# SPANVLA: LEARNING FROM NEGATIVE-RECOVERY SAMPLES WITH FAST ACTION BRIDGING FOR VISION-LANGUAGE-ACTION MODEL

**Anonymous authors**
Paper under double-blind review

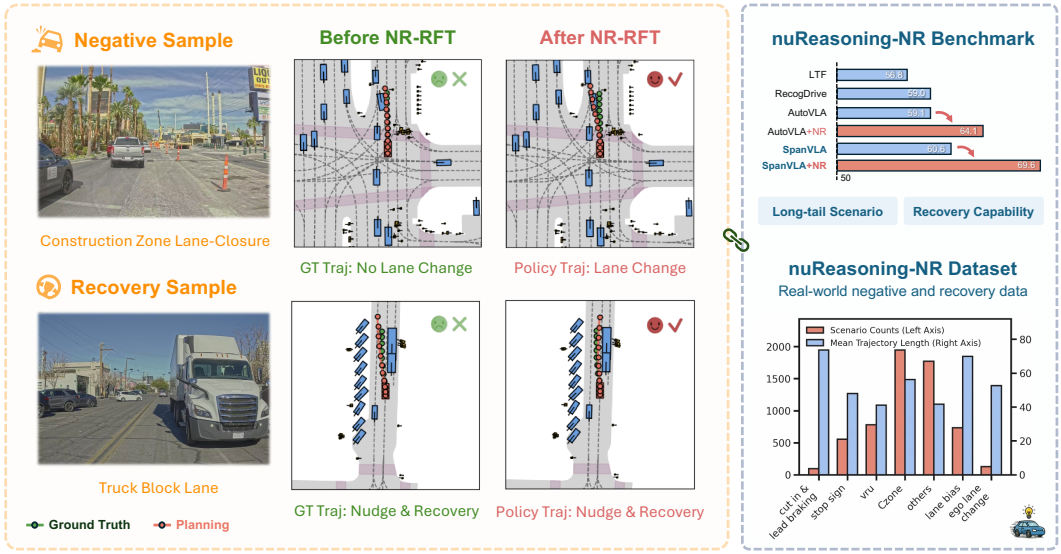



Figure 1: **Learning what to avoid and how to recover.** With real-world negative and recovery supervision, SPANVLA predicts an earlier merge before a lane closure and a return to the original lane after passing an obstruction (left). We introduce nuReasoning-NR, the first real-world driving dataset and benchmark for learning from negative and recovery samples.

## ABSTRACT

Vision-Language-Action (VLA) models bring rich world knowledge and reasoning capabilities to autonomous driving. However, most driving VLAs are trained only on positive expert demonstrations, which teach models what to do but provide limited supervision on which behaviors to avoid and how to recover from these failures, leading to limited robustness. We introduce **SPANVLA**, a robust and efficient driving VLA, together with **nuReasoning-NR**, the first real-world driving dataset and benchmark for learning from negative and recovery samples. An extension of nuReasoning, it comprises 36K reasoning-intensive driving scenarios, including 3K suboptimal trajectories and 3K expert recovery trajectories. To exploit their asymmetric supervision, we introduce **Negative–Recovery Reinforcement Fine-Tuning (NR-RFT)**, which penalizes undesirable behaviors and rewards expert corrective actions. Because challenging scenarios produce rollout groups dominated by similar suboptimal or even infeasible trajectories, it further incorporates an advantage correction to preserve informative optimization signals and encourage exploration. For efficient action generation, SPANVLA retains autoregressive reasoning and introduces a lightweight flow-matching action bridge conditioned on sparse-layer KV caches and initialized from the historical trajectory. Extensive experiments on NAVSIM v1/v2, nuReasoning, and the nuReasoning-NR benchmark demonstrate strong and robust planning performance and efficient action generation. Qualitative results across diverse challenging scenarios further highlight its planning robustness and recovery capability. The dataset and benchmark code will be released to support future research.



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.

# 1 INTRODUCTION

End-to-end autonomous driving systems directly map raw sensor observations to actions within a unified framework (Jiang et al., 2023; Zhou et al., 2024; 2025c; Lei et al., 2025; Xu et al., 2025). More recently, Vision-Language-Action (VLA) models have introduced the world knowledge and reasoning capabilities of Vision-Language Models (VLMs) into end-to-end driving (Ma et al., 2025; Zhao et al., 2026). Through unified perception, reasoning, and action generation, VLA models provide a promising foundation for generalizable driving, particularly in long-tail scenarios.

Despite this progress, most existing VLAs **only learn from positive demonstrations, leading to limited robustness** (Yu et al., 2025). However, real-world negative and recovery data, which capture negative behaviors that should be avoided, as well as recovery behavior from challenging scenarios, are often overlooked in datasets and models (Wang et al., 2024b; Xing et al., 2026). Such negative-recovery data can provide targeted signals for efficient gains in model performance and robustness on long-tail scenarios, rather than relying on positive data concentrated in nominal driving, but the lack of real-world negative and recovery data limits progress in this domain (Liu et al., 2025a; Gao et al., 2026). Besides, how to fully exploit those data is still an open question. First, they encode asymmetric supervision across positive, negative, and recovery behavior. Second, the challenging scenarios of negative and recovery samples often produce rollout groups dominated by similar sub-optimal or infeasible trajectories, leading to uninformative advantages and unstable training.

Current VLA models also face **high action generation latency and large action experts**. Directly representing trajectories as discrete tokens provides a simple interface for jointly learning reasoning and planning (Kim et al., 2024; Zhou et al., 2025b), but requires the large VLM backbone to decode every action autoregressively. The resulting latency grows linearly with the trajectory length, making high-frequency planning expensive. Continuous action experts provide a more efficient alternative (Li et al., 2025c; Wang et al., 2025), yet existing designs commonly rely either on the final-layer VLM feature, which may omit intermediate visual and semantic information, or on dense KV caches from all VLM layers and a large action expert, introducing substantial memory and latency.

To this end, we propose **SPANVLA**, a robust and efficient VLA framework that learns from real-world negative and recovery samples with fast action bridging, as illustrated in Figure 2. First, we introduce **nuReasoning-NR**, a public extension of nuReasoning (Huang et al., 2026) containing 36K reasoning-intensive scenarios, including 3K negative trajectories and 3K expert recovery trajectories. It is the first large-scale real-world driving dataset and benchmark with negative and recovery samples. Second, we introduce **Negative–Recovery Reinforcement Fine-Tuning (NR-RFT)** to exploit the asymmetric supervision provided by negative and recovery samples. Specifically, type-specific negative penalties suppress representative negative behaviors, while recovery rewards reinforce expert corrective actions. We additionally introduce an advantage correction to preserve informative optimization signals and encourage exploration toward feasible solutions. Third, SPANVLA retains autoregressive reasoning and introduces an **Efficient Action Bridge** that aggregates multi-granular sparse KV caches to enable fast action generation. Instead of learning the flow directly from random noise (Li et al., 2025c), it conditions on the historical trajectory and learns the transformation from past actions to future actions, improving generation quality and efficiency.

Extensive evaluation on NAVSIM (v1 & v2) (Dauner et al., 2024; Cao et al., 2025) and the nuReasoning (Huang et al., 2026) benchmark demonstrates the state-of-the-art performance of our model. Empirical results validate that learning with real-world negative and recovery samples further improves planning performance and robustness. The main contributions are as follows:

1. We propose **SPANVLA**, a robust and efficient VLA framework that learns from negative-recovery samples with fast action bridging. It leverages a sparse KV cache with historical-trajectory-initialized flow matching to reduce action-generation latency.
2. We introduce **nuReasoning-NR**, the first real-world driving dataset and benchmark containing both negative and expert recovery samples. It comprises 36K samples and enables systematic study of learning from real-world negative and recovery experiences.
3. We introduce **NR-RFT**, a post-training method that exploits asymmetric negative-recovery supervision, together with an advantage correction for difficult rollout groups.
4. SPANVLA achieves state-of-the-art performance across the NAVSIM v1, v2, and nuReasoning benchmarks, and learning with negative and recovery samples improves robustness.

## 2 Related Work

**VLA Models for Autonomous Driving.** The reasoning capabilities and world knowledge of VLMs have motivated their application in autonomous driving (Hwang et al., 2024; Cai et al., 2024; Liu et al., 2026; Zhang et al., 2026) and robotics (Kim et al., 2016; Intelligence et al., 2025; Liu et al., 2025b). Early approaches employ VLMs to provide high-level meta-actions (Xie et al., 2026; Jiang et al., 2024; Tian et al., 2024) or auxiliary supervision (Pan et al., 2024; Xu et al., 2024) for conventional planning systems, limiting end-to-end optimization. Recent VLA models instead unify reasoning and planning by representing trajectories as text (Rowe et al., 2025; Zhou et al., 2025a; Xing et al., 2025; Mao et al., 2023) and discrete action or world tokens (Zhou et al., 2025b; Zeng et al., 2025), but autoregressive action decoding incurs latency that grows with trajectory length. Thus, SpanVLA generates continuous actions efficiently through a lightweight action expert and further fine-tunes the model with real-world negative-recovery samples to improve robustness.

**Driving Reasoning and Negative–Recovery Datasets.** Large-scale real-world driving datasets provide multimodal observations and expert trajectories (Karnchanachari et al., 2024; Contributors, 2023; Xu et al., 2025). Recent datasets further introduce driving question answering and reasoning annotations (Sima et al., 2024; Wang et al., 2024c; Park et al., 2025; Arai et al., 2025; Huang et al., 2026), but remain centered on positive driving data. Takeover-based approaches collect negative or corrective behaviors (Liu et al., 2025a; Fang et al., 2025), yet primarily rely on simulation, and real-world negative-recovery data remains a gap. In contrast, **nuReasoning-NR** extends nuReasoning with real-world negative trajectories and corresponding expert recovery trajectories, providing the first large-scale driving negative and recovery dataset.

**Action Bridging for VLA Models.** Action experts bridge VLM representations with continuous control while avoiding autoregressive trajectory decoding (Wang et al., 2025; Li et al., 2025a; Dang et al., 2026; Li et al., 2025c). ReCogDrive (Li et al., 2025c) conditions a diffusion planner on the final-layer VLM representation, whereas Alpamayo (Wang et al., 2025) uses dense KV caches from all VLM layers with a flow-matching planner. The former may omit intermediate visual and semantic information, while the latter introduces substantial memory and computational overhead. In contrast, SpanVLA aggregates multi-granular KV caches from sparse VLM layers and employs a lightweight flow-matching expert initialized from the historical trajectory rather than pure noise.

## 3 nuReasoning-NR Dataset and Benchmark

We introduce **nuReasoning-NR**, the first real-world driving dataset and benchmark for learning from both negative and recovery samples. It extends nuReasoning (Huang et al., 2026) and contains 36K samples: 30K positive reasoning samples, 3K negative samples, and 3K recovery samples. All three subsets cover the same interaction-intensive and scenario categories, as shown in Figure 1.

**Real-World Negative and Recovery Data.** We collect negative and recovery samples from early-stage exploratory driving tests. A negative segment records suboptimal ego behavior, while its recovery counterpart records the subsequent expert correction. Each segment contains its own observation history and reference trajectory as a data sample.

These samples provide distinct forms of supervision. A negative trajectory identifies behavior that should be suppressed, including inefficient or hesitant maneuvers as well as potentially unsafe actions. An expert recovery trajectory demonstrates how to correct behavior from a difficult driving state. Their event-level correspondence captures both the suboptimal behavior and the subsequent correction, while their different reference semantics motivate the asymmetric supervision in NR-RFT. Representative examples are shown in Figure 1.

**Positive Reasoning Data.** The positive subset provides expert trajectories and reasoning annotations for supervised fine-tuning. We generate reasoning annotations using an automated pipeline based on Gemini-3-Pro (Gemini Team, Google DeepMind, 2025). The pipeline identifies decision-relevant scene elements, analyzes their states, selects longitudinal and lateral actions, and summarizes the resulting decisions into concise reasoning traces. Automated checks assess annotation-format completeness and alignment with the reference actions, followed by a human quality audit. Further annotation details are provided in Appendix B.

