported in Tab. 3. Notably, no additional training is applied in this setting. Under this fair evaluation protocol, DriveFine outperforms the diffusion-based methods [30, 33] on both stage metrics, with a particularly notable improvement of 5.5 points in Stage 1 EPDMS over ReCogDrive [25], further demonstrating its strong performance as well as the superior generalization capability of token-based VLAs.

4.3 Ablation Studies

Ablations of the core components. Tab. 4 presents an ablation study on the core components of DriveFine. With LLaDA mimicking expert trajectories (SFT), the model achieves a PDMS of 86.7. GRPO reinforcement training

Table 4: Ablation results of core components of DriveFine.

ID	SFT	GRPO	Offline-RFT	Online-RFT	NC	DAC	TTC	Conf.	EP	**PDMS**
1	✓	✗	✗	✗	98.0	95.2	94.9	99.4	81.6	86.7
2	✓	✓	✗	✗	98.3	96.7	95.1	99.2	85.0	89.7$_{+3.0}$
3	✓	✓	✓	✗	98.5	97.3	95.2	99.9	85.4	90.4$_{+0.7}$
4	✓	✓	✓	✓	98.6	97.9	95.2	99.9	85.5	90.8$_{+0.4}$

Table 5: Ablation analysis of the Robustness.

Model	EP↑	EC↑	PDMS↑	EPMDS↑
DriveFine-SFT	86.8	83.3	86.7	86.8
+RFT(PDMS)	88.7	83.8	89.9$_{+3.2}$	89.1$_{+2.3}$

Table 6: Impact of different decoding strategies on planning performance.

Decoding	DAC ↑	EP ↑	TTC ↑	PDMS ↑
Adaptive	96.7	85.4	95.1	89.7
Causal	96.2	85.2	94.4	89.0
Inv-Causal	96.4	85.4	94.2	89.2

increases PDMS to 89.7. Incorporating the refinement expert with offline reinforcement training further improves performance by 0.7 point, while online reinforcement training raises the model's performance ceiling to 90.8.

We observe that the refinement mechanism improves nearly all metrics, particularly DAC and Conf. Further visualization of trajectories before and after refinement (as shown in the Fig. 6) clearly demonstrates that the refinement expert can effectively correct anomalies when individual tokens lead to collisions or off-road events, which would otherwise result in complete trajectory failure. Additionally, it significantly mitigates fluctuations caused by noncausal decoding, enhancing overall trajectory smoothness. In addition, we observe that outliers arising during the trajectory generation stage can be effectively corrected, thereby ensuring the quality of the final trajectory.

Ablation study on the robustness of DriveFine. We further evaluated the robustness of DriveFine. As shown in Tab. 5, after PDMS-oriented RFT, extended metrics such as LK, EC, and DDC remain stable or show slight improvements. Consequently, the increase in PDMS is accompanied by a synchronous improvement in EPMDS. Compared with diffusion-based models in Fig. 3, DriveFine demonstrates a clear robustness advantage, further confirming the potential of token-based models.

Impact of Decoding Order. In Tab. 6, we analyze the impact of different decoding strategies, including adaptive (confidence-based) decoding, forward causal decoding, and reverse causal decoding. As shown in the results, adaptive decoding consistently achieves the best performance compared to fixed-order decoding strategies. This observation aligns with our motivation: trajectory generation should not be constrained by a predefined order, but instead be adaptively determined by the model according to its contextual understanding.

