LaST-VLA: Thinking in Latent Spatio-Temporal Space for Vision-Language-Action in Autonomous Driving

Yuechen Luo[1,3*] **Fang Li**[2,3*] **Shaoqing Xu**[2,3,*,†] **Yang Ji**[3] **Zehan Zhang**[3,†]
Bing Wang[3] **Yuannan Shen**[3] **Jianwei Cui**[3] **Long Chen**[3]
Guang Chen[3] **Hangjun Ye**[3] **Zhi-xin Yang**[2✉] **Fuxi Wen**[1✉]
[1]Tsinghua University [2]University of Macau [3]Xiaomi EV
https://github.com/luo-yc17/LaST-VLA

Abstract

While Vision-Language-Action (VLA) models have revolutionized autonomous driving by unifying perception and planning, their reliance on explicit textual Chain-of-Thought (CoT) leads to semantic-perceptual decoupling and perceptual-symbolic conflicts. Recent shifts toward latent reasoning attempt to bypass these bottlenecks by thinking in continuous hidden space. However, without explicit intermediate constraints, standard latent CoT often operates as a physics-agnostic representation. To address this, we propose the **La**tent **S**patio-**T**emporal **VLA** (**LaST-VLA**), a framework that shifts the reasoning paradigm from discrete symbolic processing to a physically grounded Latent Spatio-Temporal CoT. By implementing a dual-feature alignment mechanism, we distill geometric constraints from 3D foundation models and dynamic foresight from world models directly into the latent space. Coupled with a progressive SFT training strategy that transitions from feature alignment to trajectory generation, and further refined via Reinforcement Learning with Group Relative Policy Optimization (GRPO) for safety and rule compliance, LaST-VLA achieves state-of-the-art performance across multiple autonomous driving benchmarks, including **NAVSIM v1**, **NAVSIM v2**, **Bench2Drive**, **SURDS**, and **NuDynamics**.

1 Introduction

The evolution of autonomous driving is undergoing a profound shift: from traditional modular pipelines [1, 2, 3] to Vision-Language-Action (VLA) models [4, 5, 6, 7]. While modular approaches offer interpretability, they often struggle with complex and long-tail scenarios due to rigid rule-based designs and error propagation across modules. In contrast, VLA models unify visual perception and driving policy into a holistic system, demonstrating superior scene generalization and natural language instruction following capabilities.

Despite progress, VLA models still face limitations in real-world deployment [8]. Early works [4, 9] generate trajectories without explicit intermediate reasoning, as shown in Fig. 1(a), suffering from poor interpretability. Recent efforts [10, 11] unlock strong reasoning capabilities through explicit Chain-of-Thought (CoT), thereby achieving interpretability and enabling the VLA to solve complex scenarios in a step-by-step manner, as shown in Fig. 1(b). However, projecting dense visual data into discrete text creates an inherent semantic gap, often leading to hallucinations inconsistent with

*Equal contribution. † Project leaders.
✉Corresponding authors.

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

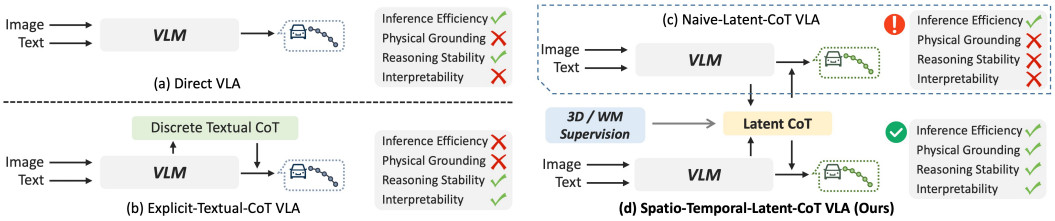


Figure 1: **Comparison of VLA Paradigms. (a) Direct VLA** is efficient but lacks reasoning. **(b) Explicit Textual CoT** is interpretable but suffers high latency and hallucinations. **(c) Naive Latent CoT (w/o supervision)** is efficient but unstable (model collapse). **(d) Our Spatio-Temporal Latent CoT (supervision)** aligns latent features with physical priors, achieving efficiency, stability, and grounding.

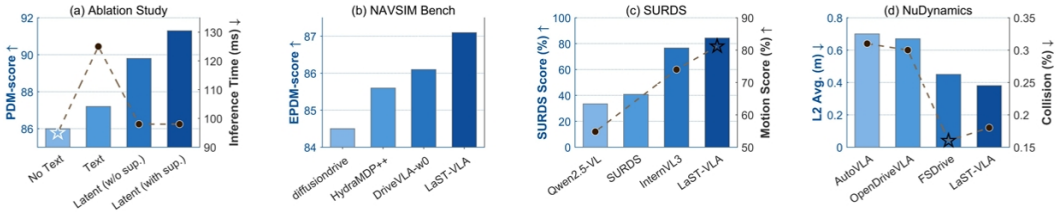


Figure 2: Performance comparison across multiple autonomous driving benchmarks.

visual input. Consequently, the planner often disregards visual evidence and instead follows flawed linguistic guidance, leading to potentially hazardous decision failures. Moreover, the generation of extensive intermediate sequences increases inference cost and results in redundant over-thinking, as discovered in AdaThinkDrive [12] and AutoVLA [13].

In response, recent methods [14, 15] argue that natural language may not always be the optimal medium for structured reasoning. They shift from explicit textual reasoning to implicit latent space reasoning, as shown in Fig. 1(c), achieving higher computational efficiency by bypassing the generation of lengthy textual chains. However, our empirical findings indicate that these methods rely on supervision from the final answer, without intermediate constraints. This absence of intermediate guidance leads to an unstable training process and renders the model susceptible to collapse.

