MINDDRIVE: A MAP-INFORMED ADAPTIVE DECISION METHOD FOR EFFICIENT END-TO-END DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in Vision-Language-Action (VLA) models have demonstrated the potential of combining multimodal understanding, reasoning, and trajectory generation for end-to-end autonomous driving. However, existing methods often infer traffic context directly from multi-view images, making them susceptible to hallucinations about planning-critical scene elements. Moreover, their reasoning strategies are typically insensitive to scene complexity, leading to insufficient deliberation in challenging scenarios and unnecessary reasoning overhead in routine ones. In this paper, we propose MindDrive, a **M**ap-**In**formed Adaptive **D**ecision VLA model for efficient driving. First, we design a language-map construction and retrieval pipeline that converts structured map elements into spatially indexed natural-language annotations, providing accurate, decision-relevant scene priors. Then, during supervised fine-tuning stage, the model learns to assess scenario complexity and accordingly perform either direct action generation or deliberative reasoning. Finally, we introduce a reinforcement learning-based post-training strategy that contrasts different decision modes within the same scene, further improving planning performance and reasoning efficiency. Extensive experiments on NAVSIM and nuScenes benchmarks demonstrate the competitive performance of MindDrive.

1 INTRODUCTION

End-to-end autonomous driving, which learns unified policies that map sensory observations directly to planning trajectories, has advanced rapidly (Chen et al., 2024). Early studies jointly optimize perception, prediction, and planning within a differentiable framework, reducing the reliance on manually designed interfaces between individual modules (Hu et al., 2022; Chitta et al., 2022; Shao et al., 2023; Jia et al., 2023). Subsequent efforts have further improved planning efficiency and trajectory quality through sparse scene representations and generative trajectory modeling (Hu et al., 2023; Jiang et al., 2023; Sun et al., 2025; Liao et al., 2025). More recently, advances in vision-language models (Zhou et al., 2025) have extended this paradigm to vision-language-action (VLA) driving, a formulation first established by general-purpose VLA agents for embodied control (Kim et al., 2025; Black et al., 2024; Zhao et al., 2025). By coupling pretrained multimodal representations with action generation, VLA models can integrate visual observations and navigation instructions, perform planning-oriented reasoning, and generate future trajectories within a unified architecture (Jiang et al., 2025b). Whether serving as an interpretable reasoning module that assists a conventional planner (Tian et al., 2024; Wang et al., 2023; Jiang et al., 2026; Chen et al., 2025), or directly generating driving decisions and planning trajectories end-to-end (Hwang et al., 2024; Wang et al., 2025; Zhou et al., 2026a;b), VLM-based driving systems have achieved remarkable progress, enhancing the flexibility, context-awareness, and interpretability of end-to-end autonomous driving (Jiang et al., 2025b).

Despite recent progress, existing VLA driving models still face two key limitations: 1) *Limited grounding in planning-relevant physical context.* General-purpose vision-language representations often lack the precise spatial grounding required for reliable driving decisions. Therefore, some methods incorporate additional perception modules to compensate for the limited understanding of 3D scene structure and semantic information (Tian et al., 2024; Zhou et al., 2026a; Wang et al., 2023; Jiang et al., 2026). Although effective, these approaches require additional online perception and feature-alignment components to recover planning-relevant structures from sensor observations.

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

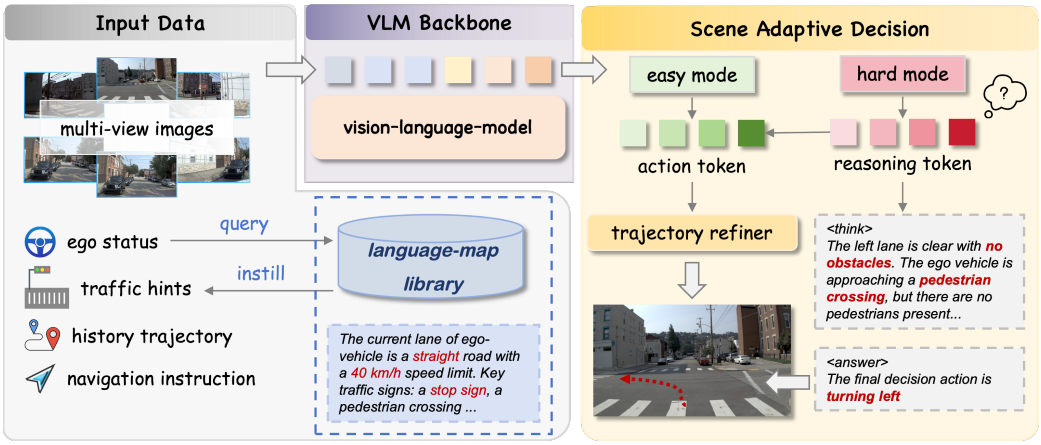


Figure 1: Overview of MindDrive. Given multi-view images, ego status, historical trajectory, and navigation instructions as input, MindDrive first retrieves decision-relevant descriptions of the surrounding environment from the offline language-map library. All input data is fed into vision-language model to generate the fused scene representation. Then, the model performs scene-adaptive decision making by generating mode tokens that determine whether to invoke an easy mode for direct action prediction or a hard mode for deliberative reasoning.

Meanwhile, static yet decision-critical information, such as lane-level speed limits, crosswalk locations, and traffic-control infrastructure, are either invisible to cameras or require rule-level knowledge that cannot be inferred from bird's-eye-view(BEV) features or 3D queries alone. 2) *Inefficient and inflexible reasoning across diverse scenarios.* General driving VLA models employ a fixed generation strategy, either directly predicting actions or consistently producing explicit chain-of-thought reasoning (Hwang et al., 2024; Shao et al., 2024a; Zhou et al., 2026a; Xing et al., 2025; Wang et al., 2024; Nie et al., 2024). Direct action generation may provide insufficient deliberation in challenging situations, while overlong reasoning introduces redundant descriptions and unnecessary latency in routine scenarios (Ma et al., 2025). Moreover, conventional supervised fine-tuning assigns reasoning patterns based on predefined annotations, while existing reinforcement post-training typically optimizes final planning performance without explicitly evaluating whether deliberation is beneficial for each scene (Wang et al., 2025; Zhou et al., 2026a;b; Yuan et al., 2026; Jiang et al., 2025a; Xiong et al., 2026).