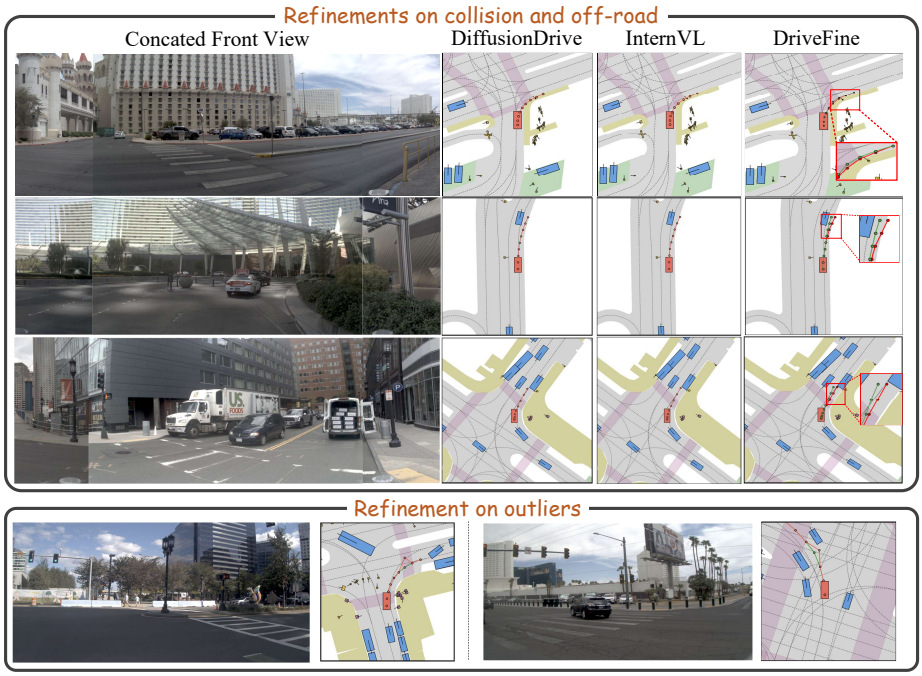

Fig. 6: Qualitative visualization comparison between DriveFine and other SOTA methods. The red and green lines denote the trajectories before and after refinement, respectively.

Efficiency-Performance Trade-off Analysis. We evaluate the impact of the number of diffusion steps s on the efficiency–performance trade-off, comparing DriveFine with VLA models of similar scale, as shown in Fig. 7. As expected, inference latency increases roughly proportionally with s. In general, a larger number of diffusion steps allows the model to "think" more thoroughly, resulting in better performance.

With 4-step generation and 1-step refinement, DriveFine achieves a PDMS of 90.51—with an average latency of 280 ms, which is significantly faster than ReCogDrive-8B [25], demonstrating an optimal efficiency–performance trade-off. These results suggest that the flexible decoding of dLLMs effectively balances efficiency and performance.

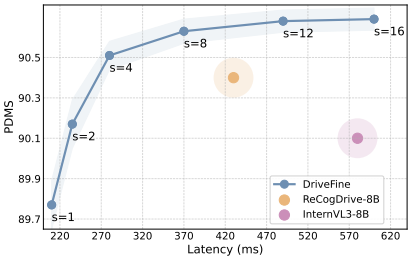

Fig. 7: PDMS-Latency trade-off

Effect of Group Size. As shown in Tab. 7, DriveFine is moderately sensitive to the group size. Even with a small group size $G = 2$, it outperforms the SFT baseline by 1.7 PDMS. Increasing G leads

Table 7: Sensitivity of group size G.

G	0	2	4	6	8	10
EP↑	81.6	83.7	84.4	84.7	85.1	85.2
DAC↑	95.2	95.8	96.3	96.5	96.9	96.8
PDMS↑	86.7	88.5	89.2	89.5	89.7	89.7

to a monotonic performance gain, with the best performance achieved at $G = 8$.

5 Conclusion

In this paper, we systematically analyze the two dominant planners–diffusion-based and token-based VLAs–highlighting their respective limitations. To address these shortcomings, we propose DriveFine, exploring masked diffusion LLMs as a unified solution. To overcome the irreversible decoding issues and further push the potential of DriveFine, we design a plug-and-play block-MoE architecture tailored with an online–offline hybrid reinforcement learning strategy to inject the refinement capability into the dLLM. The decoupling of experts prevents mutual interference, facilitating the progressive learning and improvement. Extensive experiments on NAVSIMv1, NAVSIMv2, and the more challenging Navhard benchmarks demonstrate its efficacy and robustness.

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025) 5
2. Caesar, H., Kabzan, J., Tan, K.S., Fong, W.K., Wolff, E., Lang, A., Fletcher, L., Beijbom, O., Omari, S.: nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. arXiv preprint arXiv:2106.11810 (2021) 10
3. Chen, C., Yang, Y., Tan, Z., Wang, Y., Zhan, R., Liu, H., Mao, X., Bao, J., Tang, X., Yang, L., et al.: Devil is in narrow policy: Unleashing exploration in driving vla models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1062–1072 (2026) 11
4. Chen, S., Jiang, B., Gao, H., Liao, B., Xu, Q., Zhang, Q., Huang, C., Liu, W., Wang, X.: Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. arXiv preprint arXiv:2402.13243 (2024) 1, 4
5. Chen, X., Huang, L., Ma, T., Fang, R., Shi, S., Li, H.: Solve: Synergy of language-vision and end-to-end networks for autonomous driving. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12068–12077 (2025) 5
6. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 24185–24198 (2024) 3, 5
7. Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. IEEE transactions on pattern analysis and machine intelligence **45**(11), 12878–12895 (2022) 4, 11, 12
8. Contributors, O.: Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, Vancouver, Canada. pp. 18–22 (2023) 10
9. Dang, C., Wang, J., Li, G., Hou, Z., You, Z., Ye, H., Ma, J., Chen, L., Wang, Y.: Sparseoccvla: Bridging occupancy and vision-language models via sparse queries