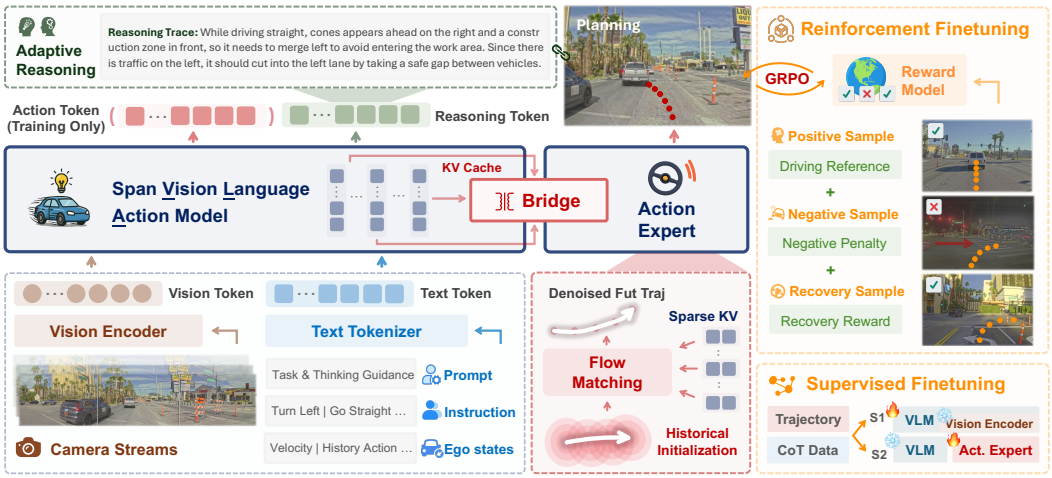



Figure 2: **Overview of SPANVLA.** Supervised fine-tuning aligns reasoning and planning, while NR-RFT uses real-world negative and recovery samples to penalize undesirable behaviors and reward expert corrections. NR-RFT updates the policy using corrected group-relative advantages. For efficient inference, a lightweight action bridge generates continuous trajectories through flow matching conditioned on sparse-layer KV caches and initialized from the historical trajectory.

**nuReasoning-NR Benchmark.** Within the 36K samples, we designate 300 negative scenarios and 300 recovery scenarios for benchmark evaluation. The negative subset evaluates planning in situations associated with suboptimal recorded behavior, while the recovery subset evaluates corrective planning from difficult states. To support the evaluation, we adapt the NAVSIM evaluation pipeline (Dauner et al., 2024) using the dataset's object annotations and HD maps. This benchmark enables systematic evaluation of planning performance across both scenario types and of the benefits of negative-recovery supervision. Additional dataset details are provided in Appendix B.

## 4 SPANVLA

SPANVLA combines a VLM backbone for reasoning with a lightweight action bridge for continuous trajectory generation, as illustrated in Figure 2. Training uses supervised fine-tuning (SFT) and Negative–Recovery Reinforcement Fine-Tuning (NR-RFT).

### 4.1 VLM BACKBONE

**Model Inputs.** The backbone takes a multimodal query $q = (\mathcal{V}^t, \mathcal{T}^t)$ at driving time $t$. The visual input $\mathcal{V}^t$ contains four frames per camera at 2 Hz. The textual input $\mathcal{T}^t$ includes a task prompt, a driving instruction, and ego states comprising acceleration, velocities, and the historical trajectory.

**Reasoning with Autoregressive Decoding.** During training, to enable the model to learn how to reason for planning, we introduce an additional discrete action generation task following reasoning in the VLM, which unifies reasoning and planning within the SFT, as following:

$$[\mathcal{T}_{\text{Reason}}, (A_{\text{token}})] = \text{VLM}(\mathcal{V}^t, \mathcal{T}^t); A_{\text{token}} = \{a^\tau_{\text{token}}\}^{t+T_f}_{\tau=t}, \tag{1}$$

where we leverage the action codebook of AutoVLA (Zhou et al., 2025b) to discretize the trajectories into action tokens $A_{\text{token}}$. We further adopt adaptive reasoning: slow thinking generates reasoning before actions, while fast thinking directly generates actions. At inference, the VLM emits an action-start token and delegates continuous trajectory generation to the action bridge.

### 4.2 EFFICIENT ACTION BRIDGE

**Sparse KV-Cache Conditioning.** The action bridge uses a lightweight Transformer $f_\phi$ conditioned on KV caches from selected VLM layers, denoted by $\mathbf{c}_{\text{vlm}}$. These caches encode visual

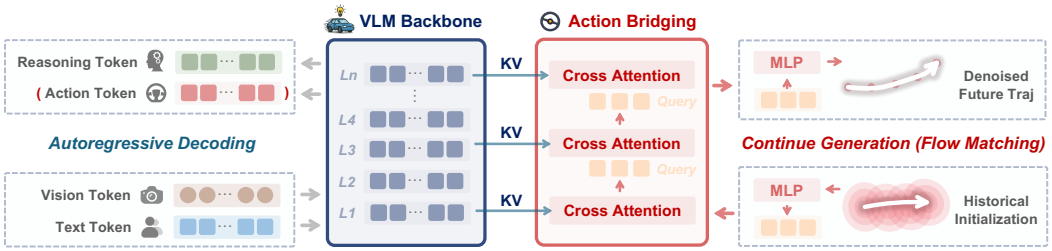



Figure 3: Overview of the efficient action bridging of the SPANVLA model. The VLM backbone leverages the autoregressive decoding to generate the reasoning results, and we introduce an action bridging to utilize the sparse KV cache to efficiently generate the continuous trajectory with historical initialization based on flow-matching, avoiding the linearly increasing latency of the autoregressive decoding with the long action length.

observations, textual inputs, and the generated reasoning. Accessing representations across selected depths provides multi-granular context while reducing the computation associated with dense layer-wise conditioning. The bridge retains the VLM's number of attention heads with a smaller hidden dimension. At each flow step, it attends to the selected caches to predict a vector field $\mathbf{v}_\tau$ from the current trajectory state and a flow-time embedding, as illustrated in Figure 3.

**Historical-Trajectory Initialization.** Rather than starting from pure Gaussian noise $\mathcal{N}(0, \mathbf{I})$ (Wang et al., 2025), our method initializes from the historical trajectory embedding $\mathbf{a}_{\text{his}}$ with MLP layers and directly learns the transition from the historical action space to the future action space: $\mathbf{a}_{\text{his}} \xrightarrow{\mathbf{v}_\tau} \mathbf{a}$, improving performance and sampling efficiency. Moreover, to enhance robustness, we inject some Gaussian noise into the historical trajectory embeddings during training. In practice, we employ the optimal transport displacement map (Lipman et al., 2022). For training, we sample $\mathbf{a}_\tau = \tau\mathbf{a} + (1-\tau)\mathbf{a}_{\text{his}}$. In inference, the flow matching denoising with in each $\Delta\tau$ step is:

$$\mathbf{a}_{\tau+\Delta\tau} = \mathbf{a}_\tau + \Delta\tau\, f_\phi(\mathbf{a}_\tau, \tau, \mathbf{c}_{\text{vlm}}). \tag{2}$$

### 4.3 SUPERVISED FINE-TUNING

**VLM Backbone.** We supervise the VLM with expert trajectories and reasoning annotations. Slow-thinking targets contain reasoning followed by action tokens, while fast-thinking targets contain only action tokens. All training data are curated with ground-truth assistant responses to ensure high-quality supervision for the VLM. Thus, the loss function for the output sequence $o = [\mathcal{T}_{\text{Reason}}, A_{\text{token}}]$ is defined as:

$$\mathcal{L}_{\text{LM}} = -\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(o_i \mid o_{<i}, \mathcal{V}^t, \mathcal{T}^t), \mathcal{L}_{\text{Action}} = -\frac{1}{T}\sum_{i=L+1}^{L+T} \log p_\theta(o_i \mid o_{<i}, \mathcal{V}^t, \mathcal{T}^t), \tag{3}$$

where $N = L + T$, and $p_\theta$ denotes the predicted distribution with policy $\theta$. The final loss is $\mathcal{L}_i^{\text{SFT}} = w_i \cdot \mathcal{L}_{\text{LM},i} + \mathcal{L}_{\text{Action},}$, where $w_i = 0$, and $w_i = 1$ when CoT is activated.

**Action Bridging.** It is trained with a conditional flow matching loss, which encourages the model to learn the optimal transport path from the historical action to the future planning action :

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\tau\in[0,1], \mathbf{c}_{\text{vlm}}\in[\mathcal{V}^t, \mathcal{T}^t, \mathcal{T}_{\text{Reason}}]} \| f_\theta(\mathbf{a}_\tau, \tau, \mathbf{c}_{\text{vlm}}) - \mathbf{v}_\tau(\mathbf{a}_\tau) \|^2, \tag{4}$$

where $\tau$ is sampled by shifted normal distribution during training, and the target vector field is $\mathbf{v}_\tau(\mathbf{a}_\tau) = \frac{d\mathbf{a}_\tau}{d\tau} = \mathbf{a} - \mathbf{a}_{\text{his}}$. Moreover, following Alpamayo (Wang et al., 2025), we stop gradients for both the VLM backbone and the bridging module to preserve their learned knowledge. In practice, we first train the VLM backbone and then fine-tune the action bridge separately.

### 4.4 NEGATIVE–RECOVERY REINFORCEMENT FINE-TUNING

NR-RFT adapts group relative policy optimization (GRPO) (Shao et al., 2024; Jiang et al., 2025b) to the asymmetric supervision of negative and recovery samples, with an advantage correction for

difficult rollout groups to encourage exploration. For each query $q$, we sample $G$ VLM responses and decode their action tokens into trajectories $o_i = \{\mathbf{p}_{i,t}\}_{t=0}^{T}$. Rewards evaluate each trajectory and its associated reasoning, and update the VLM token policy.

**Asymmetric Behavioral Rewards.** Given the reference trajectory $o^{\text{ref}} = \{\mathbf{p}_t^{\text{ref}}\}_{t=0}^{T}$, we define a bounded reference-matching term:

$$d(o_i, o^{\text{ref}}) = \frac{1}{T+1}\sum_{t=0}^{T} \|\mathbf{p}_{i,t} - \mathbf{p}_t^{\text{ref}}\|_2, \qquad r_{\text{Ref}}(q, o_i) = \text{clip}\left(1 - \frac{d(o_i, o^{\text{ref}})}{\delta}, 0, 1\right), \tag{5}$$

where $\delta > 0$ is a proximity threshold. We set $r_{\text{Negative},i} = r_{\text{Ref}}(q, o_i)$ for negative samples and $r_{\text{Recover},i} = r_{\text{Ref}}(q, o_i)$ for recovery samples; each term is zero for other types. The final reward is

$$r_i = r_{\text{Driving},i} - \lambda_{\text{N}} r_{\text{Negative},i} + \lambda_{\text{R}} r_{\text{Recover},i} - \lambda_{\text{C}} r_{\text{CoT},i}, \tag{6}$$

where the non-negative weights control the corresponding terms. This design penalizes proximity to undesirable behavior and rewards agreement with expert corrections. The reference-matching term vanishes beyond $\delta$, so further deviation from a negative reference provides no additional benefit.

The driving reward uses PDMS or EPDMS according to the evaluation setting (Dauner et al., 2024; Cao et al., 2025). For negative samples, we omit ego progress because the recorded trajectory is an undesirable reference. The reasoning cost $r_{\text{CoT},i}$ penalizes unnecessarily long reasoning and is replaced by a fixed penalty when a rule-based check detects disagreement between reasoning and the predicted maneuver. Detailed reward definitions are provided in the supplementary material.