To address these challenges, we propose LaST-VLA, a novel latent reasoning framework. Drawing on world-model priors [16, 17] and VLM methods with spatial forcing [18, 19], we leverage latent representations from geometry foundation models and video world models as supervision targets, as shown in Fig. 1(d). This design paradigm enables the model to receive temporally stable, step-level supervisory signals while embedding spatial-temporal perceptual capabilities into the latent reasoning process.

Our progressive training strategy first aligns latent features with spatial-temporal priors, then learns trajectory generation through supervised fine-tuning (Fig. 3, top right). We subsequently apply Reinforcement Learning (RL) using driving safety and comfort metrics to refine planning (Fig. 3, bottom right).

In summary, our key contributions are:

- We highlight two critical deficiencies in existing VLA-based autonomous driving: the disconnect between language and physical reality, and the safety hazards induced by excessive reliance on linguistic priors.
- We propose LaST-VLA, which unifies instruction following and dynamic prediction via a latent spatio-temporal CoT. This design surpasses both textual CoT and unsupervised latent approaches, overcoming the imprecision of the former and the training instability of the latter.
- We design a progressive training strategy that first equips the model with spatial-temporal understanding capabilities, and subsequently enables it to acquire planning ability. As shown in Fig. 2, experiments demonstrate that our method significantly outperforms **state-of-the-art** baselines across multiple autonomous driving benchmarks.

2 Related Work

2.1 VLA models in Autonomous Driving

Traditional E2E methods [3, 1] rely on modular pipelines but often lack natural interfaces for high-level intent. In contrast, VLAs unify these stages and demonstrate superior contextual understanding. Early research focused on scene understanding and meta-action generation [20, 21]. Some approaches employ VLMs directly for textual trajectory prediction. For instance, Orion [9] and OmniDrive [4] adopt StreamPETR [22] as Q-Former3D to compress scene features. This architecture bridges the visual reasoning space and facilitates subsequent textual trajectory prediction. Similarly, EMMA [10], trained on large-scale datasets, leverages Gemini [23] to predict discrete textual perception and planning outputs. Others integrate VLMs with E2E models in fast-slow systems: DriveVLM [24] uses VLMs for coarse trajectory prediction, which is refined by an E2E model. More recent methods introduce a textual cot prior to trajectory prediction, leveraging the common sense reasoning capabilities of LLM backbones to enhance planning performance, particularly in rare or complex scenarios [10, 6, 25, 26].

2.2 Latent Chain-of-Thought

Textual CoT has become a popular strategy for eliciting reasoning in VLMs and VLAs. However, recent studies [27] have shown that generated text can suppress visual evidence, causing the model to rely less on the original visual input. Other works [28, 29] further indicate that excessive or uninformative CoT may reduce both inference efficiency and prediction accuracy. Related LLM work improves reasoning efficiency [30, 31] and trains tool-using coding agents [32]. In autonomous driving, AdaThinkDrive [12] and AutoVLA [13] also show that explicit textual CoT can hinder planning performance in certain scenarios. To address these limitations, several works [33, 14] have begun to explore latent reasoning in LLMs, where intermediate computation occurs in continuous hidden space rather than textual space. This paradigm enables more efficient reasoning by avoiding the cost and information loss of verbose textual rationales. Building on this direction, recent methods [34, 35] extend latent reasoning to VLMs and achieve latent spatial reasoning. More recently, 3DThinker [36] introduces geometric supervision by aligning 3D special-token hidden states with a 3D foundation model, demonstrating the value of physically grounded intermediate representations. In contrast, LaST-VLAfollows a fully continuous latent reasoning paradigm: the hidden feature at each Latent CoT position is propagated directly as the next-step embedding, rather than mediated by decoded discrete special tokens. This design better preserves high-dimensional spatio-temporal information and models the future dynamic states required for trajectory planning. Nevertheless, we observe that leaving continuous latent CoT without physical guidance is insufficient for robust autonomous driving. To address this, we propose **Think with Latent Spatio-Temporal**, which leverages geometric and dynamic priors to supervise the latent CoT, thereby improving the fidelity and robustness of downstream trajectory planning.

3 Method

In this section, we present the LaST-VLA framework, as illustrated in Fig. 3. This architecture bridges the gap between perception and planning via a Latent Spatio-Temporal CoT mechanism. The overall pipeline comprises three core components: (1) Latent Spatio-Temporal CoT, (2) Progressive Two-Stage SFT Strategy, and (3) Latent-Grounded Trajectory Refinement via GRPO.

3.1 Preliminaries

In this work, we formulate end-to-end autonomous driving planning as a conditional generation problem augmented by latent reasoning.

Inputs. At any given time step t, the system receives a multimodal query tuple $\mathbf{Q}_t = (I_t, L)$, comprising the front-view camera image $I_t \in \mathbb{R}^{H\times W\times 3}$ and the text input L. Specifically, L includes the navigation instruction l (e.g., *"Turn left"*), the ego-vehicle state vector $s_{\text{ego}} \in \mathbb{R}^{d_s}$ (velocity, acceleration), and historical trajectory $h_{\text{his}} \in \mathbb{R}^{T\times 3}$.