To address these limitations, we propose MindDrive, a map-informed adaptive decision method for efficient driving, as illustrated in Figure 1. MindDrive retrieves concise natural-language descriptions of the local traffic context from an offline map library according to the ego pose, providing planning-relevant priors. To adapt the reasoning process to diverse driving scenarios, we develop a two-stage training framework that equips the VLA model with scene-complexity awareness. During supervised fine-tuning, supervision derived from teacher-generated reasoning enables the model to learn complementary direct-action and deliberative planning capabilities. Subsequently, a GRPO-based reinforcement fine-tuning strategy jointly improves planning performance and efficiency. As a result, MindDrive focuses on decision-critical context and performs explicit reasoning only when it is beneficial to planning. Our main contributions are summarized as follows:

- We design an offline language-map library construction pipeline that supports retrieving local planning-relevant road information quickly to provide language-native spatial priors.
- We propose an adaptive-decision VLA framework trained with a two-stage pipeline, which adapts its reasoning depth to the complexity of each scenario.
- The experiments conducted on benchmarks NAVSIM and nuScenes achieved a PDM score of 90.0 and the average L2 error of 0.51, respectively, demonstrating the effectiveness of MindDrive.

2 Related Work

2.1 End-to-end Autonomous Driving

A line of end-to-end autonomous driving systems aims to integrate perception, prediction, and planning into a unified differentiable framework, mapping sensor observations to future ego trajectories. UniAD (Hu et al., 2023) introduces a planning-oriented full-stack framework that jointly models object detection and tracking, motion forecasting, occupancy prediction, and ego planning, with task-specific supervision designed to benefit the final planning objective. VAD (Jiang et al., 2023) further improves efficiency by representing agents and map elements as vectorized instances, thereby reducing the dependence on dense rasterized scene representations. SparseDrive (Sun et al., 2025) reduced computational overhead through applying sparse query-based representations. More recently, DiffusionDrive (Liao et al., 2025) formulates trajectory planning as a conditional denoising process and employed a truncated diffusion decoder to improve inference efficiency.

Another emerging direction seeks to align motion planning with human driving preferences rather than relying solely on imitation losses or hand-crafted planning metrics. TrajHF (Li et al., 2025) introduces a human feedback-driven finetuning framework for trajectory generation. DRAMA (Yuan et al., 2024) introduces a Mamba-based end-to-end motion planner that efficiently reduces sequence modeling complexity. Hydra-MDP (Li et al., 2024b) introduces a multi-teacher distillation framework that transfers knowledge from both human and rule-based teachers to a multi-head student decoder. DriveDPO (Shang et al., 2026) further formulates trajectory optimization as a pairwise preference learning problem and applies direct preference optimization to align planned trajectories with preferred driving behaviors.

Despite their architectural diversity, these methods rely on purely imitative, appearance-level cues and offer no interpretable rationale for their decisions, motivating end-to-end planners combined with the world knowledge and explicit reasoning of large vision-language models.

2.2 Vision-language-action Models for Autonomous Driving

Vision-language models have been incorporated into autonomous driving through two main paradigms (Yang et al., 2023). The first integrates a VLM as an auxiliary reasoning module into a conventional planner (Li et al., 2024a). DriveMLM (Wang et al., 2023) performs interpretable multimodal behavior planning by mapping language decisions to standardized vehicle control states. DriveVLM (Tian et al., 2024) decomposes the driving stack into a slow VLM-based reasoner and a fast trajectory planner. Senna (Jiang et al., 2026) bridges a large vision-language model with a conventional end-to-end planner through semantic decision supervision. SOLVE (Chen et al., 2025) enhances an end-to-end planner with asynchronously generated VLM forecasts, while ReCogDrive (Xiong et al., 2026) couples an autoregressive VLM with a diffusion-based planner to enable stable and precise trajectory generation.

The second paradigm casts autonomous driving as a vision-language-action problem, in which multimodal models directly generate planning outputs from visual observations and navigation instructions. Early attempts such as GPT-Driver (Mao et al., 2023) and DriveGPT4 (Xu et al., 2024) explored language-model-based formulations of driving understanding and planning. EMMA (Hwang et al., 2024) demonstrated that a large multimodal model can directly map raw camera observations, navigation instructions, and ego states to language-represented planning trajectories. SimLingo (Renz et al., 2025) achieved vision-only closed-loop driving through language-action alignment. OmniDrive (Wang et al., 2025) introduced a holistic vision-language dataset with counterfactual reasoning, providing dense supervision to bridge language-based reasoning and trajectory planning. OpenDriveVLA (Zhou et al., 2026a) further introduced hierarchical alignment between language embeddings and 2D–3D instance-level visual tokens, enabling spatially grounded trajectory generation, and ORION (Fu et al., 2025) integrated a generative planner into the VLA architecture to improve trajectory feasibility. OpenEMMA (Xing et al., 2025) showed that explicit chain-of-thought reasoning consistently improves planning performance across a diverse set of open-source multimodal backbones. More recently, AutoVLA (Zhou et al., 2026b) discretized continuous trajectories into feasible action tokens, unifying semantic reasoning and trajectory generation within a single autoregressive model. AutoDrive-R^2 (Yuan et al., 2026) constructed a four-step chain-of-thought dataset with explicit self-reflection and further employed GRPO with physics-grounded rewards.

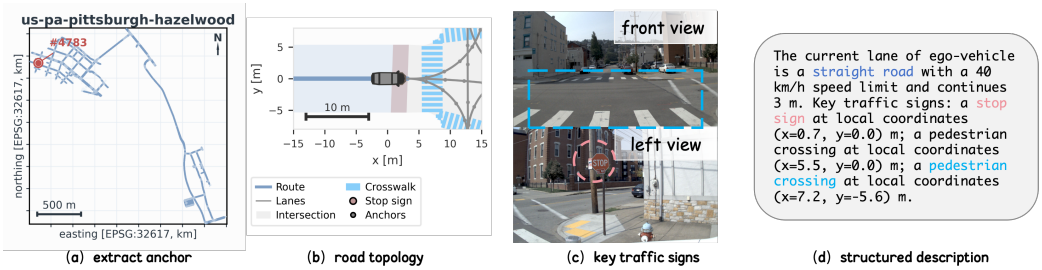


Figure 2: The pipeline of building offline language-map library. (a) Extract anchors from global city map. (b) Target anchor's local map visualization in bird-eye-view. (c) Key traffic signs in camera views. (d) Generate structured map description.