**Advantage Correction.** Challenging negative and recovery scenarios can produce rollout groups with nearly identical rewards or all-zero driving rewards. Standard GRPO uses $A_i^{\text{GRPO}} = (r_i - \mu)/(\sigma + \varepsilon_{\text{num}})$, where $\mu$ and $\sigma$ are the group reward mean and standard deviation. Standard-deviation normalization can amplify small reward differences, while groups with identical final rewards yield zero advantages.

We remove standard-deviation normalization to preserve the original reward differences. For groups with all-zero driving rewards, we additionally include a virtual reward of 1 in the calculation. Let $F = \mathbb{I}[r_{\text{Driving},1} = \cdots = r_{\text{Driving},G} = 0]$, while $r_i$ remains the final reward defined in equation 6. Then, the corrected advantage is

$$\widetilde{\mu} = \frac{\sum_{j=1}^{G} r_j + F}{G + F}, \qquad A_i = r_i - \widetilde{\mu}. \tag{7}$$

When $F = 1$ and the final rewards are also all zero, every sampled output receives $A_i = -1/(G+1)$, providing a penalty signal on the sampled zero-reward outputs. The virtual reward affects only the baseline and contributes no additional output to the policy objective. When $F = 0$, $A_i = r_i - \mu$.

# 5 Experiments

## 5.1 Experimental Setup

**Dataset.** Training data are drawn from the NAVSIM `navtrain` split (100K scenarios) (Dauner et al., 2024), nuReasoning, and the training portion of nuReasoning-NR. Expert trajectories and reasoning annotations provide supervised training signals, while negative and recovery samples introduce supervision on undesirable behaviors and expert corrections during NR-RFT.

**Benchmark.** We evaluate on NAVSIM v1 `navtest` using PDMS and NAVSIM v2 `navtest` and `navhard` using EPDMS (Dauner et al., 2024; Cao et al., 2025). The `navhard` split further assesses planning in challenging scenarios through pseudo closed-loop simulation. On nuReasoning (Huang et al., 2026), we report NPS and ADE to evaluate planning in interaction-intensive scenarios. On nuReasoning-NR, we report PDMS separately for the 300 negative and 300 recovery scenarios, together with their average and component metrics.

Table 1: Comparison on the **nuReasoning-NR benchmark**. Results are reported in PDMS.

| Methods | Negative ↑ | Recovery ↑ | Average PDMS ↑ | NC ↑ | DAC ↑ | EP ↑ | TTC ↑ | Comf. ↑ |
|---|---|---|---|---|---|---|---|---|
| *Models Trained with Positive Data Only* | | | | | | | | |
| LTF (Chitta et al., 2022) | 62.7 | 51.0 | 56.8 | 99.2 | 59.9 | 55.7 | 99.8 | 89.4 |
| ReCogDrive (Li et al., 2025c) | 64.0 | 54.0 | 59.0 | 99.6 | 61.8 | 58.0 | 99.7 | 90.9 |
| AutoVLA (Zhou et al., 2025b) | 64.2 | 54.0 | 59.1 | 99.4 | 62.2 | 58.0 | 99.7 | 90.4 |
| SPANVLA | 65.1 | 56.1 | 60.6 | 99.9 | 62.3 | 58.6 | 100.0 | 92.5 |
| *Models Trained with Additional NR Data* | | | | | | | | |
| AutoVLA + NR | 68.8 | 59.5 | 64.1 | 99.7 | 67.3 | 62.5 | 99.8 | 90.5 |
| SPANVLA + NR | **74.1** | **65.1** | **69.6** | **100.0** | **71.2** | **66.9** | **100.0** | **93.2** |

Table 2: Comparison on the **nuReasoning benchmark**. Results are reported using planning Performance.

| Methods | NC ↑ | DA ↑ | EP ↑ | CF ↑ | HL ↑ | NPS ↑ | ADE ↓ |
|---|---|---|---|---|---|---|---|
| UniAD (Hu et al., 2023) | 88.9 | 87.6 | 89.6 | 92.6 | 48.8 | 55.7 | 2.1 |
| DiffusionDrive (Liao et al., 2024) | 90.2 | 88.3 | 90.5 | 95.0 | 52.0 | 57.9 | 1.9 |
| AutoVLA (Zhou et al., 2025b) | 90.9 | 86.5 | 89.3 | **99.9** | 49.9 | 59.1 | 2.1 |
| Alpamayo-1.5 (Wang et al., 2025) | 90.3 | 86.1 | 86.5 | 97.9 | 33.8 | 50.5 | 2.9 |
| nuVLA (Huang et al., 2026) | 96.3 | 96.3 | **90.7** | 99.7 | 62.2 | 73.1 | 1.6 |
| SPANVLA (Post-RFT) | **96.5** | **96.4** | 90.3 | **99.9** | **63.4** | **73.3** | **1.4** |

**Implementation Details.** We use the Qwen2.5-VL-3B model (Bai et al., 2025) as the VLM backbone. For SFT, we employ Fully Sharded Data Parallel (FSDP) training on 8 NVIDIA H200 GPUs. The per-GPU batch size is set to 4, with a gradient accumulation step of 4. For the action bridge, the batch size is set to 16, and we selected the feature from the VLM with an interval of two. For NR-RFT, we adopt LoRA adapters (Hu et al., 2022) for efficient fine-tuning. The learning rate for NR-RFT is set to $3 \times 10^{-5}$ and the group sample size is 64. We perform a single policy update per step, resulting in a simplified objective that eliminates the need for clipping or maintaining an old policy. Additional implementation details are provided in the Appendix.

### 5.2 MAIN RESULTS

This section reports the main results of SPANVLA, and more results are provided in the Appendix.

**nuReasoning-NR Benchmark.** Table 1 evaluates whether real-world negative and recovery supervision improves planning in challenging scenarios. Incorporating NR supervision increases the average PDMS of SPANVLA by 14.9%. The gains occur on both subsets: negative scenarios improve by 13.8%, and recovery scenarios improve by 16%. The benefit extends to AutoVLA, whose average PDMS increases from 59.1 to 64.1 with NR training. This supports the value of negative-recovery supervision beyond the SPANVLA architecture.

**NAVSIM Benchmark.** Table 3 compares SPANVLA with state-of-the-art (SOTA) conventional end-to-end and VLA-based methods on the `navtest` benchmarks of NAVSIM v1 and NAVSIM v2, as well as the NAVSIM v2 `navhard` benchmark. RFT substantially improves the one-shot policy, while incorporating negative and recovery data provides further gains over positive-only RFT on challenging `navhard`. SPANVLA achieves the best overall performance across all three benchmarks. Detailed metric breakdowns are provided in Appendix Tables S1 to S3.

**nuReasoning Benchmark.** On nuReasoning, SPANVLA achieves the highest NPS and lowest ADE among the methods in Table 2. These results complement the negative-recovery evaluation by demonstrating strong planning performance across interaction-intensive driving scenarios.

**Efficient Action Bridging.** Table 4 compares various action policies. For 50-waypoint trajectories, the bridge reduces action-generation latency, achieving a $21.5\times$ speedup. The reported total latency has a reduction of approximately 74%. Increasing the output from 10 to 50 waypoints leaves the bridge's measured generation time approximately unchanged, supporting efficient trajectory generation at high temporal resolution. The $L_1$ head is faster at 0.02 s, but achieves 85.1 PDMS compared with 90.8 for flow matching in the reported 10-waypoint evaluation. The proposed bridge thus provides a favorable balance between planning quality and action-generation efficiency.

Table 3: Comparison with SOTA methods on the `navtest` benchmarks of **NAVSIM v1** and **NAVSIM v2**, and on the `navhard` benchmark of **NAVSIM v2**. NAVSIM v1 is evaluated using PDMS, while NAVSIM v2 is evaluated using EPDMS. * denotes best-of-N selection with N = 6, where the candidate with the highest PDMS is chosen. Detailed metric breakdowns are provided in Appendix Tables S1 to S3.

| Methods | Cam. | Lid. | **NAVSIM v1** PDMS ↑ | **NAVSIM v2** EPDMS ↑ |
|---|---|---|---|---|
| *Conventional End-to-end-based Methods* | | | | |
| TransFuser (Chitta et al., 2022) | ✓ | ✓ | 84.0 | 76.7 |
| Hydra-MDP (Li et al., 2024) | ✓ | ✓ | 86.5 | 81.4 |
| DiffusionDrive (Liao et al., 2024) | ✓ | ✓ | 88.1 | 84.5 |
| *VLA-Based Methods* | | | | |
| AutoVLA (Zhou et al., 2025b) | ✓ | – | 89.1 | 82.4 |
| ReCogDrive (Li et al., 2025c) | ✓ | – | 89.6 | 83.6 |
| DriveVLA-W0 (Li et al., 2025a) | ✓ | – | 90.2 | 86.1 |
| Qwen-Drive (Zhou et al., 2026) | ✓ | – | 90.7 | — |
| ***Ours*** | | | | |
| SPANVLA (One-shot) | ✓ | – | 82.1 | 79.4 |
| SPANVLA (Post-RFT, positive only) | ✓ | – | 90.8 | 86.9 |
| SPANVLA* | ✓ | – | **93.3** | **90.5** |

| Methods | **v2 navhard** EPDMS ↑ |
|---|---|
| *Conventional End-to-end-based Methods* | |
| TransFuser (Chitta et al., 2022) | 23.1 |
| DiffusionDrive (Liao et al., 2024) | 24.2 |
| Senna-E2E (Jiang et al., 2026) | 27.2 |
| MindDrive (Sun et al., 2025) | 30.9 |
| World4Drive (Zheng et al., 2025) | 34.9 |
| Navformer (Wang et al., 2024a) | 34.1 |
| iPad (Guo et al., 2026) | 34.7 |
| RAP-DINO (Feng et al., 2025a) | 39.6 |
| ***Ours*** | |
| SPANVLA (Post-RFT, positive only) | 40.1 |
| SPANVLA (Post-RFT, with NR) | 40.8 |
| SPANVLA* | **47.9** |

Table 4: Comparison for action policy on the **NAVSIM v1**. The fixed VLM encoding + prefilling time is with the same inputs, and reasoning tokens are the average tokens across 10K samples. FM: flow-matching.

| Methods | Action | PDMS ↑ | VLM Encoding + Prefilling (s) ↓ | Reasoning Generation (s)↓ | Trajectory Generation (s)↓ | Total (s) ↓ |
|---|---|---|---|---|---|---|
| *Autoregressive Decoding* | | | | | | |
| AutoVLA (Zhou et al., 2025b) | 10 | 89.1 | 0.09 | 0.76 (23 tokens) | 0.40 (12 tokens) | 1.25 |
| | 50 | – | 0.09 | 0.76 (23 tokens) | 1.72 (52 tokens) | 2.57 |
| *Action Expert* | | | | | | |
| SPANVLA ($L_1$ Head) | 10 | 85.1 | 0.09 | 0.50 (15 tokens) | 0.02 | 0.61 |
| | 50 | – | 0.09 | 0.50 (15 tokens) | 0.02 | 0.61 |
| SPANVLA (FM) | 10 | **90.8** | 0.09 | 0.50 (15 tokens) | 0.08 (5 steps) | 0.67 -46% |
| | 50 | – | 0.09 | 0.50 (15 tokens) | 0.08 (5 steps) | 0.67 -74% |

### 5.3 ABLATION STUDIES FOR NR-RFT

**Negative and Recovery Supervision.** Figure 4 (left) compares training with positive samples, positive and negative samples, and all three sample types. Adding negative supervision improves PDMS on nuReasoning-NR from 60.6 to 64.2, while incorporating recovery supervision further increases it to 69.6. The additional gain from recovery samples supports their value beyond negative supervision alone. Together, these results support learning both which behaviors to avoid and how to correct behavior from difficult driving states.