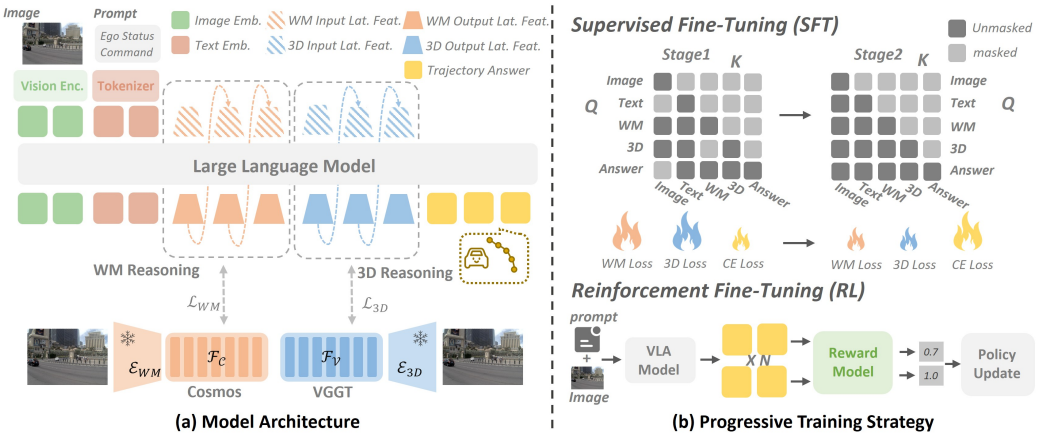


Figure 3: **Overview of the LaST-VLA framework.** (a) **Model Architecture:** The model constructs a continuous Latent CoT by recurrently propagating hidden states in the latent space, while aligning them with dynamic and geometric priors distilled from foundation models (Cosmos and VGGT) via adapters. (b) **Progressive Training Strategy:** The pipeline uses a two-stage SFT phase with structured causal masking to enforce physically grounded latent reasoning, followed by RL fine-tuning to optimize the policy for driving safety and compliance.

Latent Reasoning. Deviating from standard VLA paradigms that map inputs directly to actions, we introduce a *Latent Spatio-Temporal CoT*, denoted as H, as a continuous reasoning bridge. Derived from the final hidden states of the VLM backbone, we explicitly structure H into two decoupled features: $H = \{H^{\text{dyn}}, H^{\text{geo}}\}$. Here, H^{dyn} captures temporal dynamics evolution (WM), while H^{geo} encodes spatial geometry (3D). These continuous latent variables are autoregressively generated to explicitly ground the reasoning process in physical properties.

Probabilistic Modeling. Our objective is to learn a hierarchical model that generates the trajectory $\mathbf{a}_t$ conditioned on the latent CoT H. We formalize this by decoupling the joint distribution into a thinking phase and a planning phase:

$$P_\theta(\mathbf{a}_t, \mid \mathbf{Q}_t) = \underbrace{P(H^{\text{dyn}}, H^{\text{geo}} \mid \mathbf{Q}_t)}_{\text{Thinking Process}} \cdot \underbrace{P(\mathbf{a}_t \mid H^{\text{dyn}}, H^{\text{geo}}, \mathbf{Q}_t)}_{\text{Planning Process}}. \tag{1}$$

In this formulation, the model first acts as a *Thinker* to generate continuous reasoning states H, then functions as a *Planner* to predict trajectory waypoints $\mathbf{a}_t$ conditioned on these grounded thoughts. Notably, both probability distributions are parameterized by the same VLA, realized through a unified autoregressive process.

3.2 Latent Spatio-Temporal CoT

To teach the model to reason with 3D geometry and world dynamics, conventional approaches typically rely on explicit reconstruction targets, such as predicting dense depth maps or future video frames. However, these methods suffer from high computational overhead and information redundancy, as they compel the model to focus on irrelevant textural details rather than critical physical states. Instead, as depicted in Fig. 3 (a), we introduce external foundation models as feature-level teachers during training, distilling their structured knowledge into the continuous latent tokens generated from the VLM's reasoning process. This strategy facilitates effective physical understanding without the burden of pixel-level generation.

Specifically, the reasoning process begins with visual encoding. Given the input image I_t, vision encoder $\mathcal{V}$ is employed to extract patch-level visual embeddings $E_{\text{img}} = \mathcal{V}(I_t) \in \mathbb{R}^{N \times D}$. These visual features are concatenated with the embeddings of linguistic instructions E_L to form the multimodal input sequence. The VLM π_θ autoregressively generates the hidden states. Let h_k denote the output of the last layer at step k. The latent CoT sequence H is derived as:

$$H = \{h_k\}_{k=1}^{K} = \pi_\theta(E_{\text{img}}, E_{\text{L}}), \tag{2}$$

where K is the length of the reasoning chain. This sequence is strictly partitioned into dynamic (H^{dyn}) and geometric (H^{geo}) streams to facilitate two specialized physical alignments with adapters.

To prevent the adapter from shortcutting alignment by attending solely to raw pixel patterns, we apply a random binary mask M to the visual embeddings, yielding $\tilde{E}_{\text{img}} = E_{\text{img}} \odot M$. These masked features and continuous CoT hidden states are as input to adapters for alignment with different priors.

To ground the implicit reasoning process in physical reality, we introduce dynamic and geometric feature alignment mechanisms. Specifically for temporal dynamics, a transformation function Φ_{dyn} projects the sequential hidden states H^{dyn} into the representation space of a video world model (Cosmos [37]). Unlike the static nature of standard linguistic tokens, this dynamic manifold inherently encodes temporal evolution, enabling the model to effectively capture the future motion priors of traffic participants and continuous environmental changes. Concurrently for spatial geometry, another transformation function Φ_{geo} fuses the hidden states H^{geo} with the original visual embeddings $\tilde{E}_{\text{img}}$ to align with the dense feature space of a 3D foundation model (VGGT [38]). This allows the model to recover metrically accurate spatial priors, such as scene depth and occupancy structures, directly within the latent space. The overall transformation process is formally defined as:

$$\mathbf{p}^{\text{dyn}} = \Phi_{\text{dyn}}(H^{\text{dyn}}, \tilde{E}_{\text{img}}), \quad \mathbf{p}^{\text{geo}} = \Phi_{\text{geo}}(H^{\text{geo}}, \tilde{E}_{\text{img}}). \tag{3}$$

Ultimately, the latent CoT serves as an implicit reasoning bridge that encodes what the world looks like in 3D space and how it will evolve, while remaining fully differentiable and intrinsically aligned with the downstream trajectory output space.