UniWorldVLA (Liu et al., 2026) further interleaved world modeling and planning by alternately predicting future frames and ego actions, allowing each planning step to be conditioned on newly imagined future observations.

Beyond planning quality, recent work has also sought to improve the efficiency of driving VLAs. FastDriveVLA (Cao et al., 2026) trains a reconstruction-based token pruner to preserve foreground-relevant visual tokens, while DriveMoE (Yang et al., 2026) learns a vision router that selects context-relevant camera views prior to backbone processing and skips the remaining views entirely.

3 MindDrive

As illustrated in Figure 1, MindDrive takes multi-view image sequences, the ego-vehicle state, and a high-level driving instruction as input, and predicts the future ego trajectory in an autoregressive manner. Given the current ego pose, we query an offline language-map library to retrieve a concise textual description of the local road context (Section 3.1). The visual observations and textual context are then jointly processed by the Qwen2.5-VL-3B backbone (Team, 2025). Before trajectory generation, the model predicts a mode token that adaptively determines the inference strategy according to the complexity of the driving scenario. For simple scenarios, MindDrive directly generates planning waypoints, whereas for challenging scenarios, it first performs deliberative chain-of-thought reasoning before trajectory generation. We adopt a two-stage training paradigm consisting of supervised fine-tuning (SFT) and reinforcement fine-tuning (RFT). The SFT stage equips the model with preliminary scene understanding and adaptive decision-making capabilities (Section 3.3). In the subsequent RFT stage, we further optimize the model with a pairwise rollout-based reward design, encouraging it to distinguish safer and more goal-consistent trajectories from inferior alternatives (Section 3.4).

3.1 Offline Language-Map Library Construction Pipeline

To enable efficient and interpretable map-based reasoning, we construct an *Offline Language-Map Library (OLML)* through the following pipeline:

Offline Map Elements Preprocessing. For each map, we first extract a set of *anchors* from all lanes and lane connectors. We uniformly sample anchors along the lane centerline at an interval of δ_s meters, where each anchor is defined by its position and heading:

$$\mathbf{a}_k = (\mathbf{p}_k, \theta_k) \tag{1}$$

For each anchor $\mathbf{a}_k$, we extract the surrounding map elements within a predefined region with radius R and transform them into an anchor-centric coordinate system, with the anchor position as the origin and its heading defining the longitudinal direction. This canonical representation allows map information to be described consistently regardless of the anchor's global position and orientation.

The extracted map elements are then organized according to their relevance to local navigation. For the lane on which the anchor is located, we retain attributes such as the speed limit, remaining lane length, and downstream geometry (e.g., straight or turning). For neighboring lanes, only the

closest lanes on the left and right are retained. Similarly, for pedestrian-related and traffic-control elements, such as crosswalks, traffic lights, and stop signs, we retain the closest relevant element ahead along the current lane. Their relative distances and directions with respect to the anchor are also recorded. This selection strategy produces a compact representation that emphasizes road and traffic information most relevant to local driving decisions. The resulting language-map library is defined as

$$\mathcal{D}_{\text{map}} = \{(\mathbf{a}_k, \mathcal{T}_k)\}_{k=1}^{N_A} \tag{2}$$

where $\mathcal{T}_k$ denotes the local map description associated with anchor $\mathbf{a}_k$ and N_A stands for total number of anchors.

Online Retrieval. At runtime, the preprocessed language-map library is queried according to the ego vehicle's current position to retrieve the most relevant geographic description. Given the ego position $\mathbf{p}_{\text{ego}}$, we employ a KD-Tree to efficiently search for the best-matched anchor, defined as the anchor whose position has the minimum Euclidean distance to the ego vehicle:

$$\mathbf{a}^* = \arg\min_{\mathbf{a}_i \in \mathcal{A}} |\mathbf{p}_{\text{ego}} - \mathbf{p}_i|_2 \tag{3}$$

where $\mathcal{A}$ denotes the set of all anchors in the language-map library. The KD-Tree enables efficient nearest-neighbor search with an average lookup complexity of $\mathcal{O}(\log N_A)$. The map description $\mathcal{T}^*$ associated with the retrieved anchor $\mathbf{a}^*$ is then used as the static geographic prior for the current ego state.

3.2 Chain-of-Thought Dataset Generation

We first employ Qwen2.5-VL-72B model (Team, 2025) as a teacher model to generate chain-of-thought (CoT) annotations for all samples in the dataset. For each sample, the teacher model takes historical multi-view images, the ego vehicle's historical trajectory, navigation instructions, and local map description as input, and produces structured reasoning that captures the key factors underlying the driving decision. Specifically, we prompt the teacher model to reason in three stages: *1) identifying critical dynamic agents*, *2) attending to relevant traffic signals with the aid of local map information*, and *3) generating an explainable driving decision*. This procedure provides explicit supervision that connects multimodal scene understanding with the final driving decision.

Based on the generated CoT annotations, we further assign a scenario complexity label to each sample for subsequent supervised fine-tuning. To reduce unnecessary teacher-model inference on trivial driving scenarios, we adopt a two-stage labeling strategy. First, the kinematics-based rule is used to identify simple scenarios: samples in which the ego vehicle travels straight at an approximately constant speed or remains nearly stationary are directly labeled as *easy*. For the remaining samples, which typically involve more complex driving behaviors, we also employ the teacher model to assess scenario complexity based on the generated CoT annotations.