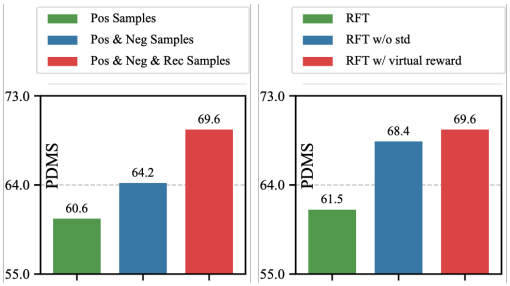



Figure 4: Ablations of NR-RFT on nuReasoning-NR. **Left:** Positive-only, positive–negative, and positive–negative–recovery supervision. **Right:** Standard GRPO, mean-centered advantages without standard-deviation normalization, and the full estimator with virtual-reference correction.

**Advantage Correction.** Figure 4 (right) evaluates the two components of advantage correction. Removing standard-deviation normalization improves PDMS from 61.5 to 68.4. Adding the virtual reference for groups with all-zero final rewards further increases PDMS to 69.6. These results support their effectiveness.

### 5.4 ABLATION STUDIES FOR ACTION BRIDGING

**Effect of Historical Initialization.** Table 5 isolates historical initialization under the same interval-2 cache configuration. Historical initialization improves PDMS from 86.4 to 90.8 while

Table 5: Comparison for SPANVLA with different bridging layers on **NAVSIM v1**. Full Caching: Following Alpamayo (Wang et al., 2025), an empty VLM decoding module of the backbone for bridging. "Last Layer," "Interval 4," and "Interval 2" denote extracting features from the final layer, every four layers, and every two layers. HI: historical initialization.

| Metrics | Full Caching | Last Layer | Interval 4 | Interval 2 w/o HI | **Interval 2** |
|---|---|---|---|---|---|
| PDMS ↑ | 88.1 | 79.3 | 82.2 | 86.4 | **90.8** |
| Traj. Gen. (s) ↓ | 0.18 | **0.01** | 0.05 | 0.08 | 0.08 |

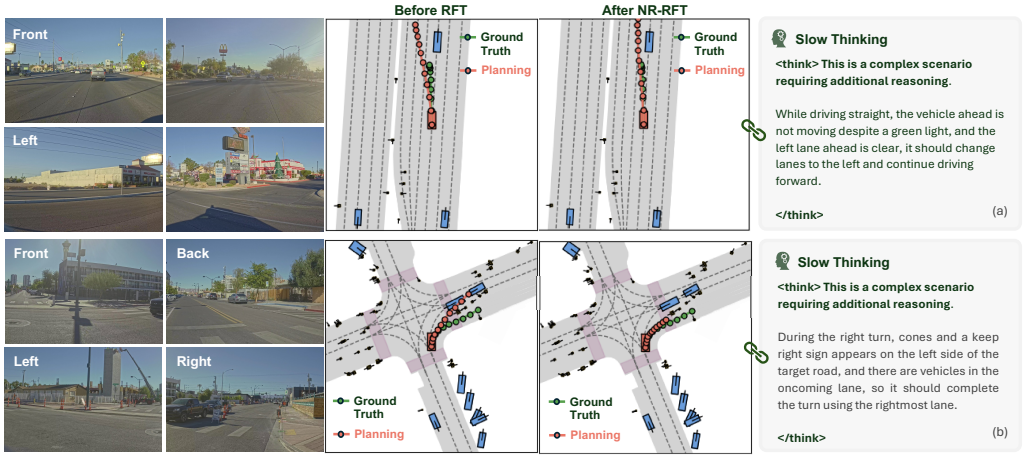



Figure 5: Comparison of planning and reasoning performance of SPANVLA in **negative and recovery samples** of nuReasoning-NR before and after NR-RFT.

maintaining a trajectory-generation latency of 0.08 s. This shows that past motion provides useful guidance for future trajectory generation without increasing the measured inference time.

**Sparse KV-Cache Conditioning.** Conditioning on every second VLM layer achieves 90.8 PDMS, compared with 82.2 using every fourth layer and 79.3 using only the final layer. These results support accessing representations across VLM depths, while showing that overly sparse conditioning degrades planning. The full-caching baseline uses a larger bridge without historical initialization, achieves 88.1 PDMS with 0.18 s latency. Our configuration achieves higher planning performance at 0.08 s, providing an effective balance between contextual information and computation.

### 5.5 QUALITATIVE RESULTS

Figures 1 and 5 illustrate how NR-RFT improves planning in negative and recovery scenarios. In the negative scenarios, the updated policy predicts an earlier merge before a construction-zone lane closure (Figure 1, upper left) and a more decisive lane change around a stationary lead vehicle (Figure 5(a)). In the recovery scenarios, it predicts a return to the original lane after passing an obstructing truck (Figure 1, lower left) and a right turn aligned with the available lane through a constrained intersection (Figure 5(b)). Additional examples are provided in Appendix F.4.

## 6 CONCLUSION

We presented SPANVLA, a robust and efficient driving VLA, together with **nuReasoning-NR**, the first real-world driving dataset and benchmark for learning from negative and recovery samples. NR-RFT exploits their asymmetric supervision by penalizing undesirable behaviors and rewarding expert corrections, with advantage correction for difficult rollout groups. For efficient action generation, SPANVLA retains autoregressive reasoning and uses a lightweight flow-matching bridge conditioned on sparse-layer KV caches and initialized from the historical trajectory. Experiments on NAVSIM v1/v2, nuReasoning, and nuReasoning-NR demonstrate strong planning performance, improved robustness in challenging scenarios, and efficient trajectory generation. These results highlight the value of real-world negative and recovery supervision for driving VLAs. The dataset and benchmark code will be released to support further research.

### AI USE STATEMENT

We used Gemini-3-Pro to generate reasoning annotations for the positive reasoning subset of nuReasoning-NR. The annotation pipeline includes automated checks of output format and consistency with reference actions, followed by human quality auditing. We also used GPT-6 to assist with manuscript drafting and polish. The authors reviewed and revised the manuscript content and take responsibility for the methodology, experimental results, and claims.

### ETHICS STATEMENT

This research aims to improve driving robustness by learning from recorded suboptimal behaviors and expert recovery actions. Before publicly releasing nuReasoning-NR, we will process the recordings to obscure identifying information, including faces and license plates.

## REFERENCES

Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1933–1943. IEEE, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Tianhui Cai, Yifan Liu, Zewei Zhou, Haoxuan Ma, Seth Z Zhao, Zhiwen Wu, and Jiaqi Ma. Driving with regulation: Interpretable decision-making for autonomous vehicles with retrieval-augmented reasoning via llm. *arXiv preprint arXiv:2410.04759*, 2024.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

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.

OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. https://github.com/OpenDriveLab/OpenScene, 2023.

Chenxu Dang, Sining Ang, Yongkang Li, Haochen Tian, Jie Wang, Guang Li, Hangjun Ye, Jie Ma, Long Chen, and Yan Wang. Drivefine: Refining-augmented masked diffusion vla for precise and robust driving. *arXiv preprint arXiv:2602.14577*, 2026.

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.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021.

Shiyu Fang, Yiming Cui, Haoyang Liang, Chen Lv, Peng Hang, and Jian Sun. Corevla: A dual-stage end-to-end autonomous driving framework for long-tail scenarios via collect-and-refine. *arXiv preprint arXiv:2509.15968*, 2025.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. *arXiv preprint arXiv:2510.04333*, 2025a.

Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *IEEE Robotics and Automation Letters*, 11(1):226–233, 2025b.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Hongwei Xie, Bing Wang, Guang Chen, Dingkang Liang, and Xiang Bai. Minddrive: A vision-language-action model for autonomous driving via online reinforcement learning. *arXiv Preprint arXiv:2512.13636*, 2025.

Yinfeng Gao, Deqing Liu, Qichao Zhang, Yupeng Zheng, Haochen Tian, Guang Li, Hangjun Ye, Long Chen, Da-Wei Ding, and Dongbin Zhao. Learning from mistakes: Post-training for driving vla with takeover data. *arXiv preprint arXiv:2603.14972*, 2026.

Gemini Team, Google DeepMind. Gemini 3: A new era of intelligence with gemini 3. *Technical Report*, 2025. URL https://deepmind.google/models/gemini/.

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

Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. ipad: Iterative proposal-centric end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 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.

Zhiyu Huang, Johnson Liu, Rui Song, Zewei Zhou, Ruining Yang, Yun Zhang, Tianhui Cai, Hanyin Zhang, Mingxuan Gao, Valeria Xu, et al. nureasoning: A reasoning-centric dataset and benchmark for long-tail autonomous driving. *arXiv preprint arXiv:2605.31572*, 2026.

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

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. pi0.6: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

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

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8350, 2023.

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. *arXiv preprint arXiv:2410.22313*, 2024.

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*, 2025b.

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*, 134(9):407, 2026.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 629–636. IEEE, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. *URL https://arxiv. org/abs/2502.19645*, 2016.

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. *arXiv preprint arXiv:2406.09246*, 2024.

Mingyue Lei, Zewei Zhou, Hongchen Li, Jiaqi Ma, and Jia Hu. Risk map as middleware: Toward interpretable cooperative end-to-end autonomous driving for risk-aware planning. *IEEE Robotics and Automation Letters*, 11(1):818–825, 2025.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. *arXiv preprint arXiv:2510.12796*, 2025a.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27137–27146, 2025b.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. *arXiv preprint arXiv:2506.08052*, 2025c.

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*, 2024.

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. *arXiv preprint arXiv:2411.15139*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Deqing Liu, Yinfeng Gao, Deheng Qian, Qichao Zhang, Xiaoqing Ye, Junyu Han, Yupeng Zheng, Xueyi Liu, Zhongpu Xia, Dawei Ding, et al. Takead: Preference-based post-optimization for end-to-end autonomous driving with expert takeover data. *IEEE Robotics and Automation Letters*, 11 (2):1738–1745, 2025a.

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

Xinyi Liu, Mohammadreza Fani Sani, Zewei Zhou, Julius Wirbel, Bahram Zarrin, and Roberto Galeazzi. Robopilot: Generalizable dynamic robotic manipulation with dual-thinking modes. *arXiv preprint arXiv:2510.00154*, 2025b.

Yingzi Ma, Yulong Cao, Wenhao Ding, Shuibai Zhang, Yan Wang, Boris Ivanovic, Ming Jiang, Marco Pavone, and Chaowei Xiao. dvlm-ad: Enhance diffusion vision-language-model for driving via controllable reasoning, 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.

Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro G Allievi, Senem Velipasalar, and Liu Ren. Vlp: Vision language planning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14760–14769, 2024.

Sung-Yeon Park, Can Cui, Yunsheng Ma, Ahmadreza Moradipari, Rohit Gupta, Kyungtae Han, and Ziran Wang. Nuplanqa: A large-scale dataset and benchmark for multi-view driving scene understanding in multi-modal large language models. *arXiv preprint arXiv:2503.12772*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Ishaan Rawal, Shubh Gupta, Yihan Hu, and Wei Zhan. Nord: A data-efficient vision-language-action model that drives without reasoning. *arXiv preprint arXiv:2602.21172*, 2026.

Luke Rowe, Rodrigue de Schaetzen, Roger Girgis, Christopher Pal, and Liam Paull. Poutine: Vision-language-trajectory pre-training and reinforcement learning post-training enable robust end-to-end autonomous driving. *arXiv preprint arXiv:2506.11234*, 2025.

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

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European Conference on Computer Vision*, pp. 256–274. Springer, 2024.

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

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.

Haitong Wang, Aaron Hao Tan, and Goldie Nejat. Navformer: A transformer architecture for robot target-driven navigation in unknown and dynamic environments. *IEEE Robotics and Automation Letters*, 9(8):6808–6815, 2024a.