Latent-Conditioned Planning. Finally, the grounded latent chain-of-thought is integrated in the trajectory generation process. Following the latent reasoning phase, our policy autoregressively predicts the future waypoints $\mathbf{a}_t$ (represented as textual tokens) conditioned on these physical reasonings:

$$\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t \mid H^{\text{dyn}}, H^{\text{geo}}, \mathbf{Q}_t). \tag{4}$$

This ensures that the final planning decision is explicitly driven by the spatio-temporal understanding distilled in the latent space.

3.3 Progressive Two-Stage SFT Strategy

As illustrated in Fig. 3 (b), our method introduces external foundation models as feature teachers during training, and distills their structured knowledge into the continuous latent tokens generated in the VLM's reasoning process. To implement this effectively, we propose a *Progressive Two-Stage Supervised Fine-Tuning strategy* that decouples the complex objective into learning to *think* and learning to *action*.

Unified Optimization Objective. We first formulate the unified training objective for both stages. To ground the latents in physical reality, we employ feature distillation from frozen foundation models, denoting F_{Cosmos} as the dynamic feature derived from the Cosmos and F_{VGGT} as the geometric features from the VGGT aggregator. For dynamic representation, we designate three sets of latent tokens to align with the temporal features of F_{Cosmos}. These tokens encode dynamics across progressive temporal scales, capturing short-term, medium-term, and long-term motion. Simultaneously, we allocate a distinct set of latent tokens to align with F_{VGGT} to capture static geometric constraints. The alignment losses are calculated as the Mean Squared Error (MSE) between the adapter-projected features $\mathbf{p}$ and these teacher targets:

$$\mathcal{L}_{\text{WM}} = \|\mathbf{p}^{\text{dyn}} - F_{\text{Cosmos}}\|_2^2, \mathcal{L}_{\text{3D}} = \|\mathbf{p}^{\text{geo}} - F_{\text{VGGT}}\|_2^2, \tag{5}$$

The total SFT loss combines the trajectory prediction error with these physical alignment terms:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{action}}\mathcal{L}_{\text{CE}} + \lambda_{\text{WM}}\mathcal{L}_{\text{WM}} + \lambda_{\text{3D}}\mathcal{L}_{\text{3D}}. \tag{6}$$

where $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss for action generation, and λ are the balancing coefficients adjusted dynamically across stages.

Phase I: Physics-Aware Alignment. In this initial phase, we prioritize learning physical knowledge over trajectory generation. We set the loss weights to $\lambda_{\text{WM}} = \lambda_{\text{3D}} = 1.0 \gg \lambda_{\text{action}} = 0.01$. This forces the latent CoT H to strictly align with the teacher models' geometric and dynamic representations. To ensure the planner relies on this reasoning, we apply a **Structured Causal Masking** (Fig. 3(b)): (1) **Latent Mutual Masking:** We mask the 3D and WM tokens from each other so they learn independently. (2) **Visual Bottleneck Masking:** We prevent the action tokens from

attending to the raw image embeddings E_{img}. This forces the model to compress all necessary visual information into H, making the latent thoughts the only information bridge for decision-making.

Phase II: Latent-Grounded Planning. Once the reasoning capability is established, we switch to the second phase to refine the driving policy. We invert the weights to $\lambda_{\text{action}} = 1.0 \gg \lambda_{\text{WM}} = \lambda_{\text{3D}} = 0.01$, prioritizing accurate trajectory prediction. In this stage, we allow action tokens to attend to both the latent CoT H and the raw image embeddings E_{img}. This allows the model to combine two types of information: the high-level physical understanding from H and the fine-grained visual details from the original image. The reduced alignment weights maintain consistent reasoning, while the planner learns to use both signals for robust driving.

3.4 Latent-Grounded Trajectory Refinement via GRPO

After the progressive SFT phase, our policy has acquired robust spatio-temporal capabilities, with the geometric and dynamic latent CoT grounding the reasoning process. To further elevate the policy's execution capability, we freeze the dynamics and geometry adapters. We employ Group Relative Policy Optimization (GRPO) [39] to optimize the VLA's action generation by maximizing trajectory-level rewards, using established latent reasoning as stable internal guidance.

Reward Formulation. To incentivize the model to generate safe, compliant, and precise driving behaviors, we design a composite reward function R comprising three distinct components. **PDMS Reward** (R_{traj}) evaluates the overall quality of the predicted trajectory using the Predictive Driver Model Score [40], normalized to a continuous value in $[0, 1]$. **Format Reward** (R_{fmt}) is a discrete indicator that strictly penalizes failures to adhere to the required output structure. **Goal Reward** (R_{goal}) encourages endpoint precision by assigning tiered incentives based on the L_1 distance between the predicted and ground-truth endpoints. Calculation details are provided in the Appendix D. The total reward for a trajectory is integrated as:

$$R = \lambda_{traj} R_{\text{traj}} + \lambda_{fmt} R_{\text{fmt}} + \lambda_{goal} R_{\text{goal}}. \tag{7}$$