For the i-th scenario, the teacher VLM evaluates its complexity along three dimensions: 1) traffic participant density s_i^{dens}, 2) environmental complexity s_i^{env}, and 3) decision-making complexity s_i^{dec}. These scores are then combined to obtain an overall scenario complexity score:

$$S_i = \lambda s_i^{\text{dens}} + \beta s_i^{\text{env}} + \gamma s_i^{\text{dec}} \tag{4}$$

where λ, β, and γ are parameters that determine the relative contributions of the three dimensions.

We then derive binary difficulty labels by designating the top $K\%$ of scenarios in the overall score distribution as `<hard_mode>` and the remaining scenarios as `<easy_mode>`. In total, we obtain approximately 52.6K hard samples and 69.7K easy samples.

3.3 Supervised Fine-tuning Training

Let F denote the multi-view image sequence, H_{traj} the historical ego trajectory, M_{local} the retrieved local map description, and I the navigation instruction. The multimodal input is represented as

$$X = (F, H_{\text{traj}}, M_{\text{local}}, I) \tag{5}$$

Given the multimodal context X, the VLM first judges the scenario as either `<easy_mode>` or `<hard_mode>`. For hard scenarios, the model is trained to generate an explicit CoT reasoning

Table 1: Evaluation results on the NAVSIM benchmark. The first block lists end-to-end planners and the second block lists VLM-based methods. "3+1" denotes the use of three historical frames and the current frame.

Method	Sensor	Cams	Frames	Sub-metrics						
				NC ↑	DAC ↑	DD ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
TransFuser (Chitta et al., 2022)	Cam+LiDAR	3	1	97.7	92.8	–	79.2	92.8	100.0	84.0
DRAMA (Yuan et al., 2024)	Cam+LiDAR	3	1	98.0	93.1	–	80.1	94.8	100.0	85.5
DriveDPO (Shang et al., 2026)	Cam+LiDAR	3	3+1	98.5	98.1	–	84.3	94.8	99.9	90.0
Hydra-MDP (Li et al., 2024b)	Cam+LiDAR	3	3+1	98.3	96.0	–	78.7	94.6	100.0	86.5
Centaur (Sima et al., 2025)	Cam+LiDAR	1	3+1	**99.5**	**98.9**	–	85.9	98.0	100.0	92.6
TrajHF (Li et al., 2025)	Cam+LiDAR	3	1	96.6	96.6	–	84.5	92.1	100.0	87.6
AutoVLA (RFT) (Zhou et al., 2026b)	Cam	3	3+1	98.4	95.6	95.4	81.9	98.0	99.9	89.1
UniWorldVLA (Liu et al., 2026)	Cam	1	3+1	98.7	96.7	–	83.2	96.1	100.0	89.4
ReCogDrive (Xiong et al., 2026)	Cam	8	1	97.9	97.3	–	87.3	94.9	100.0	**90.8**
MindDrive (SFT)	Cam	3	3+1	98.4	91.6	95.3	79.1	93.7	100.0	83.0
MindDrive (RFT)	Cam	3	3+1	99.4	98.8	98.2	**86.4**	91.8	100.0	90.0

sequence before trajectory prediction, while easy scenarios directly proceed to planning without explicit reasoning. Thus, the model learns to adapt its reasoning process according to scenario complexity rather than performing costly explicit reasoning for every sample. The planning waypoint sequence generated by the VLM is denoted as $\mathbf{T}_{\text{init}}$. We further introduce a trajectory refiner that takes the final-layer features of the VLM as input and learns a residual correction term between the predicted trajectory $\mathbf{T}_{\text{init}}$ and the ground-truth trajectory. The final refined trajectory is denoted as $\mathbf{T}^{\text{ref}}$.

The overall training loss consists of three components. We employ an autoregressive cross-entropy loss $\mathcal{L}_{\text{cot}}$ for CoT generation and a binary classification loss $\mathcal{L}_{\text{mode}}$ for scenario mode prediction. For trajectory planning, we use the L_2 loss to supervise the refined trajectory.

$$\mathcal{L}_{\text{sft}} = \lambda_{\text{cot}}\mathcal{L}_{\text{cot}} + \lambda_{\text{mode}}\mathcal{L}_{\text{mode}} + \lambda_{\text{traj}}\mathcal{L}_{\text{traj}} \tag{6}$$

where the λ terms are weighting coefficients that balance the three objectives of supervision.

3.4 Reinforcement Learning Post-Training

Although supervised fine-tuning equips the model with reasoning and trajectory-generation capabilities, it does not directly optimize the quality of the resulting driving behavior. We therefore refine the SFT policy using reinforcement post-training with verifiable planning rewards. For each scene, we construct a group of paired rollouts by forcing the model to follow either the direct-action mode or the deliberative mode. All rollouts are decoded into future trajectories and evaluated using the Predictive Driver Model Score (PDMS) (Dauner et al., 2024). Since legitimate reasoning lengths may vary across scenarios, we apply a penalty only to completions that exceed a predefined length threshold. Following GRPO (Shao et al., 2024b), the reward is normalized within each rollout group to obtain the relative advantage:

$$r^{(g)} = R_{\text{PDMS}}\left(o^{(g)}\right) - \lambda_{\text{len}}\mathbb{I}\left[|o^{(g)}| > L_{\max}\right], \qquad A^{(g)} = \frac{r^{(g)} - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}} + \epsilon} \tag{7}$$

where R_{PDMS} denotes the PDMS-based planning reward, λ_{len} is the length-penalty coefficient, $o^{(g)}$ denotes the g-th rollout, $L_{\max}$ is the maximum allowed completion length. And $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are the mean and standard deviation of rewards within the group. This relative formulation requires no additional value network and allows direct and deliberative responses to be compared under the same driving condition.