Renxi Wang, Haonan Li, Xudong Han, Yixuan Zhang, and Timothy Baldwin. Learning from failure: Integrating negative examples when fine-tuning large language models as agents. *arXiv preprint arXiv:2402.11651*, 2024b.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*, 2024c.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

Chengen Xie, Bin Sun, Tianyu Li, Junjie Wu, Zhihui Hao, XianPeng Lang, and Hongyang Li. Latentvla: Efficient vision-language models for autonomous driving via latent action prediction. *arXiv preprint arXiv:2601.05611*, 2026.

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 *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 1001–1009, 2025.

Zebin Xing, Yupeng Zheng, Qiang Chen, Linbo Wang, Yichen Zhang, Pengxuan Yang, Junli Wang, Deheng Qian, Xiaoqing Ye, Junyu Han, et al. Drivevla-m0: Failure-aware memory augmentation for autonomous driving. *arXiv preprint arXiv:2608.10413*, 2026.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. *arXiv preprint arXiv:2510.26125*, 2025.

Yi Xu, Yuxin Hu, Zaiwei Zhang, Gregory P Meyer, Siva Karthik Mustikovela, Siddhartha Srinivasa, Eric M Wolff, and Xin Huang. Vlm-ad: End-to-end autonomous driving through vision-language model supervision. *arXiv preprint arXiv:2412.14446*, 2024.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning. *arXiv preprint arXiv:2506.06659*, 2025.

Zhaoshu Yu, Bo Wang, Pengpeng Zeng, Haonan Zhang, Ji Zhang, Lianli Gao, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. A survey on efficient vision-language-action models. *arXiv preprint arXiv:2510.24795*, 2025.

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.

Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, Xing Wei, and Ning Guo. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025.

Lingjun Zhang, Yujian Yuan, Changjie Wu, Xinyuan Chang, Xin Cai, Shuang Zeng, Linzhe Shi, Sijin Wang, Hang Zhang, and Mu Xu. Minddriver: Introducing progressive multimodal reasoning for autonomous driving. *arXiv preprint arXiv:2602.21952*, 2026.

Seth Z Zhao, Luobin Wang, Hongwei Ruan, Yuxin Bao, Yilan Chen, Ziyang Leng, Abhijit Ravichandran, Honglin He, Zewei Zhou, Xu Han, et al. Bridgesim: Unveiling the ol-cl gap in end-to-end autonomous driving. *arXiv preprint arXiv:2604.10856*, 2026.

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

Xin Zhou, Zongchuang Zhao, Zhibo Yang, Mingsheng Li, Humen Zhong, Shuai Bai, Du Chu, Ruizhe Chen, Zhaohai Li, Jun Tang, et al. Qwen-drive-1.0: An initial step towards a vision-language foundation model for autonomous driving. *arXiv preprint arXiv:2609.00111*, 2026.

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

Zewei Zhou, Hao Xiang, Zhaoliang Zheng, Seth Z Zhao, Mingyue Lei, Yun Zhang, Tianhui Cai, Xinyi Liu, Johnson Liu, Maheswari Bajji, et al. V2XPnP: Vehicle-to-everything spatio-temporal fusion for multi-agent perception and prediction. *arXiv preprint arXiv:2412.01812*, 2024.

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

Zewei Zhou, Seth Z Zhao, Tianhui Cai, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Turbotrain: Towards efficient and balanced multi-task learning for multi-agent perception and prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4391–4402, 2025c.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving. *arXiv preprint arXiv:2512.07745*, 2025.

# SpanVLA Appendix


**A Additional Related Work** **15**

**B nuReasoning-NR Dataset and Benchmark** **16**

B.1 Scenario Coverage and Data Representation . . . . . 16

B.2 Real-World Negative and Recovery Data . . . . . 16

B.3 Positive Reasoning Annotations . . . . . 17

B.4 Benchmark and Evaluation Protocol . . . . . 17

**C Details of SFT** **18**

C.1 Implementation Details . . . . . 18

C.2 Action Codebook . . . . . 18

**D Details of RFT with Negative-Recovery Samples** **19**

D.1 GRPO Objective . . . . . 19

D.2 Reward Function . . . . . 19

D.3 Kullback–Leibler (KL) Divergence . . . . . 21

**E Additional Implementation Details** **21**

**F Additional Experimental Results** **21**

F.1 Detailed Main Results . . . . . 21

F.2 Additional Ablation Studies for Action Bridging . . . . . 23

F.3 Additional Ablation Studies for RFT . . . . . 23

F.4 Additional Qualitative Results . . . . . 25

**G Limitations and Future Work** **25**


## A Additional Related Work

**Reinforcement Fine-Tuning.** Reinforcement fine-tuning (RFT) has improved general reasoning (Guo et al., 2025) and driving-policy optimization (Fu et al., 2025; Rawal et al., 2026; Jiang et al., 2025a; Zou et al., 2025). Most driving approaches build on expert-demonstration datasets without explicit real-world negative-recovery supervision. Early takeover-based methods, including TakeAD and CoReVLA, use preference optimization to learn from expert corrections collected in simulation (Liu et al., 2025a; Fang et al., 2025; Rafailov et al., 2023). TakeVLA further explores reinforcement fine-tuning in reconstructed takeover scenarios (Gao et al., 2026). These studies motivate learning from mistakes, while leaving real-world negative and recovery data largely unexploited. SpanVLAaddresses this gap through NR-RFT, which combines sample-type-specific penalties and rewards with an advantage correction for difficult rollout groups to stabilize the training. Together with nuReasoning-NR, it provides real-world data, a tailored training method, and a dedicated benchmark for studying negative-recovery learning.

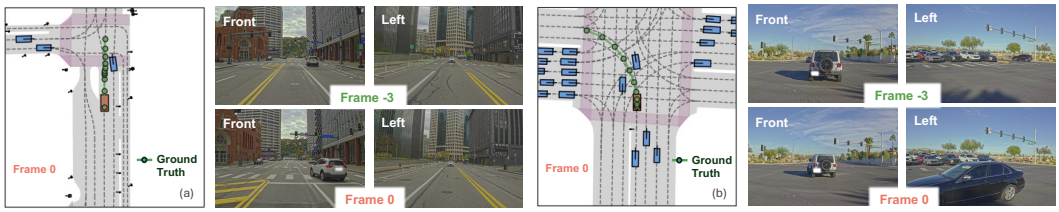



Figure S1: Complex scenario samples (positive samples) of nuReasoning-NR dataset. (a) Cut-in scenario: A vehicle from the front-right abruptly cuts in at the intersection. (b) Intersection scenario: The ego vehicle has already entered the intersection; despite the red traffic signal, it should yield appropriately and follow the leading vehicle to complete the left turn. Frame $-3$ and Frame 0 denote the first history frame and the current frame for the camera streams.

# B nuReasoning-NR Dataset and Benchmark

nuReasoning-nr extends nuReasoning (Huang et al., 2026) with real-world negative and expert recovery samples, providing the first real-world driving dataset and benchmark for learning from both forms of supervision. It contains 36K samples: 30K positive reasoning samples, 3K negative samples, and 3K expert recovery samples. The three subsets cover common categories of interaction-intensive driving scenarios and provide complementary supervision for learning expert driving decisions, suppressing undesirable behaviors, and acquiring corrective actions. They share a common data representation with eight surrounding camera streams, reference ego trajectories, object annotations, and HD maps, supporting both policy learning and planning evaluation.

These samples provide trajectory-level supervision with distinct behavioral semantics. Negative trajectories identify inefficient, hesitant, or potentially unsafe behaviors to suppress, while expert recovery trajectories specify corrective actions from difficult driving states. Their event-level correspondence links suboptimal behavior to the subsequent expert correction, with each segment retaining its own timestamp and observation history. This structure motivates the asymmetric supervision in NR-RFT.

## B.1 Scenario Coverage and Data Representation

**Scenario Coverage.** We focus on complex and long-tail driving scenarios and collect 36K driving samples from Las Vegas, Boston, Pittsburgh, and Singapore, covering ego lane changes, lane bias, vulnerable road users (VRU), construction zones, stop signs, cut-in, and lead vehicle braking scenarios, as shown in Figure 1. The history horizon of nuReasoning-NR dataset is 1.5 seconds, and the future horizon is 5 seconds (2 Hz) to include the key driving elements and events. Two typical complex driving scenarios of nuReasoning-NR dataset are shown in Figure S1.

**Sensor Settings.** The vision data of nuReasoning-NR consists of eight surrounding camera streams, including front, front-left, front-right, left, right, back, back-left, and back-right. In addition, we provide annotations (e.g., trajectory, size, and category) for each object, e.g., vehicles, pedestrians, and traffic cones. Moreover, the HD map data is provided for these driving scenarios following the nuplan map format (Karnchanachari et al., 2024) with driving lane, driving direction, and traffic light status.

## B.2 Real-World Negative and Recovery Data

**Event-Based Construction.** We curate negative and recovery samples from early-stage exploratory real-world driving tests. For each selected event, a negative segment records suboptimal ego behavior, and its corresponding recovery segment records the subsequent expert correction. Each segment forms a separate sample with its own observation history and reference trajectory. The resulting subsets contain 3K negative samples and 3K expert recovery samples. The correspondence is temporal and event-based: the negative and recovery segments occur at different moments and therefore have different observation inputs. They do not constitute two alternative trajectories conditioned on an identical driving state.

**Supervision Semantics.** Negative trajectories identify behaviors to avoid, including inefficient or hesitant maneuvers and potentially unsafe actions. Recovery trajectories demonstrate how an expert corrects behavior from a difficult driving state. Accordingly, a recorded negative trajectory serves as an undesirable reference, whereas an expert recovery trajectory serves as a desirable reference. These distinct meanings motivate the negative-behavior penalty and recovery-behavior reward in NR-RFT. Additional examples are shown in Figures S6 and S7.

### B.3 Positive Reasoning Annotations

The positive subset provides expert trajectories and reasoning annotations for supervised fine-tuning. We use Gemini-3-Pro (Gemini Team, Google DeepMind, 2025) to generate structured annotations linking driving decisions to decision-relevant scene elements. Representative scenarios and annotations are shown in Figures S1 and S2.

**Annotation Inputs.** The annotation model receives multi-view camera observations, ego states, and high-level driving instructions. Following Zhou et al. (2025b), recorded expert actions are also supplied as reference cues to encourage consistency between the explanations and demonstrated behavior. These future actions are used for offline annotation; they are not provided as inputs to the driving policy at inference.

**System Prompt and Output Structure.** The system prompt specifies the annotation VLM's role, annotation task, expected CoT reasoning format, and expert examples of CoT reasoning. The VLM role is an advanced full self-driving system, and the task is to determine the optimal driving decision based on the driving video from camera sensors and vehicle status inputs with CoT reasoning. Refer to the reasoning format of Alpamayo (Wang et al., 2025), we also leverage the compacted reasoning trace to keep the reasoning short and reduce redundant reasoning. Although the VLA model utilizes the short reasoning trace as the reasoning supervision signal, we still need a comprehensive reasoning output with critical components and driving decisions to ensure the reasoning quality. Moreover, we carefully provide some typical expert annotation examples for annotation VLM as a reference. Then, the model requires generation with JSON format to reason critical components, driving decision, and reasoning trace one by one.

**Quality Control.** We evaluate the reasoning quality based on the accuracy and completeness for the three parts: critical components, driving decision, and reasoning trace. The evaluation follows a strict binary protocol, and any error in each component results in a score of 0 for the corresponding part. Our human expert assessed 250 randomly selected samples, yielding an overall accuracy of 80.4% in our complex scenarios. Erroneous samples were subsequently corrected or removed to construct the final high-quality reasoning dataset. Figure S2 shows two typical CoT data in nuReasoning-NR dataset.

### B.4 Benchmark and Evaluation Protocol

**Evaluation Subsets.** The benchmark contains 300 negative samples and 300 recovery samples, all included in the 36K dataset total. Negative and recovery samples correspond to different timestamps, each with its own observation history and reference trajectory. During dataset curation, we prioritized scenario-level separation between training and evaluation to minimize overlap and avoid assigning negative and recovery segments from the same scenario to different splits. The negative subset evaluates planning in situations associated with recorded suboptimal behavior. The recovery subset evaluates corrective planning from the difficult states in which expert recovery begins. Results are reported separately for the two subsets and averaged across the benchmark.