Optimization Objective. We employ GRPO to optimize the policy π_θ. For each input query q, we sample a group of G candidate outputs $\{o_1, o_2, \ldots, o_G\}$ from the sampling policy $\pi_{\theta_{old}}$. The optimization process leverages the relative advantage of these outputs to update the policy, incorporating a clipped objective to ensure training stability and a KL-divergence term to prevent excessive deviation from the reference policy π_{ref}. The objective function is formulated as follows:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim D, \{o_i\}\sim\pi_{\theta_{old}}} \left[\frac{1}{G} \sum_{i=1}^{G} \mathcal{J}_i - \beta \mathbb{D}_{KL}(\pi_\theta || \pi_{\text{ref}}) \right], \tag{8}$$

$$\mathcal{J}_i = \min\left(c_i A_i, \text{clip}\left(c_i, 1-\epsilon, 1+\epsilon\right) A_i\right). \tag{9}$$

Here, $c_i = \pi_\theta(o_i|q)/\pi_{\theta_{old}}(o_i|q)$, ϵ and β are hyperparameters controlling the clipping range and regularization strength, respectively. Advantage A_i is computed by standardizing the rewards within the group: $A_i = (R_i - \text{mean}(R))/\text{std}(R)$, where R_i is the reward for output.

4 Experiment

4.1 Implementation details

Datasets. We conduct evaluations across multiple autonomous driving benchmarks. Specifically, we use **NAVSIM** [40], a planning-oriented benchmark derived from OpenScene, and curate a 24k challenging subset from its standard 85k split, denoted as navtrain-hard-24k, for efficient training. We also use **SURDS** [41], built on nuScenes, to evaluate 3D spatial reasoning, and construct **NuDynamics** following SURDS to assess dynamic scene understanding. In addition, we evaluate open-loop driving performance on **nuScenes** [42] and closed-loop interactive driving performance on **Bench2Drive** [43]. All details of the dataset are in Appendix C.

Metrics. We evaluate LaST-VLAacross two primary dimensions: trajectory evaluation and spatial-dynamic reasoning. For trajectory evaluation, we consider closed-loop planning on **NAVSIMv1** [40] and **NAVSIMv2** [44], as well as interactive closed-loop driving on **Bench2Drive** [43]. Specifically, we report PDMS on NAVSIMv1, EPDMS on NAVSIMv2, and Driving Score (DS) and Success

Table 1: Comparison with state-of-the-art methods on NAVSIMv1 with PDMS. * indicates models utilizing Textual CoT.

Method	Venue	Image	Lidar	NC↑	DAC↑	TTC↑	CF↑	EP↑	PDMS↑
Constant Velocity	-			68.0	57.8	50.0	**100**	19.4	20.6
Ego Status MLP	-			93.0	77.3	83.6	**100**	62.8	65.6
UniAD [1]	CVPR 2023	✓		97.8	91.9	92.9	**100**	78.8	83.4
DiffusionDrive [46]	CVPR 2025	✓	✓	98.2	96.2	94.7	**100**	82.2	88.1
WoTE [47]	ICCV 2025	✓	✓	98.5	96.8	94.9	99.9	81.9	88.3
DriveVLA-W0-7B [48]	ICLR 2026	✓		**98.7**	**99.1**	94.9	99.3	83.3	90.2
AdaThinkDrive-8B* [12]	ICRA 2026	✓		98.4	97.8	95.2	**100**	84.4	90.3
Recogdrive-2B [5]	ICLR 2026	✓		97.9	97.3	94.9	**100**	**87.3**	90.8
InternVL3-8B (SFT)*	-	✓		98.3	92.3	94.7	**100**	77.2	83.9
InternVL3-8B (RL)*	-	✓		98.3	94.0	94.7	**100**	83.0	87.2
LaST-VLA-2B (SFT)	-	✓		98.5	95.2	**95.6**	**100**	80.4	87.0
LaST-VLA-2B (RL)	-	✓		98.6	97.7	**95.6**	**100**	86.5	91.1
LaST-VLA-8B (SFT)	-	✓		**98.7**	95.4	**95.6**	**100**	80.5	87.3
LaST-VLA-8B (RL)	-	✓		**98.7**	97.9	**95.6**	**100**	86.8	**91.3**

Table 2: Comparison with state-of-the-art methods on NAVSIMv2 with EPDMS.

Method	NC ↑	DAC ↑	DDC ↑	TLC ↑	EP ↑	TTC ↑	LK ↑	HC ↑	EC ↑	EPDMS ↑
Ego Status	93.1	77.9	92.7	99.6	86.0	91.5	89.4	**98.3**	85.4	64.0
TransFuser [49]	96.9	89.9	97.8	99.7	87.1	95.4	92.7	**98.3**	87.2	76.7
DiffusionDrive [46]	98.2	95.9	99.4	**99.8**	87.5	97.3	**96.8**	**98.3**	**87.7**	84.5
HydraMDP++ [50]	98.5	98.5	**99.5**	99.7	87.4	97.9	95.8	98.2	75.7	85.6
DriveVLA-W0-7B [48]	98.5	**99.1**	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
LaST-VLA-2B (RL)	**98.6**	97.7	99.1	99.7	90.2	**98.2**	96.6	**98.3**	86.8	86.8
LaST-VLA-8B (RL)	**98.7**	97.9	99.2	99.7	**90.3**	**98.2**	96.6	**98.3**	86.3	**87.1**

Rate (SR) on Bench2Drive. Metric definitions are summarized in Appendix D. For spatial-dynamic reasoning, we evaluate 3D spatial understanding on SURDS using Yaw, Pixel, Depth, Dis, L/R, and F/B accuracy, and assess dynamic understanding on NuDynamics with the Motion State Estimation (Motion) metric.