Inspired by DAPO (Yu et al., 2026), we remove the reference model and the associated KL regularization. Instead of averaging the loss independently for each response, we optimize all valid

completion tokens jointly:

$$\mathcal{L}_{\mathrm{RFT}} = -\frac{\sum_{g=1}^{G}\sum_{t=1}^{|o^{(g)}|} m_t^{(g)} \exp\Big(\ell_t^{(g)} - \mathrm{sg}\Big[\ell_t^{(g)}\Big]\Big) A^{(g)}}{\sum_{g=1}^{G}\sum_{t=1}^{|o^{(g)}|} m_t^{(g)}} \tag{8}$$

where $\ell_t^{(g)} = \log \pi_\theta(o_t^{(g)} \mid x, o_{<t}^{(g)})$, x denotes the multimodal input of the current scene, $m_t^{(g)}$ is the completion mask, and $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. This token-level normalization assigns equal weight to each valid token and avoids response-level normalization biases caused by the substantial length difference between direct and deliberative outputs.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on the NAVSIM (Dauner et al., 2024) and nuScenes (Caesar et al., 2020) datasets. NAVSIM contains 103,288 training samples and 12,146 test samples, while nuScenes contains 19,030 training samples and 5,569 validation samples. For both datasets, we use images from the front, front-left, and front-right cameras as visual inputs. The model takes four historical frames sampled at 2 Hz as input and predicts a planned trajectory for the subsequent 4 seconds.

Evaluation Metrics. On NAVSIM, we evaluate planning performance using No At-Fault Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time-to-Collision Within Bound (TTC), Comfort (C), Driving Direction Compliance (DD), and the aggregated metric Predictive Driver Model Score (PDMS). On nuScenes, following ST-P3 (Hu et al., 2022), we report the L2 trajectory error and collision rate.

Training Details. During supervised fine-tuning, we jointly train the model on a mixture of the NAVSIM and nuScenes datasets. We freeze the visual encoder while optimizing the multimodal projector and the LLM backbone. The model is trained for 10 epochs on 16 NVIDIA A100 GPUs with a batch size of 6, an initial learning rate of 2.0×10^{-5}, and a gradient accumulation factor of 2. Specifically, we conduct RFT solely on NAVSIM and subsequently transfer the resulting policy to nuScenes. We only update the parameters of the trajectory refiner on nuScenes for evaluation. We train the policy for 5,000 steps on eight NVIDIA A100 GPUs, with one sample per worker at each step, a learning rate of 4.0×10^{-5}, weight decay of 0.01, and gradient norm clipping with a threshold of 1.0.

4.2 Main results

NAVSIM Results. Table 1 compares MindDrive with two groups of baselines on the NAVSIM benchmark, including end-to-end planners and recent VLM-based methods. After reinforcement fine-tuning, the model's planning ability has been significantly improved. Compared with end-to-end planning baselines, MindDrive (RFT) obtains a PDMS score of 90.0, demonstrating the effectiveness of introducing vision-language reasoning into end-to-end motion planning. Among VLM-based methods, MindDrive also shows competitive performance. However, MindDrive is still slightly behind ReCogDrive in terms of PDMS, with a gap of 0.8 points. This is mainly because ReCogDrive adopts a VLM backbone that is further pretrained on richer driving-oriented VQA data, which provides stronger domain-specific visual reasoning and scene comprehension capabilities.

nuScenes Results. As shown in Table 2, RFT brings the most notable improvement in collision avoidance. Compared with the SFT model, MindDrive (RFT) reduces the average collision rate from 0.14% to 0.08%, achieving the lowest average collision rate among methods with reported collision results. This indicates that RFT effectively encourages safer trajectory generation, particularly for longer-horizon planning. Nevertheless, MindDrive is less competitive than end-to-end planners in terms of L2 error.

Table 2: Evaluation results on the nuScenes benchmark. L2 error (m) and collision rate (%) are reported at the 1s, 2s, and 3s horizons following the ST-P3 convention.

Method	Cams	Frames	L2 (m) ↓				Col. (%) ↓			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
ST-P3 (Hu et al., 2022)	6	3+1	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD (Hu et al., 2023)	6	1	0.44	0.67	0.96	0.69	0.04	0.08	0.23	0.12
VAD (Jiang et al., 2023)	6	1+1	0.17	0.34	0.60	0.37	0.07	0.10	0.24	0.14
DriveVLM (Tian et al., 2024)	6	3+1	0.18	0.34	0.68	0.40	0.10	0.22	0.45	0.27
SOLVE-E2E (Chen et al., 2025)	6	1	0.14	0.28	0.50	0.31	0.04	0.17	0.68	0.30
EMMA (Hwang et al., 2024)	6	1	0.14	0.29	0.54	0.32	–	–	–	–
OmniDrive (Wang et al., 2025)	6	1	0.14	0.29	0.55	0.33	0.00	0.13	0.78	0.30
FastDriveVLA (Cao et al., 2026)	6	1	0.14	0.29	0.54	0.32	0.00	0.15	0.60	0.25
DriveMoE (Yang et al., 2026)	2	1+1	0.45	0.70	1.08	0.74	0.11	0.15	0.26	0.17
OpenDriveVLA-3B (Zhou et al., 2026a)	6	1	0.14	0.30	0.55	0.33	0.02	0.07	0.22	0.10
OpenEMMA (Xing et al., 2025)	6	1	1.45	3.21	3.76	2.81	–	–	–	–
Impromptu-VLA (Chi et al., 2026)	6	1	0.13	0.27	0.52	0.30	–	–	–	0.24
AutoVLA (Zhou et al., 2026b)	3	3+1	0.22	0.39	0.61	0.41	0.10	0.17	0.28	0.18
AutoDrive-R^2 (Yuan et al., 2026)	6	1	0.35	0.49	0.62	0.49	0.13	0.44	0.87	0.48
MindDrive (SFT)	3	3+1	0.33	0.64	1.03	0.67	0.00	0.09	0.34	0.14
MindDrive (RFT)	3	3+1	0.21	0.48	0.83	0.51	0.00	0.04	0.18	0.08