**Evaluation Pipeline.** We adapt the NAVSIM evaluation framework (Dauner et al., 2024) to nuReasoning-nr using its object annotations and HD maps. The object trajectories, sizes, and categories, together with HD maps, enable candidate ego trajectories to be evaluated against surrounding traffic and road geometry. This supports quantitative assessment of how negative and recovery supervision affects safety, progress, and comfort in challenging driving scenarios. The benchmark reports PDMS and its component metrics, covering collision avoidance, drivable-area compliance,

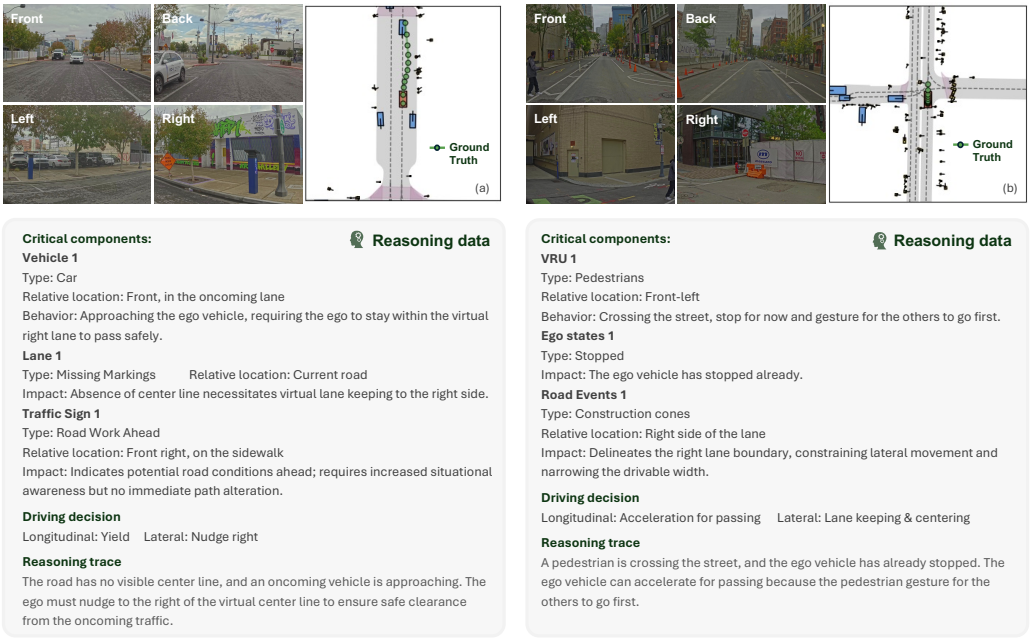



Figure S2: Reasoning data scenario samples (positive samples) of nuReasoning-NR dataset. (a) An oncoming vehicle encroaches into the current ego-vehicle's lane; the ego vehicle should nudge to the right to yield the oncoming vehicle. (b) A pedestrian gestures for the ego vehicle to proceed; since the ego vehicle has already come to a stop, it can safely start and pass.

ego progress, time-to-collision, and comfort. The reported benchmark scores measure planning performance; the additional negative and recovery reward terms are used for training. This evaluation supports systematic comparison of positive-only learning and learning with negative and recovery supervision.

# C Details of SFT

## C.1 Implementation Details

SpanVLA supports mixed vision and textual prompt inputs. First, it takes multi-frame and multi-view image data as the vision inputs $\mathcal{V}^t = \bigcup_i \{c_i^\tau\}_{\tau=t-T_h}^t$ where $c_i$ is each camera stream, which consists of four history frames per camera stream sampled at 2hz. For language inputs $\mathcal{T}^t = \{I^t, \{S_{\text{ego}}^\tau\}_{\tau=t-T_h}^t\}$, the system prompt describes the role and reasoning and planning task, and the high-level instruction $I$ (e.g., turn right), and the ego states $S_{\text{ego}}$ are inputted and tokenized as text. The latter consists of ego acceleration, history velocity, and trajectory.

Our SpanVLA model is trained with supervision finetuning via a two-stage paradigm. First, we leverage the CoT reasoning data and action tokens from the ground-truth trajectory to train the VLM backbone, enabling the backbone with the reasoning capability and aligning it with planning. We directly utilize and freeze the vision encoder of Qwen2.5-VL-3B and only train the LLM module. Then, the action expert is finetuned with the frozen VLM backbone without action token generation.

## C.2 Action Codebook

We leverage dual-representation for future action, where one is the continuous waypoint for the flow matching, and one is the discrete action token for autoregressive decoding. For the latter, we leverage the same codebook as (Zhou et al., 2025b), comprising 2048 action tokens for 0.5 second vehicle motion from the Waymo Open Motion dataset (Ettinger et al., 2021).

# D Details of RFT with Negative-Recovery Samples

## D.1 GRPO Objective

We introduce Negative-Recovery Reinforcement Fine-tuning (NR-RFT) to suppress negative driving behaviors and reinforce expert recovery actions. During training, we update the VLM backbone using positive, negative, and recovery samples. We adopt group relative policy optimization (GRPO) (Shao et al., 2024; Jiang et al., 2025b) for post-training.

Given a scenario query $q = (\mathcal{V}^t, \mathcal{T}^t)$, we sample $G$ candidate token sequences $\mathcal{Y} = \{y_1, \ldots, y_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$. Each $y_i$ contains reasoning when present and action tokens. Following the notation of the main paper, we denote the trajectory decoded from its action tokens by $o_i = \{\mathbf{p}_{i,t}\}_{t=0}^{T}$, where $\mathbf{p}_{i,t} \in \mathbb{R}^2$ is the ego center position at time step $t$. Policy probabilities and reasoning penalties are evaluated on $y_i$, while driving and reference-matching rewards are evaluated on $o_i$. We optimize the current policy $\pi_\theta$ using

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\mathcal{Y}}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\mathcal{J}_i^R - \beta\mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot \mid q)\,\|\,\pi_{\text{ref}}(\cdot \mid q)\big)\right)\right]. \tag{S1}$$

The expectation is over training queries and output groups sampled from the old policy. The clipped reward objective is

$$\mathcal{J}_i^R = \min\left(\frac{\pi_\theta(y_i \mid q)}{\pi_{\theta_{\text{old}}}(y_i \mid q)}A_i,\quad \text{clip}\left(\frac{\pi_\theta(y_i \mid q)}{\pi_{\theta_{\text{old}}}(y_i \mid q)}, 1-\epsilon, 1+\epsilon\right)A_i\right). \tag{S2}$$

Here, $\epsilon$ controls clipping, $\beta$ weights KL regularization, and $\pi_{\text{ref}}$ is the reference policy obtained from SFT. Standard GRPO computes the advantage as

$$A_i^{\text{GRPO}} = \frac{r_i - \mu}{\sigma + \varepsilon_{\text{num}}}, \qquad \mu = \frac{1}{G}\sum_{j=1}^{G} r_j, \tag{S3}$$

where $r_i$ is the total reward associated with $y_i$ and its decoded trajectory $o_i$, $\sigma = \text{std}(\{r_j\}_{j=1}^{G})$, and $\varepsilon_{\text{num}} > 0$ ensures numerical stability. NR-RFT uses the sample-type-dependent rewards defined below and replaces $A_i^{\text{GRPO}}$ with the corrected advantage $A_i$ defined in the main paper. For the advantage correction, the all-zero condition is evaluated on the trajectory driving rewards $\{r_{\text{Driving},i}\}_{i=1}^{G}$ (PDMS or EPDMS), before applying the negative, recovery, and CoT terms.

## D.2 Reward Function

Following the main paper, the total reward combines driving quality, negative-behavior suppression, expert recovery, and reasoning cost:

$$r_i = r_{\text{Driving},i} - \lambda_N r_{\text{Negative},i} + \lambda_R r_{\text{Recover},i} - \lambda_C r_{\text{CoT},i}. \tag{S4}$$

The non-negative coefficients $\lambda_N$, $\lambda_R$, and $\lambda_C$ weight the negative-behavior penalty, recovery-behavior reward, and CoT penalty, respectively. The trajectory-based terms are computed from $o_i$, and $r_{\text{CoT},i} = r_{\text{CoT}}(y_i)$ is computed from its associated reasoning. We define each term below.

### D.2.1 Driving Reward

The reward function provides the primary training signal during RFT and guides policy optimization toward high-quality driving behaviors. To improve overall driving performance beyond merely imitating expert trajectories, we adopt a hybrid reward design in which the main driving reward, denoted by $r_{\text{Driving}}$, measures trajectory quality from the perspectives of safety, rule compliance, comfort, and progress.

For the NAVSIM v1 benchmark, we use the Predictive Driver Model Score (PDMS) (Dauner et al., 2024) as the primary driving reward. For the NAVSIM v2 benchmark, we use its extended version, the Extended Predictive Driver Model Score (EPDMS), which introduces additional rule-compliance metrics. For the NAVSIM v1 setting, the driving reward is defined as

$$r_{\text{Driving}} = \text{PDMS}, \quad \text{PDMS} = \text{NC} \times \text{DAC} \times \left( \frac{5\,\text{TTC} + 2\,\text{C} + 5\,\text{EP}}{12} \right), \tag{S5}$$

where NC denotes No at-fault Collision, DAC denotes Drivable Area Compliance, EP denotes Ego Progress, C denotes Comfort, and TTC denotes Time-to-Collision. Following the standard definition, these sub-scores are normalized to $[0, 1]$, and their weighted combination yields a comprehensive measure of driving quality. If the planner fails to produce a valid output, we assign a reward of 0.

For the NAVSIM v2 benchmark, we adopt the Extended Predictive Driver Model Score (EPDMS) as the driving reward (Cao et al., 2025). Compared with PDMS, EPDMS further incorporates Driving Direction Compliance (DDC) and Traffic Light Compliance (TLC), and replaces the single comfort term with more fine-grained comfort metrics, including Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC). Specifically, EPDMS is defined as

$$\begin{aligned} r_{\text{Driving}} = \text{EPDMS} = &\left( \prod_{m \in \{\text{NC,DAC,DDC,TLC}\}} \text{filter}_m(\text{agent}, \text{human}) \right) \\ &\cdot \left( \frac{\sum_{m \in \{\text{TTC,EP,HC,LK,EC}\}} w_m \, \text{filter}_m(\text{agent}, \text{human})}{\sum_{m \in \{\text{TTC,EP,HC,LK,EC}\}} w_m} \right), \end{aligned} \tag{S6}$$

Their corresponding weights are $w_{\text{TTC}} = 5$, $w_{\text{EP}} = 5$, $w_{\text{HC}} = 2$, $w_{\text{LK}} = 2$, and $w_{\text{EC}} = 2$. Here, the filtering function is defined as

$$\text{filter}_m(\text{agent}, \text{human}) = \begin{cases} 1.0, & \text{if } m(\text{human}) = 0, \\ m(\text{agent}), & \text{otherwise.} \end{cases} \tag{S7}$$

For negative samples, we omit the ego-progress (EP) term from the driving reward because the recorded trajectory is an undesirable reference.

### D.2.2 Negative-Behavior Penalty and Recovery-Behavior Reward

For each predicted trajectory $o_i$, we compute the negative-behavior penalty or recovery-behavior reward according to the sample type. Let $o^{\text{ref}} = \{\mathbf{p}_t^{\text{ref}}\}_{t=0}^{T}$ denote the reference trajectory associated with the current training sample. We define the average distance between $o_i$ and $o^{\text{ref}}$ as

$$d(o_i, o^{\text{ref}}) = \frac{1}{T+1} \sum_{t=0}^{T} \left\| \mathbf{p}_{i,t} - \mathbf{p}_t^{\text{ref}} \right\|_2. \tag{S8}$$

Based on this distance, we define a unified reference-matching term

$$r_{\text{Ref}}(q, o_i) = \text{clip}\left( 1 - \frac{d(o_i, o^{\text{ref}})}{\delta}, 0, 1 \right), \tag{S9}$$

where $\delta$ is a distance threshold. This term reaches its maximum value of 1 when $o_i$ exactly matches the reference trajectory, decreases linearly as the deviation grows, and becomes 0 when $d(o_i, o^{\text{ref}}) \geq \delta$.