Training Details. We utilize InternVL3-2B and InternVL3-8B [45] as the foundation model, training across two sequential phases: SFT and RL. In the first SFT stage, we fine-tune on navtrain-hard-24k for 2 epochs (learning rate 1×10^{-5}, batch size 2). In the second stage, we train on the full navtrain dataset for 2 epochs (learning rate 1×10^{-5}, batch size 2). Finally, during the RL phase, we employ 32 NVIDIA H20 GPUs to train for 3 epochs (learning rate 2×10^{-6}, batch size 2). For SURDS, we only perform SFT for 2 epochs (learning rate 1×10^{-5}, batch size 2). Additional details and hyperparameters are provided in the Appendix D.

4.2 Performance Comparison

NAVSIM Benchmark. Table 1 presents the performance comparison on the NAVSIMv1 benchmark. LaST-VLA-8B achieves a PDMS of 91.3, setting a new state-of-the-art (SOTA). This result surpasses the previous best vision-only method (Recogdrive-2B) by 0.5 PDMS. Remarkably, even our lightweight Last-VLA-2B delivers exceptional performance with 91.1 PDMS, outperforming Recogdrive-2B despite similar parameters. Furthermore, LaST-VLA-8B outperforms VLM baselines, with its SFT and RL variants surpassing InternVL3-8B baseline by 3.4 and 4.1 PDMS, respectively. Notably, improvements in No at-Fault Collision (NC) and Drivable Area Compliance (DAC) are attributed to the enhanced 3D spatial awareness, while superior Time-to-Collision (TTC) and Ego Progress (EP) performance validates the dynamic foresight capabilities distilled from the world model. On NAVSIMv2 (Table 2), LaST-VLA maintains its superiority, achieving a new SOTA with 87.1 EPDMS, and outperforming the previous best method, DriveVLA-W0-7B, by 1.0 EPDMS. Remarkably, even the lightweight 2B variant delivers exceptional performance with 86.8 EPDMS, surpassing the prior SOTA. These results demonstrate that, compared to standard VLMs, our method significantly enhances comprehensive driving capabilities, particularly in complex scenarios that require 3D reasoning and future prediction. Moreover, consistent excellence across both bench-

Table 3: **Comparison with state-of-the-art methods on the SURDS and NuDynamics benchmarks. Score** represents the average performance on SURDS. **Motion** denotes the Motion State Estimation accuracy on NuDynamics. * indicates models trained with SFT+RL, and † indicates models trained with Only SFT. All methods except LaST-VLA utilize Textual CoT.

Method	Single-object(%)↑			Multi-object(%)↑			Score(%)↑	Motion(%)↑
	Yaw	Pixel	Depth	Dis	L/R	F/B		
Qwen2.5-VL-72B-Instruct	11.57	6.13	44.00	58.05	66.16	14.92	33.47	54.79
Qwen2.5-VL-7B-Instruct	7.57	3.46	25.95	11.46	17.95	9.30	12.61	48.77
Qwen2.5-VL-3B-Instruct	6.27	3.81	27.68	17.84	14.81	10.49	13.48	35.34
SURDS-3B* [41]	20.97	44.81	69.84	49.30	51.35	8.54	40.80	-
InternVL3-2B†	37.30	48.90	99.90	80.54	82.81	77.95	71.23	66.54
InternVL3-8B†	45.19	61.83	**100.00**	82.05	86.16	84.43	76.61	73.97
LaST-VLA-2B†	60.50	54.10	100.00	84.32	89.24	88.22	79.40	71.80
LaST-VLA-8B†	**70.16**	**71.28**	**100.00**	**86.05**	**90.27**	**88.00**	**84.29**	**81.19**

marks confirms that our approach is not merely overfitting to PDMS; rather, it demonstrates strong generalization by performing robustly on the more holistic and challenging EPDMS.

SURDS and NuDynamics Benchmark. Table 3 presents the comparative evaluation on both the SURDS and NuDynamics benchmarks. With only SFT, LaST-VLA-8B and LaST-VLA-2B consistently outperform their respective InternVL3 baselines. On the SURDS benchmark, LaST-VLA-8B achieves improvements of 43.49% and 7.68% over the SURDS-3B method and InternVL3-8B baseline, respectively, while LaST-VLA-2B delivers an 8.17% improvement over InternVL3-2B. Specifically, LaST-VLA-8B dominates intrinsic geometric reasoning, with Yaw Determination and Pixel Estimation reaching 70.16% and 71.28%, demonstrating precise absolute localization capabilities. Furthermore, exceptional performance on relational tasks, such as Left/Right (90.27%) and Front/Behind (88.00%), highlights its robust spatial compositional reasoning, mitigating the spatial disorientation often observed in prior VLMs. On NuDynamics, LaST-VLA-8B and LaST-VLA-2B show exceptional dynamic scene understanding with Motion scores of 81.19% and 71.80%, respectively, outperforming both their fine-tuned baselines and the general Qwen2.5-VL-72B [51]. This validates that aligning with world models endows the planner with the capability to accurately perceive and predict the motion states of dynamic agents across different parameter scales.