for unified 4d scene understanding and planning. arXiv preprint arXiv:2601.06474 (2026) 5
10. 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. Advances in Neural Information Processing Systems **37**, 28706–28719 (2024) 3, 10
11. Feng, R., Xi, N., Chu, D., Wang, R., Deng, Z., Wang, A., Lu, L., Wang, J., Huang, Y.: Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. arXiv preprint arXiv:2504.19580 (2025) 12
12. 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. arXiv preprint arXiv:2503.19755 (2025) 5
13. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17853–17862 (2023) 1, 4, 11
14. Huang, Z., Tang, T., Chen, S., Lin, S., Jie, Z., Ma, L., Wang, G., Liang, X.: Making large language models better planners with reasoning-decision alignment. In: European Conference on Computer Vision. pp. 73–90. Springer (2024) 5
15. 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) 5
16. 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: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8350 (2023) 1, 4
17. 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) 5
18. Kirby, E., Boulch, A., Xu, Y., Yin, Y., Puy, G., Zablocki, É., Bursuc, A., Gidaris, S., Marlet, R., Bartoccioni, F., et al.: Driving on registers. arXiv preprint arXiv:2601.05083 (2026) 5, 11
19. Li, J., Wu, J., Hu, D., Huang, X., Sun, B., Hao, Z., Lang, X., Zhu, X., Zhang, L.: Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. arXiv preprint arXiv:2601.05640 (2026) 2, 5
20. Li, S., Gu, J., Liu, K., Lin, Z., Wei, Z., Grover, A., Kuen, J.: Lavida-o: Elastic large masked diffusion models for unified multimodal understanding and generation. arXiv preprint arXiv:2509.19244 (2025) 3, 7
21. Li, S., Kallidromitis, K., Bansal, H., Gokul, A., Kato, Y., Kozuka, K., Kuen, J., Lin, Z., Chang, K.W., Grover, A.: Lavida: A large diffusion language model for multimodal understanding. arXiv preprint arXiv:2505.16839 (2025) 3, 10
22. Li, Y., Fan, L., He, J., Wang, Y., Chen, Y., Zhang, Z., Tan, T.: Enhancing end-to-end autonomous driving with latent world model. arXiv preprint arXiv:2406.08481 (2024) 11
23. Li, Y., Shang, S., Liu, W., Zhan, B., Wang, H., Wang, Y., Chen, Y., Wang, X., An, Y., Tang, C., et al.: Drivevla-w0: World models amplify data scaling law in autonomous driving. arXiv preprint arXiv:2510.12796 (2025) 11, 12
24. Li, Y., Wang, Y., Liu, Y., He, J., Fan, L., Zhang, Z.: End-to-end driving with online trajectory evaluation via bev world model. arXiv preprint arXiv:2504.01941 (2025) 11

25. 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. arXiv preprint arXiv:2506.08052 (2025) 2, 3, 5, 8, 10, 11, 12, 14
26. Li, Z., Li, K., Wang, S., Lan, S., Yu, Z., Ji, Y., Li, Z., Zhu, Z., Kautz, J., Wu, Z., et al.: Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. arXiv preprint arXiv:2406.06978 (2024) 1, 5, 11
27. Li, Z., Yao, W., Wang, Z., Sun, X., Chen, J., Chang, N., Shen, M., Wu, Z., Lan, S., Alvarez, J.M.: Generalized trajectory scoring for end-to-end multimodal planning. arXiv preprint arXiv:2506.06664 (2025) 5, 11
28. Li, Z., Yu, Z., Lan, S., Li, J., Kautz, J., Lu, T., Alvarez, J.M.: Is ego status all you need for open-loop end-to-end autonomous driving? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14864–14873 (2024) 12
29. Liang, W., Yu, L., Luo, L., Iyer, S., Dong, N., Zhou, C., Ghosh, G., Lewis, M., Yih, W.t., Zettlemoyer, L., et al.: Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. arXiv preprint arXiv:2411.04996 (2024) 3, 7
30. 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: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12037–12047 (2025) 2, 5, 11, 12
31. 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) 2, 3, 5, 8, 11
32. Mao, J., Qian, Y., Ye, J., Zhao, H., Wang, Y.: Gpt-driver: Learning to drive with gpt. arXiv preprint arXiv:2310.01415 (2023) 5
33. Nie, M., Peng, R., Wang, C., Cai, X., Han, J., Xu, H., Zhang, L.: Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In: European Conference on Computer Vision. pp. 292–308. Springer (2024) 5, 12
34. Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.R., Li, C.: Large language diffusion models. arXiv preprint arXiv:2502.09992 (2025) 3, 10
35. 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) 2, 5
36. Sun, W., Lin, X., Shi, Y., Zhang, C., Wu, H., Zheng, S.: Sparsedrive: End-to-end autonomous driving via sparse scene representation. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 8795–8801. IEEE (2025) 4
37. Tan, T., Zheng, Y., Liang, R., Wang, Z., Zheng, K., Zheng, J., Li, J., Zhan, X., Liu, J.: Flow matching-based autonomous driving planning with advanced interactive behavior modeling. arXiv preprint arXiv:2510.11083 (2025) 5
38. Tian, H., Li, T., Liu, H., Yang, J., Qiu, Y., Li, G., Wang, J., Gao, Y., Zhang, Z., Wang, L., et al.: Simscale: Learning to drive via real-world simulation at scale. arXiv preprint arXiv:2511.23369 (2025) 12
39. Tian, X., Gu, J., Li, B., Liu, Y., Wang, Y., Zhao, Z., Zhan, K., Jia, P., Lang, X., Zhao, H.: Drivevlm: The convergence of autonomous driving and large vision-language models. arXiv preprint arXiv:2402.12289 (2024) 5
40. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: Siglip 2:

Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786 (2025) 3, 6, 10
41. Wang, J., Liu, X., Zheng, Y., Xing, Z., Li, P., Li, G., Ma, K., Chen, G., Ye, H., Xia, Z., et al.: Meanfuser: Fast one-step multi-modal trajectory generation and adaptive reconstruction via meanflow for end-to-end autonomous driving. arXiv preprint arXiv:2602.20060 (2026) 11
42. 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: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22442–22452 (2025) 5
43. Wang, T., Xie, E., Chu, R., Li, Z., Luo, P.: Drivecot: Integrating chain-of-thought reasoning with end-to-end driving. arXiv preprint arXiv:2403.16996 (2024) 5
44. Wang, Y., Yang, L., Li, B., Tian, Y., Shen, K., Wang, M.: Revolutionizing reinforcement learning framework for diffusion large language models. arXiv preprint arXiv:2509.06949 (2025) 8
45. Xing, Z., Zhang, X., Hu, Y., Jiang, B., He, T., Zhang, Q., Long, X., Yin, W.: Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1602–1611 (2025) 5
46. Xu, M., Cui, J., Cai, F., Shang, H., Zhu, Z., Luan, S., Xu, Y., Zhang, N., Li, Y., Cai, J., et al.: Wam-diff: A masked diffusion vla framework with moe and online reinforcement learning for autonomous driving. arXiv preprint arXiv:2512.11872 (2025) 5
47. Yao, W., Li, Z., Lan, S., Wang, Z., Sun, X., Alvarez, J.M., Wu, Z.: Drivesuprim: Towards precise trajectory selection for end-to-end planning. arXiv preprint arXiv:2506.06659 (2025) 12
48. Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., Kong, L.: Dream 7b: Diffusion large language models. arXiv preprint arXiv:2508.15487 (2025) 3
49. You, Z., Nie, S., Zhang, X., Hu, J., Zhou, J., Lu, Z., Wen, J.R., Li, C.: Llada-v: Large language diffusion models with visual instruction tuning. arXiv preprint arXiv:2505.16933 (2025) 3
50. Zhao, S., Gupta, D., Zheng, Q., Grover, A.: d1: Scaling reasoning in diffusion large language models via reinforcement learning. arXiv preprint arXiv:2504.12216 (2025) 8
51. Zheng, W., Song, R., Guo, X., Zhang, C., Chen, L.: Genad: Generative end-to-end autonomous driving. In: European Conference on Computer Vision. pp. 87–104. Springer (2024) 4
52. Zheng, Y., Liang, R., Zheng, K., Zheng, J., Mao, L., Li, J., Gu, W., Ai, R., Li, S.E., Zhan, X., et al.: Diffusion-based planning for autonomous driving with flexible guidance. arXiv preprint arXiv:2501.15564 (2025) 5
53. Zhou, X., Han, X., Yang, F., Ma, Y., Tresp, V., Knoll, A.: Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. arXiv preprint arXiv:2503.23463 (2025) 2, 5
54. 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. arXiv preprint arXiv:2506.13757 (2025) 2, 3, 5, 8, 11
55. 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) 1

56. Zou, J., Chen, S., Liao, B., Zheng, Z., Song, Y., Zhang, L., Zhang, Q., Liu, W., Wang, X.: Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. arXiv preprint arXiv:2512.07745 (2025) 2, 3, 5