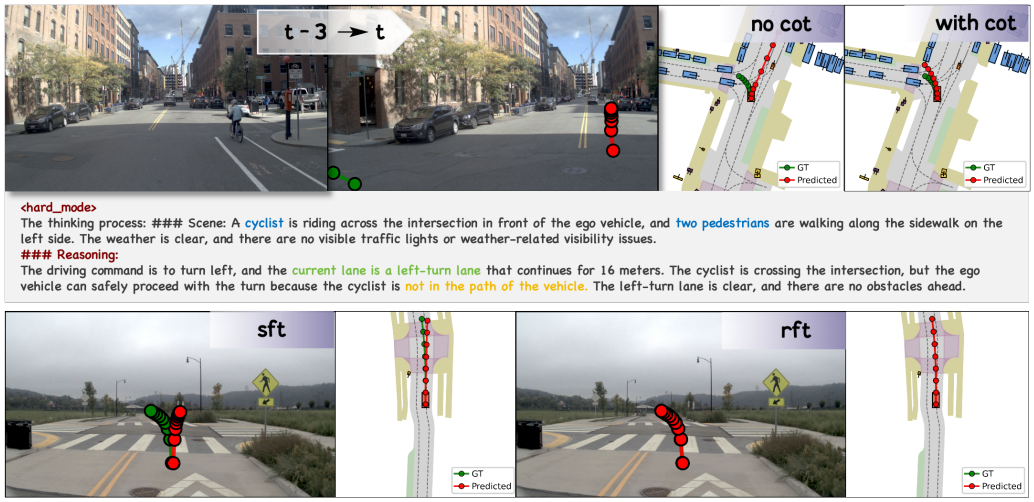


Figure 3: Qualitative comparison of planned trajectories under different conditions. The top row compares the trajectories generated by models with and without chain-of-thought (CoT) reasoning, while the bottom row compares the trajectories before and after reinforcement fine-tuning (RFT).

4.3 Ablation Studies

Effects of Map Descriptions and CoT Generation. We conduct SFT experiments with different configurations to validate the effectiveness of map descriptions and CoT generation. As shown in Table 3, adding local map descriptions improves PDMS from 79.5 to 81.9, mainly due to the gains in DAC (+1.6) and TTC (+2.8), indicating that explicit road-context information helps the model generate safer and more rule-compliant trajectories. Building upon map descriptions, CoT generation further improves PDMS to 83.0, suggesting that explicit reasoning helps the model better interpret map context and generate more goal-directed trajectories without compromising safety.

Trajectory Representation and Refinement. We also conduct ablation studies on NAVSIM and nuScenes to compare different ways of representing and refining planned trajectories. We consider four settings: (1) directly generating waypoint coordinates with the VLM; (2) adopting the action-token formulation (Zhou et al., 2026b) followed by trajectory decoding; (3) applying an MLP-based

Table 3: Ablation studies on the NAVSIM `navtest` benchmark.

Trajectory Refiner	Map Description	CoT Generation	NC ↑	DAC ↑	DD ↑	EP ↑	TTC ↑	Comf. ↑	PDMS ↑
✓	✗	✗	97.4	89.5	95.8	76.3	90.5	100.0	79.5
✓	✓	✗	96.7	91.1	92.2	75.9	93.3	100.0	81.9
✓	✓	✓	98.4	91.6	95.3	79.1	93.7	100.0	83.0

Table 4: Ablation on the CoT sample ratio.

Train Ratio (%)	Val Ratio(%)	L2 (m)
15.2	0.0	0.70
43.5	19.3	0.67
70.0	37.1	0.67

Train Ratio denotes the proportion of training samples with CoT supervision, while *Val Ratio* denotes the corresponding proportion in the validation set.

Table 5: Ablation on the trajectory refiner.

Format	PDMS	L2 (m)	Col. (%)
Action token	73.7	0.72	0.32
MLP refiner	80.1	0.67	0.14
Transformer refiner	78.4	0.69	0.18

refiner to the waypoints generated by the VLM; and (4) using a Transformer-decoder-based refiner for trajectory adjustment.

The results in Table 5 indicate that in our setting, the action-token formulation fails to reproduce the performance reported in (Zhou et al., 2026b). A possible reason is that action-token learning requires sufficient data for the model to accurately associate discrete tokens with spatial locations, whereas this mapping may remain under-learned, as our model is trained only on the NAVSIM `navtrain` split. Somewhat counterintuitively, the Transformer-based refiner also underperforms the simpler MLP refiner. We attribute this to the nature of the refinement task that rather than generating a complex sequence from scratch, the refiner only needs to predict a relatively small residual correction over an already reasonable initial trajectory. In this case, a sequence model may introduce unnecessary complexity and optimization difficulty.

Effect of CoT Supervision Ratio. Increasing the CoT supervision ratio promotes CoT generation on the validation set, but the generation rate remains substantially lower than the training ratio. Raising the ratio from 15.2% to 43.5% reduces the L2 error from 0.70 m to 0.67 m, while a further increase to 70.0% yields no additional improvement. These results suggest that moderate CoT supervision is sufficient for trajectory accuracy, whereas more supervision mainly improves the consistency of CoT generation.

Qualitative Analysis. Figure 3 presents two representative driving scenarios to qualitatively illustrate the effects of CoT reasoning and reinforcement fine-tuning. In the top row, given a left-turn command, the model without CoT reasoning incorrectly predicts a forward trajectory. In contrast, with CoT reasoning, the model incorporates the retrieved map description into its reasoning process, correctly recognizes that the ego vehicle is traveling in a left-turn lane, and consequently generates a trajectory consistent with the navigation instruction. In the bottom row, the trajectory generated by the SFT model gradually deviates from the reference trajectory at longer horizons. After reinforcement fine-tuning, the model produces a smoother trajectory that more closely follows the reference path, indicating improved long-horizon planning quality.

5 Conclusion

In this paper, we presented MindDrive, a map-informed adaptive decision framework for efficient end-to-end autonomous driving. MindDrive retrieves concise language-based map descriptions as planning-relevant spatial priors and adaptively invokes explicit reasoning according to scenario complexity. Through supervised fine-tuning and reinforcement post-training, the model learns to balance planning performance and reasoning efficiency across diverse driving scenarios. A remaining limitation is its reliance on the quality of offline map information and the accuracy of reasoning-mode selection, which we leave for future improvement.

AI USE STATEMENT

Generative AI tools were used only for literature-search assistance and minor refinement of the workflow diagram's layout. All references and final content were independently verified and reviewed by the authors, who take full responsibility for this work.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11618–11628. IEEE, 2020.