Bench2Drive Benchmark. To further evaluate the generalization ability of LaST-VLA under closed-loop autonomous driving scenarios, we conduct experiments on the Bench2Drive benchmark, which provides a comprehensive evaluation protocol for planning-oriented driving models. The benchmark reports multiple complementary metrics, including Driving Score (DS), Success Rate (SR), Efficiency, and Comfort, thereby assessing not only task completion but also driving quality and execution stability. As shown in Table 5, LaST-VLA achieves strong performance at both the 2B and 8B parameter scales. In particular, LaST-VLA-2B already outperforms Recogdrive-8B in DS and SR, indicating that the proposed latent reasoning design can improve planning performance without simply relying on larger model capacity. When scaled to 8B parameters, LaST-VLA further achieves the best DS and SR among the compared methods, surpassing recent state-of-the-art models such as Recogdrive and Orion. Although some baselines obtain higher scores on individual secondary metrics such as Efficiency or Comfort, LaST-VLA delivers the strongest overall driving performance, demonstrating the effectiveness of latent spatio-temporal reasoning for robust decision-making in complex driving environments.

nuScenes Benchmark. Table 4 presents the open-loop planning evaluation on the nuScenes dataset. LaST-VLA-8B achieves SOTA performance in trajectory precision, recording the lowest average L2 error of 0.38m among all compared VLM-based methods. Beyond superior accuracy, the model maintains an exceptionally low collision rate of 0.18%, demonstrating that our latent reasoning paradigm effectively balances precise geometric adherence with safety. Furthermore, the lightweight LaST-VLA-2B variant also delivers robust performance, validating the effectiveness of our approach across different model sizes.

Table 4: Comparison with state-of-the-art methods on the nuScenes. * indicates that the ego status is use.

Method	Reason	VLM Model	L2 (m) ↓				Collision (%) ↓			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
Conventional End-to-end Methods										
UniAD* [1]	-	-	0.20	0.42	0.75	0.46	0.02	0.25	0.84	0.37
VLMs-based Methods										
Doe-1 [52]	-	Lumina-mGPT-7B	0.50	1.18	2.11	1.26	0.04	0.37	1.19	0.53
AutoVLA* [13]	Text CoT	Qwen2.5-VL-3B	0.28	0.66	1.16	0.70	0.14	0.25	0.53	0.31
RDA-Driver* [53]	-	LLaVA-7B	0.23	0.73	1.54	0.80	0.00	0.13	0.83	0.32
OpenDriveVLA* [54]	-	Qwen2.5VL-3B	0.19	0.58	1.24	0.67	0.02	0.18	0.70	0.30
FSDrive* [55]	Visual CoT	Qwen2-VL-2B	0.18	0.39	0.77	0.45	**0.00**	**0.06**	**0.42**	**0.16**
LaST-VLA* (Ours)	Latent CoT	InternVL3-2B	0.19	0.35	0.71	0.42	0.03	0.16	0.47	0.22
LaST-VLA* (Ours)	Latent CoT	InternVL3-8B	**0.17**	**0.33**	**0.64**	**0.38**	**0.00**	0.11	**0.42**	0.18

Table 5: Comparison with state-of-the-art methods on Bench2Drive.

Method	DS↑	SR↑	Efficiency↑	Comfort↑
UniAD-Base [1]	45.81	16.36	129.21	**43.58**
DriveAdapter [56]	64.22	33.08	70.22	16.01
ReasonPlan [57]	64.01	34.55	**180.64**	25.63
Recogdrive-8B [5]	71.36	45.45	138.18	17.45
Orion-7B [9]	77.74	54.62	151.48	17.38
InternVL-8B (RL)	65.34	34.09	105.42	16.37
LaST-VLA-2B (RL)	73.17	49.55	142.73	17.89
LaST-VLA-8B (RL)	**78.37**	**55.45**	149.47	18.71

Table 6: Ablation on geometric (3D) and dynamic (WM) latent CoT.

Mode	WM	3D	NC↑	DAC↑	TTC↑	CF↑	EP↑	PDMS↑
SFT			98.3	92.3	94.7	100	77.2	83.9
	✓		98.6	94.4	95.5	100	78.4	85.8
		✓	98.6	94.8	**95.7**	100	79.1	86.4
	✓	✓	**98.7**	**95.4**	**95.6**	**100**	**80.5**	**87.3**
RL			98.3	94.0	94.7	100	83.0	87.2
	✓		98.5	97.7	95.2	100	86.7	90.9
		✓	98.6	97.7	95.6	100	86.4	91.0
	✓	✓	**98.7**	**97.9**	**95.6**	**100**	**86.8**	**91.3**

4.3 Ablation Studies

On Latent Spatio-temporal Reasoning. Table 6 validates the impact of our spatiotemporal reasoning components. Explicitly incorporating both geometric (3D) and dynamic (WM) pathways yields consistent gains across training phases. During SFT, the full model achieves a 3.4 PDMS increase over the baseline, driven by a substantial rise in DAC. This confirms that the geometric latent CoT grounds the planner in physical reality and ensures precise adherence to boundaries. The RL phase further amplifies this advantage, widening the performance gap to a 4.1 PDMS improvement. Notably, the superior TTC and EP metrics highlight the role of the dynamic latent CoT: by distilling foresight from the World Model, the agent anticipates evolving dynamics for safer and more efficient planning.

On Reasoning. Table 7 investigates reasoning modalities. Compared to the baseline without textual CoT, introducing explicit textual CoT improves PDMS by 1.2, suggesting symbolic planning aids task decomposition but is limited by the semantic gap. Employing the latent CoT without supervision achieves 89.8 PDMS, effectively bypassing the textual bottleneck.

However, this unsupervised approach functions as an ungrounded "black box" lacking physical constraints. This manifests as training instability: as visualized in Fig. 4, the unsupervised variant suffers from marked performance oscillations even at later stages (orange line). In contrast, explicit physical supervision on latent tokens yields optimal performance, surpassing the unsupervised variant by 1.5 PDMS with robust convergence stability (red line). Notably, this physical grounding translates into critical safety gains, boosting DAC to 97.9 and TTC to 95.6. These results demonstrate that aligning latent CoT with physical priors transforms unstable features into a robust, grounded reasoning engine.