For negative samples, $o^{\text{ref}}$ corresponds to an undesirable behavior trajectory, and $r_{\text{Negative},i} = r_{\text{Ref}}(q, o_i)$ enters Eq. equation S4 with coefficient $-\lambda_N$. For recovery samples, $o^{\text{ref}}$ corresponds to a desirable recovery trajectory, and $r_{\text{Recover},i} = r_{\text{Ref}}(q, o_i)$ enters with coefficient $+\lambda_R$. Each term is zero for other sample types. The two terms share the same reference-matching function but use opposite signs to suppress undesirable behavior and reinforce expert corrections.

### D.2.3 Reasoning Penalty and Action-Reasoning Alignment

To discourage unnecessarily long reasoning and achieve adaptive reasoning, we apply a length-based penalty on the generated reasoning. Let $L(y_i)$ denote the length in tokens of the CoT part in output $y_i$. We define the CoT penalty as a sigmoid function:

$$r_{\text{CoT}}(y_i) = \frac{1}{1 + \exp(-(L(y_i) - L_{\text{tol}})\gamma)}, \tag{S10}$$

where $L_{\text{tol}}$ is the reasoning length at which the penalty equals $0.5$, and $\gamma > 0$ controls the steepness. The penalty increases smoothly with reasoning length, discouraging unnecessarily long reasoning.

In addition to length regularization, we enforce an action-reasoning alignment reward. Intuitively, the reasoning should be consistent with the high-level maneuver implied by the predicted trajectory. We implement a rule-based detector that maps (i) the reasoning text in $y_i$ to a maneuver label via keyword matching, and (ii) the predicted trajectory $o_i$ to a maneuver label via simple geometric heuristics. We then penalize inconsistent pairs.

We extract a maneuver label $\hat{k}_{\text{text}} \in \mathcal{K}$ from the CoT via keyword matching, where $\mathcal{K}$ is a small set of maneuver types (*straight*, *left turn*, *right turn*, *change lane to left*, *change lane to right*, *accelerate*, *decelerate*). For instance, keywords such as "straight" or "go forward" map to *straight*; "left turn" or "turn left" map to *left turn*; and "accelerate" or "speed up" map to *accelerate*.

A scene may contain several actions and objects, although we assign it a single maneuver label. When several maneuver cues are found, we use a fixed priority order to select the label that best describes the main focus of the scene. The same priority order is used for both the reasoning and the trajectory. In parallel, we infer a maneuver label $\hat{k}_{\text{traj}}$ from the predicted trajectory $o_i$ using simple geometric heuristics, including the signed heading change over the horizon for left/right/straight and the speed change for accelerate/decelerate. Concretely, we compute the net heading change and net speed change over the prediction horizon, and map them to maneuver types using fixed thresholds. We then check alignment between $\hat{k}_{\text{text}}$ and $\hat{k}_{\text{traj}}$. If they are inconsistent, we apply a fixed penalty by overriding the CoT term, discouraging misaligned reasoning–action pairs during RFT.

### D.3 Kullback–Leibler (KL) Divergence

Following (Zhou et al., 2025b), we include a KL regularization term in the GRPO objective in Eq. equation S1 to constrain the updated policy $\pi_\theta$ to stay close to the SFT reference policy $\pi_{\text{ref}}$. For each scenario query $q = (\mathcal{V}^t, \mathcal{T}^t)$ and a sampled token sequence $y_i \in \mathcal{Y}$, the KL divergence is formulated as:

$$\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(y_i|q)}{\pi_\theta(y_i|q)} - \log\left(\frac{\pi_{\text{ref}}(y_i|q)}{\pi_\theta(y_i|q)}\right) - 1, \tag{S11}$$

where $\theta$ denotes the parameters of the current policy $\pi_\theta$, and $\pi_{\text{ref}}$ is the reference policy obtained from the SFT stage. This regularization penalizes large deviations from $\pi_{\text{ref}}$, stabilizing policy updates and retaining useful knowledge learned during SFT while optimizing for the driving reward.

## E Additional Implementation Details

The default configuration includes three camera views, i.e., front, front-left, and front-right, and can be easily extended to additional camera streams.

## F Additional Experimental Results

Our main SpanVLA RFT comparisons use a two-stage schedule with a total of three epochs. In the first epoch, we train exclusively on positive samples to establish a basic policy for successful driving. In the second and third epochs, we vary the training-data composition across the RFT variants: the positive-only variant continues using positive samples, the positive–negative variant uses both positive and negative samples, and the full NR variant uses positive, negative, and recovery samples. The three-epoch schedule is shared across these comparisons, with the data composition varied only in the second stage. We report PDMS for NAVSIM v1 and EPDMS for NAVSIM v2, as defined in Equations (S5) and (S6), respectively. The nuReasoning-NR diagnostic benchmark is evaluated using PDMS.

### F.1 Detailed Main Results

**NAVSIM Benchmarks.** We report the detailed main results of SpanVLA. For a fair comparison, we only compare SpanVLA with methods that use neither additional information nor trajectory-selection modules. Table S1 reports the results on the `navtest` benchmark of NAVSIM v1 against

Table S1: Comparison with SOTA methods on the **NAVSIM v1** (`navtest`). PDMS (Predictive Driver Model Score), NC (No Collision), DAC (Drivable Area Compliance), EP (Ego Progress), TTC (Time-to-Collision), Comf. (Comfort). * denotes best-of-N selection with N = 6, where the candidate with the highest PDMS is chosen.

| Methods | Cam. | Lid. | PDMS ↑ | NC ↑ | DAC ↑ | EP ↑ | TTC ↑ | Comf. ↑ |
|---|---|---|---|---|---|---|---|---|
| *Conventional End-to-end-based Methods* | | | | | | | | |
| TransFuser (Chitta et al., 2022) | ✓ | ✓ | 84.0 | 97.8 | 92.6 | 78.9 | 92.9 | **100.0** |
| DRAMA (Yuan et al., 2024) | ✓ | ✓ | 86.9 | 98.2 | 95.2 | 81.3 | 94.2 | **100.0** |
| Hydra-MDP (Li et al., 2024) | ✓ | ✓ | 86.5 | 98.3 | 96.0 | 78.7 | 94.6 | **100.0** |
| DiffusionDrive (Liao et al., 2024) | ✓ | ✓ | 88.1 | 98.2 | 96.2 | 82.2 | 94.7 | **100.0** |
| WoTE (Li et al., 2025b) | ✓ | ✓ | 88.3 | 98.5 | 96.8 | 81.9 | 94.4 | 99.9 |
| *VLA-based Methods* | | | | | | | | |
| ReCogDrive (Li et al., 2025c) | ✓ | - | 89.6 | 98.2 | 97.8 | 83.5 | 95.2 | 99.8 |
| DriveVLA-W0 (**?**) | ✓ | - | 90.2 | 98.7 | **99.1** | 83.3 | 95.3 | 99.3 |
| AutoVLA (Zhou et al., 2025b) | ✓ | - | 89.1 | 98.4 | 95.6 | 81.9 | **98.0** | 99.9 |
| ***Ours*** | | | | | | | | |
| SpanVLA (One-shot) | ✓ | - | 82.1 | 97.5 | 90.8 | 76.9 | 93.7 | 99.5 |
| SpanVLA (Post-RFT) | ✓ | - | **90.8** | **99.6** | 99.1 | 81.1 | **98.1** | **100.0** |
| SpanVLA* | ✓ | - | **93.3** | **99.8** | 99.8 | 85.0 | **99.2** | **100.0** |

Table S2: Comparison with SOTA methods on the **NAVSIM v2** (`navtest`). EPDMS (Extended Predictive Driver Model Score), NC (No Collision), DAC (Drivable Area Compliance), DDC (Driving Direction Compliance), TLC (Traffic Light Compliance), EP (Ego Progress), TTC (Time-to-Collision), LK (Lane Keeping), HC (History Comfort), EC (Extended Comfort). * denotes best-of-N selection with N = 6, where the candidate with the highest PDMS is chosen.

| Methods | EPDMS↑ | NC↑ | DAC↑ | DDC↑ | TLC↑ | EP↑ | TTC↑ | LK↑ | HC↑ | EC↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| *Conventional End-to-end-based Methods* | | | | | | | | | | |
| TransFuser (Chitta et al., 2022) | 76.7 | 96.9 | 89.9 | 97.8 | 99.7 | 87.1 | 95.4 | 92.7 | **98.3** | 87.2 |
| DiffusionDrive (Liao et al., 2024) | 84.5 | 98.2 | 95.9 | **99.4** | **99.8** | 87.5 | 97.3 | 96.8 | **98.3** | **87.7** |
| Hydra-MDP++ (Li et al., 2024) | 81.4 | 97.2 | 97.5 | **99.4** | 99.6 | 83.1 | 96.5 | 94.4 | 98.2 | 70.9 |
| Drivesuprim (Yao et al., 2025) | 83.1 | 97.5 | 96.5 | **99.4** | 99.6 | **88.4** | 96.6 | 95.5 | **98.3** | 77.0 |
| ARTEMIS (Feng et al., 2025b) | 83.1 | 98.3 | 95.1 | 98.6 | **99.8** | 81.5 | 97.4 | 96.5 | **98.3** | - |
| *VLA-based Methods* | | | | | | | | | | |
| DriveVLA-W0 (Li et al., 2025a) | **86.1** | **98.5** | **99.1** | 98.0 | 99.7 | 86.4 | **98.1** | 93.2 | 97.9 | 58.9 |
| ***Ours*** | | | | | | | | | | |
| SpanVLA (One-shot) | 79.4 | 96.4 | 89.4 | 97.8 | 99.8 | 87.4 | 95.8 | 94.5 | 98.2 | 81.6 |
| SpanVLA (Post-RFT) | **86.9** | 97.4 | 98.0 | 99.5 | **99.8** | 84.3 | 97.4 | 93.4 | **98.1** | 78.0 |
| SpanVLA* | **90.5** | 98.4 | 99.4 | 99.7 | **99.8** | 86.9 | 98.3 | 96.8 | **98.2** | 78.9 |

the state-of-the-art (SOTA) end-to-end driving models and VLA-based models. SpanVLA demonstrates strong performance in PDMS, and the RFT can significantly further improve the performance under the alignment with the reward signal. Moreover, as Table S2 illustrates, SpanVLA outperforms the SOTA methods on the `navtest` benchmark of NAVSIM v2. Table S3 reports the SOTA VLA driving performance on the `navhard` benchmark.