Jiajun Cao, Qizhe Zhang, Peidong Jia, Xuhui Zhao, Bo Lan, Xiaoan Zhang, Xiaobao Wei, Sixiang Chen, Liyun Li, Xianming Liu, et al. Fastdrivevla: Efficient end-to-end driving via plug-and-play reconstruction-based token pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2571–2579, 2026.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024.

Xuesong Chen, Linjiang Huang, Tao Ma, Rongyao Fang, Shaoshuai Shi, and Hongsheng Li. Solve: Synergy of language-vision and end-to-end networks for autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12068–12077. IEEE, 2025.

Haohan Chi, Huan-ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, et al. Impromptu vla: Open weights and open data for driving vision-language-action models. *Advances in Neural Information Processing Systems*, 38, 2026.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2022.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2025.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pp. 533–549. Springer, 2022.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023.

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

Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21983–21994, 2023.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8306–8316. IEEE, 2023.

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

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *International Journal of Computer Vision*, 2026.

Sicong Jiang, Zilin Huang, Kangan Qian, Ziang Luo, Tianze Zhu, Yang Zhong, Yihong Tang, Menglin Kong, Yunlong Wang, Siwen Jiao, et al. A survey on vision-language-action models for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 4583–4595, 2025b.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Derun Li, Jianwei Ren, Yue Wang, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Zhongpu Xia, Peng Jia, Xianpeng Lang, et al. Finetuning generative trajectory model with reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025.

Yun Li, Kai Katsumata, Ehsan Javanmardi, and Manabu Tsukada. Large language models for human-like autonomous driving: A survey. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 439–446. IEEE, 2024a.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-MDP: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024b.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047. IEEE, 2025.

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

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025.

Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, pp. 292–308. Springer, 2024.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2025.

Shuyao Shang, Yuntao Chen, Yuqi Wang, Yingyan Li, and ZHAO-XIANG ZHANG. Drivedpo: Policy learning via safety dpo for end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 38:81565–81585, 2026.

Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13723–13733, 2023.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15120–15130. IEEE, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Chonghao Sima, Kashyap Chitta, Zhiding Yu, Shiyi Lan, Ping Luo, Andreas Geiger, Hongyang Li, and Jose M Alvarez. Centaur: Robust end-to-end autonomous driving with test-time training. *arXiv preprint arXiv:2503.11650*, 2025.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. 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.

Qwen Team. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

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

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

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

Wenhai Wang, Jiangwei Xie, Chuanyang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, et al. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*, 2023.

Shuo Xing, Chengyuan Qian, Yuping Wang, Hongyuan Hua, Kexin Tian, Yang Zhou, and Zhengzhong Tu. Openemma: Open-source multimodal model for end-to-end autonomous driving. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 911–919. IEEE, 2025.

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

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024.

Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. Llm4drive: A survey of large language models for autonomous driving. *arXiv preprint arXiv:2311.01043*, 2023.

Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10678–10688, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. DRAMA: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024.

Zhenlong Yuan, Chengxuan Qian, Jing Tang, Rui Chen, Zijian Song, Lei Sun, Xiangxiang Chu, Yujun Cai, Dapeng Zhang, and Shuo Li. Autodrive-r^2: Incentivizing reasoning and self-reflection capacity for vla model in autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 97934–97955, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv:2503.22020*, 2025.

Xingcheng Zhou, Mingyu Liu, Ekim Yurtsever, Bare Luka Zagar, Walter Zimmer, Hu Cao, and Alois C Knoll. Vision language models in autonomous driving: A survey and outlook. *IEEE Transactions on Intelligent Vehicles*, 2025.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13782–13790, 2026a.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38: 27920–27956, 2026b.

A Appendix

A.1 Details of Offline Language-Map Library

For every city map, we set the anchor sampling interval δ_s as 3 m. Across the four nuPlan maps this yields 104,311 anchors (54,998 on lanes, 49,313 on connectors) while the four nuScenes maps yield 69,303 anchors (38,888 on lanes, 30,415 on connectors). For each anchor, we extract static topological information within a radius of $R = 35$ m. Table 6 summarizes the anchor distribution across the maps in the two datasets. The resulting library offers dense coverage of real-world driving scenarios: every ego pose in our training and evaluation scenes is associated with a valid anchor. Despite this extensive coverage, the complete offline library requires only 28.7 MB for nuPlan and 17.9 MB for nuScenes, demonstrating its low storage overhead.

A.2 Mode Label Annotation

Figure 4 illustrates the procedure used to assign difficulty scores to individual driving cases. For each case, the Qwen2.5-VL-72B model receives multimodal observations, including camera images, historical trajectory and driving command and first generates a structured chain-of-thought description. This description consists of three components: *Scene*, which summarizes the relevant traffic and environmental conditions; *Reasoning*, which explains how these conditions affect the ego vehicle's planning problem; and *Best Driving Action*, which describes the appropriate driving behavior.

Based on this intermediate description, the VLM is asked to rate the case along three dimensions: traffic participant density, environmental complexity, and decision complexity. Each dimension is evaluated at three ordinal levels, corresponding to scores of 1, 3, and 5, where a higher score indicates a more challenging condition. The score weighting parameters λ, β, and γ are set as 0.4, 0.4 and 0.2 respectively. This example is categorized as a hard case because it contains dense traffic, a relatively complex environment, and a driving command to decelerate and stop safely. The detailed distribution of `<easy_mode>` samples and `<hard_mode>` samples is shown in Table 7.

Table 6: Anchor distribution statistics for the nuPlan and nuScenes datasets.

Dataset	Map	Lane	Connector	Total	Intersect	Traffic Sign
nuPlan	sg-one-north	11,004	9,329	20,333	33.9%	29.7%
	us-ma-boston	13,741	12,146	25,887	38.0%	44.5%
	us-nv-las-vegas-strip	23,125	24,516	47,641	47.4%	29.8%
	us-pa-pittsburgh-hazelwood	7,128	3,322	10,450	27.5%	39.9%
	Total	**54,998**	**49,313**	**104,311**	**40.4%**	**34.5%**
nuScenes	boston-seaport	13,044	11,297	24,341	37.0%	44.9%
	singapore-hollandvillage	7,669	4,211	11,880	27.1%	24.2%
	singapore-onenorth	9,945	7,531	17,476	31.0%	30.3%
	singapore-queenstown	8,230	7,376	15,606	34.7%	29.6%
	Total	**38,888**	**30,415**	**69,303**	**33.3%**	**34.2%**

Intersect indicates that the anchor is located within an intersection. *Traffic Sign* indicates that the anchor description contains at least one traffic control sign.

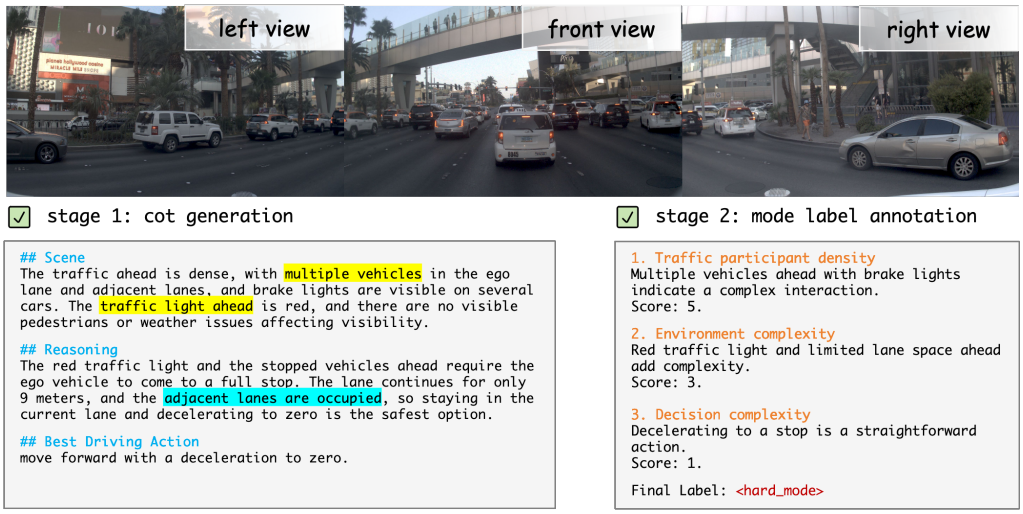


Figure 4: Pipeline for scenario complexity scoring. Building on the generated chain of thought, each scenario is scored along three dimensions: traffic participant density, environmental complexity, and decision complexity.

A.3 Failure Cases Analysis

This analysis is based on the evaluation results of the `navtest` dataset. Among all evaluated samples, we select 548 bad samples with a PDM score below 0.5, corresponding to 4.5% of the validation dataset. We summarize the four typical failure case patterns among the 548 low-scoring samples: 1) *ego vehicle makes no effective progress*; 2) *ego vehicle leaves the drivable area*; 3) *ego vehicle violates the intended driving direction*; 4) *ego vehicle collides with another vehicle or static object*. Representative examples of these four failure patterns are shown in Figure 5.

We further analyze the failure rate according to the instruction type. Turning scenarios are substantially more challenging than forward-driving scenarios. For all samples, the failure rates for left-turn and right-turn instructions are 9.8% and 7.2%, respectively, compared with only 2.3% for forward-driving and other instructions. Among the 548 low-scoring samples, 66% correspond to turning instructions.

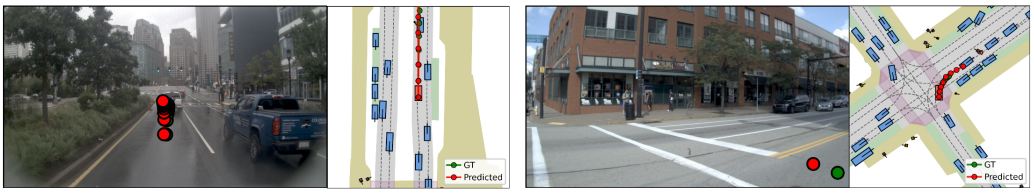
These statistics indicate that the dominant failure modes are related to drivability and progress, rather than collisions alone. They also suggest that turning scenarios constitute the primary source

Table 7: Sample statistics of the training datasets.

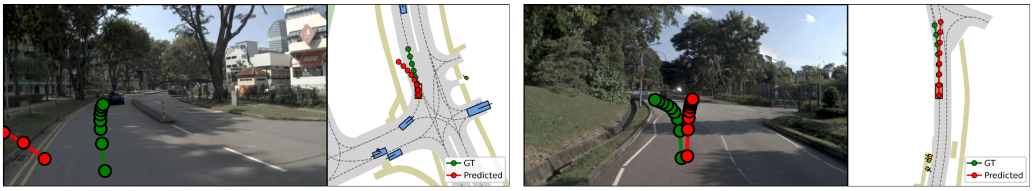
Dataset	Total	<easy_mode>	<hard_mode>
nuPlan (navtrain)	103,288	58,368 (56.5%)	44,920 (43.5%)
nuScenes (train)	19,030	11,348 (59.6%)	7,682 (40.4%)



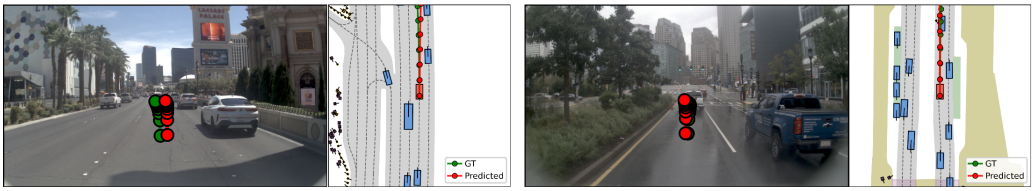
(a) ego vehicle makes no effective progress.



(b) ego vehicle leaves the drivable area.



(c) ego vehicle drives against the right direction.



(d) ego vehicle collides with another vehicle or static object.

Figure 5: Representative failure cases on the navtest validation set. From top to bottom, the rows show examples of ineffective progress, leaving the drivable area, driving in the wrong direction, and collisions with other vehicles or static objects, with two examples for each failure pattern.

of planning failures, likely because they require the model to jointly reason about road geometry, directional constraints, surrounding traffic, and the desired maneuver.