**nuReasoning-NR Benchmark.** To evaluate planning specifically in negative and recovery scenarios, nuReasoning-NR provides two diagnostic test sets, each containing 300 scenarios. Among the methods in this comparison, only SpanVLA and AutoVLA are additionally trained with NR samples; all other compared models use positive data only. As shown in Table 1, NR training improves both models over their positive-only counterparts on both test sets. The NR-trained variants also outperform all positive-only baselines on both sets, with SpanVLA achieving the highest PDMS on each subset and the highest average PDMS among the compared method

Table S3: Comparison with SOTA methods on the **NAVSIM v2** (`navhard`). * denotes best-of-N selection with N = 6, where the candidate with the highest PDMS is chosen.

| Methods | Stage | EPDMS↑ | NC↑ | DAC↑ | DDC↑ | TLC↑ | EP↑ | TTC↑ | LK↑ | HC↑ | EC↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Conventional End-to-end-based Methods* | | | | | | | | | | | |
| LTF (Chitta et al., 2022) | 1 | 23.1 | 96.2 | 79.6 | **99.1** | 99.6 | **84.1** | 95.1 | 94.2 | **97.6** | **79.1** |
| | 2 | 23.1 | 77.8 | 70.2 | 84.3 | 98.1 | 85.1 | 75.7 | 45.4 | **95.8** | **76.0** |
| RAP (Feng et al., 2025a) | 1 | 36.9 | 97.1 | **94.4** | 98.8 | 99.8 | 83.9 | 96.9 | **94.7** | 96.4 | 66.2 |
| | 2 | 36.9 | 83.2 | 83.9 | 87.4 | 98.0 | **86.9** | 80.4 | **52.3** | 95.2 | 52.4 |
| ***Ours*** | | | | | | | | | | | |
| SpanVLA (Post-RFT only Positive) | 1 | 40.1 | 98.4 | 94.3 | 97.8 | 99.9 | 85.7 | 97.2 | 94.2 | 97.6 | 72.1 |
| | 2 | 40.1 | 86.9 | 84.3 | 87.1 | 98.2 | 85.5 | 82.7 | 62.3 | 96.8 | 67.4 |
| SpanVLA (Post-RFT with nr) | 1 | **40.8** | **99.1** | 94.2 | **99.8** | **100.0** | 66.6 | **98.4** | 91.8 | 97.3 | 47.1 |
| | 2 | **40.8** | **89.8** | **84.5** | **94.8** | **98.6** | 63.4 | **88.2** | **53.1** | 95.2 | 50.8 |
| SpanVLA* | 1 | **47.9** | **99.8** | 99.1 | **99.9** | **100.0** | 71.4 | **99.3** | 97.8 | 97.3 | 49.8 |
| | 2 | **47.9** | **90.9** | **86.8** | **93.8** | **99.1** | 69.5 | **89.4** | **54.5** | 95.6 | 50.9 |

### F.2 Additional Ablation Studies for Action Bridging

Action bridging with flow matching serves a dual role: (i) extracting information from the VLM backbone via the KV cache, and (ii) predicting the flow-matching vector field. Accordingly, we instantiate three variants of Action Bridging with Flow Matching, differing in how bridging is formulated and where the vector field is predicted. (1) Action-space (SpanVLA). The bridging network directly predicts the vector field in the action space. We encode the history trajectory with an MLP to produce ten action-space queries, which are then used for vector-field estimation. (2) Latent-space. We use an MLP to encode the history and future trajectories into a latent feature, and perform vector-field prediction entirely in this latent space. (3) Sequential mode: We first use the bridging module to produce a conditional feature, and then run multi-step flow matching in the action space conditioned on this fixed feature. As Figure S3 illustrates, the action mode, which fuses and predicts the vector field within the action space, is better than the latent and sequential space.

### F.3 Additional Ablation Studies for RFT

#### F.3.1 Effect of Negative Penalty and Recovery Reward

We evaluate the negative-behavior penalty and recovery-behavior reward along three axes: the presence of these terms, their weights, and the proximity threshold that controls their activation. These experiments use positive, negative, and recovery training samples. Figure S4 summarizes the results.

**Presence of the Reward Terms.** We compare training with a PDMS-only reward against training with the negative-behavior penalty and recovery-behavior reward. Removing the two terms corresponds to setting $\lambda_N = \lambda_R = 0$. Including them improves PDMS from 87.6 to 90.8, showing a benefit from the distance-based reward terms beyond the driving reward in this setting.

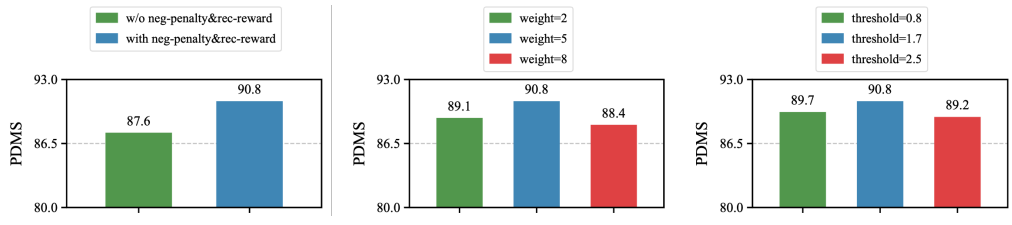



Figure S4: Ablation studies on negative-behavior penalty and recovery-behavior reward. **Left**: With vs. without the negative/recovery reward terms. **Middle**: Varying the weight of the negative/recovery terms. **Right**: Varying the proximity threshold of the negative/recovery terms.

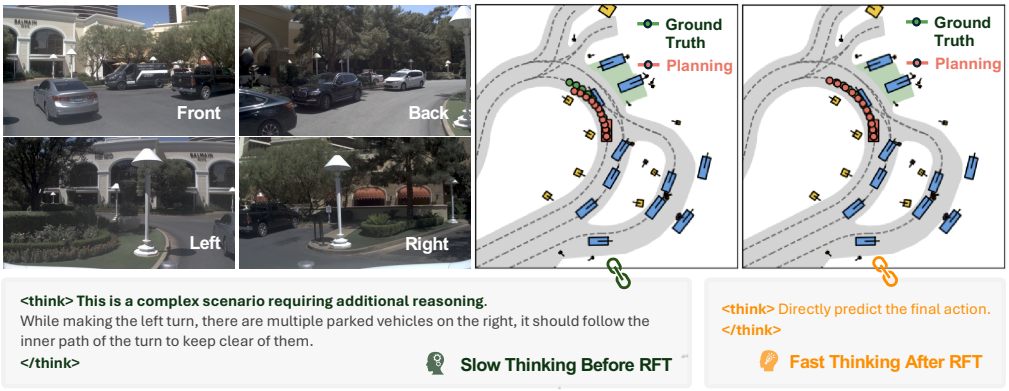



Figure S5: Additional comparison of planning and reasoning performance of SpanVLA model in **positive samples** of nuReasoning-NR dataset before and after RFT.

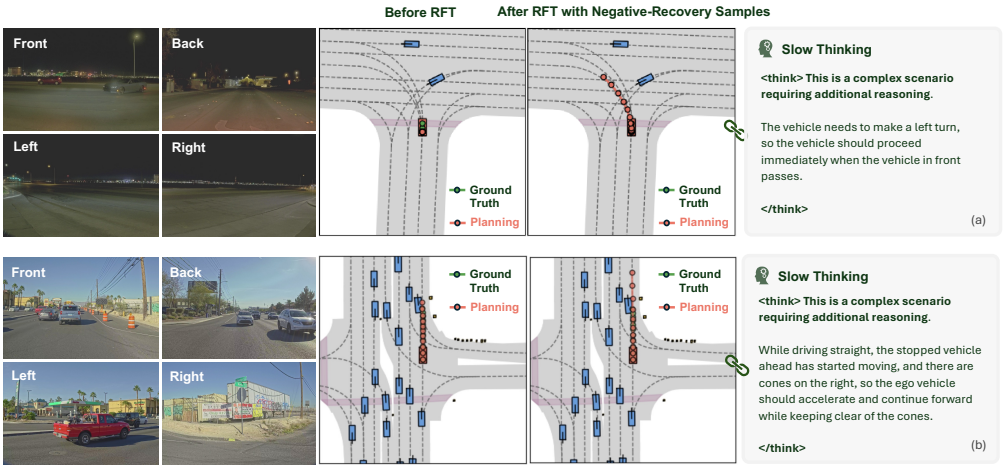



Figure S6: Additional comparison of planning and reasoning performance of SpanVLA model in **negative samples** of nuReasoning-NR dataset before and after RFT with negative-recovery samples.

**Reward-Weight Sensitivity.** The negative-behavior penalty and recovery-behavior reward are weighted by $\lambda_N$ and $\lambda_R$, respectively, following Eq. equation S4. The middle panel of Figure S4 compares different weight settings. The moderate setting achieves the highest PDMS among the tested configurations, while both smaller and larger settings yield lower scores.

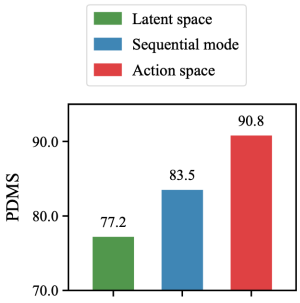



Figure S3: Comparison of three bridge modes for SpanVLA.

**Proximity-Threshold Sensitivity.** The threshold $\delta$ controls the range over which the negative-behavior penalty and recovery-behavior reward are active: the reference-matching term is nonzero when the average L2 distance to the corresponding reference trajectory is smaller than $\delta$. It also controls how the term decays with distance within this range. The right panel of Figure S4 compares $\delta \in \{0.8, 1.7, 2.5\}$, which yield PDMS values of 89.7, 90.8, and 89.2, respectively. The intermediate threshold performs best among the tested values.

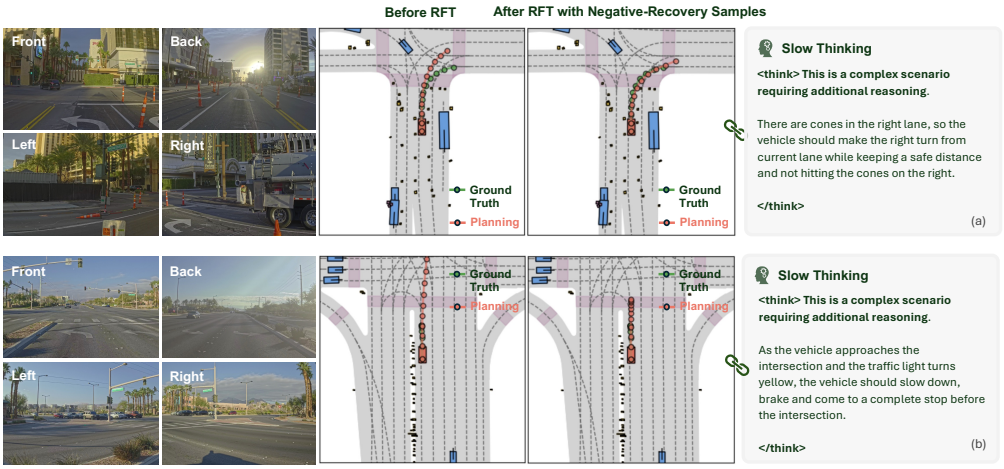



Figure S7: Additional comparison of planning and reasoning performance of SpanVLA model in **recovery samples** of nuReasoning-NR dataset before and after RFT with negative-recovery samples.

### F.4 Additional Qualitative Results

Figure S5 shows the results with positive samples, and the RFT can reduce the unnecessary reasoning in simple scenarios like (Zhou et al., 2025b). Figure S6 and Figure S7 present additional qualitative results for negative and recovery samples. Negative samples help the vehicle avoid overly conservative, yield-induced stopping behaviors during turning (Figure S6 (a)) or going straight (Figure S6 (b)), enabling it to proceed with left turns or accelerate to continue straight. Recovery samples further demonstrate how the vehicle can execute a right turn by temporarily using the only available left-turn lane when the right-turn lane is closed due to a construction zone, as shown in Figure S7 (a), and still brake to a stop when a yellow light transitions to red, as shown in Figure S7 (b).

## G Limitations and Future Work

Although our efficient action bridging significantly accelerates inference, the current runtime of 1.5 Hz remains insufficient for direct real-world deployment. Notably, we have not yet applied hardware-level optimization or deployment acceleration. For reference, Alpamayo (Wang et al., 2025) achieves 1.75 ms per token under optimized deployment, whereas our current implementation requires 33 ms per token. Such optimization strategies are orthogonal to our method and can be directly applied to our model. As future work, we plan to deploy the model on a real-world platform with system-level acceleration. In addition, while our experiments demonstrate that negative-recovery samples substantially improve policy performance and robustness, designing more effective reward functions to fully unlock the potential of these data remains an open problem and an important direction.