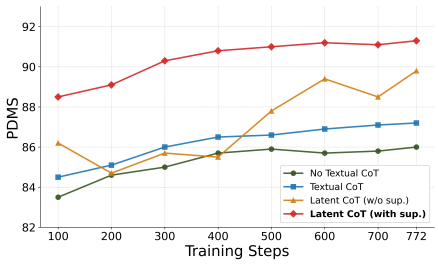


Figure 4: PDMS performance varying with training steps during the RL phase.

On Structured Causal Mask. Table 8 examines the impact of the proposed structured causal mask. Compared to the standard masking baseline, our design increases PDMS by 0.9. This quantitative gain

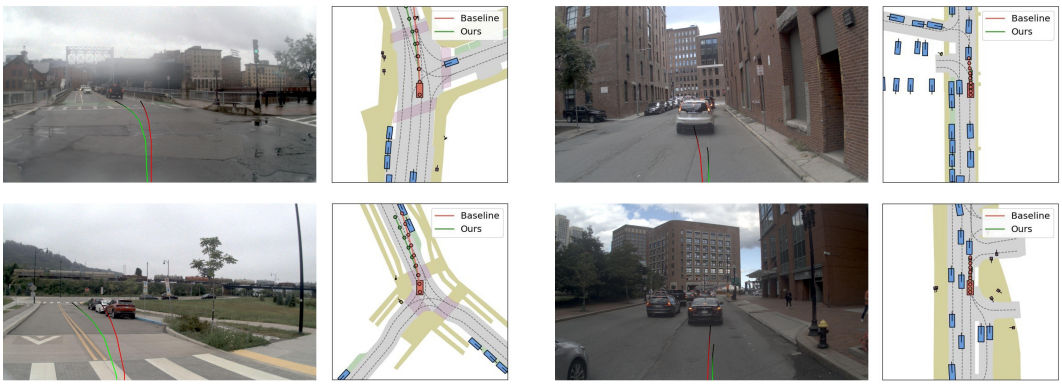


Figure 5: Qualitative visualization comparing the **Textual CoT baseline (Red)** and **LaST-VLA (Green)**. (a) **Drivable Area Compliance (DAC):** Our method maintains precise lane adherence, whereas the baseline violates spatial boundaries. (b) **Time-to-Collision (TTC):** Our method accurately anticipates dynamics to avoid rear-end collisions, while the baseline fails to brake effectively.

Table 7: Ablation study of textual and latent CoT after RL training.

Method	NC↑	DAC↑	TTC↑	CF↑	EP↑	PDMS↑
No Text	98.5	94.3	95.5	100	79.6	86.0
Text	98.3	94.0	94.7	100	83.0	87.2
Latent (w/o sup.)	98.6	96.8	95.8	100	84.6	89.8
Latent (with sup.)	**98.7**	**97.9**	**95.6**	**100**	**86.8**	**91.3**

Table 8: Ablation study on the effectiveness of the Structured Causal Mask after RL training phase.

Mode	NC↑	DAC↑	TTC↑	CF↑	EP↑	PDMS↑
Standard Mask	98.1	97.3	94.6	100	86.6	90.4
Structured Causal Mask	**98.7**	**97.9**	**95.6**	**100**	**86.7**	**91.3**

shows that blocking the direct visual shortcut effectively compels the planner to rely on high-level latent spatio-temporal reasoning, thereby improving both trajectory compliance and safety.

4.4 Qualitative Results

To visualize the efficacy of our approach, Fig. 5 presents a qualitative comparison between the textual CoT baseline and LaST-VLA. As illustrated in Fig. 5(a), LaST-VLA (Green) generates smooth, compliant trajectories that strictly adhere to lane boundaries, whereas the baseline (Red) struggles with metric precision, occasionally deviating from the drivable area. This validates that our geometric latent CoT, explicitly aligned with 3D foundation models, effectively grounds the planner with accurate spatial constraints. Furthermore, regarding dynamic safety in Fig. 5(b), while the baseline suffers from limited foresight and fails to decelerate, LaST-VLA accurately predicts future dynamics and executes timely braking to ensure safety in critical interaction scenarios. This demonstrates the significance of the dynamic latent CoT in distilling temporal foresight from the world model, equipping the planner with predictive capabilities typically lacking in pure textual reasoning. More results are available in Appendix B.

5 Conclusion

In this work, we introduced **LaST-VLA**, a framework that shifts the reasoning paradigm of autonomous driving from explicit text to a continuous Latent Spatio-Temporal Space. By distilling physical priors from 3D and video foundation models, our approach effectively addresses the challenges of inference latency, semantic hallucinations, and the lack of physical grounding inherent to textual CoT. Coupled with a progressive training strategy and GRPO, LaST-VLA achieves SOTA performance on NAVSIM, SURDS and NuDynamics benchmarks. These results demonstrate that aligning latent thinking with physical reality significantly enhances the robustness, efficiency, and safety of VLA-based planning.

Acknowledgements and Disclosures

This work was partially supported by the National Key R&D Program of China under Grant 2024YFB2505803, and in part by the National Natural Science Foundation of China under Grant 62461160260, in part by the Science and Technology Development Fund, Macau SAR (Grant no. 0092/2024/AFJ, 0212/2025/AMJ, 001/2024/SKL), in part by the University of Macau (Grant No.: MYRG-GRG2024-00299-FST-UMDF, MYRG-GRG2025-00312-FST), in part by the Hainan Department of Science and Technology (Grant no. GHYF2026042), in part by the Guangdong Department of Science and Technology (Grant no. EF2023-00243-IOTSC). The authors declare no competing interests.

References

[1] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[2] Diankun Zhang, Guoan Wang, Runwen Zhu, Jianbo Zhao, Xiwu Chen, Siyu Zhang, Jiahao Gong, Qibin Zhou, Wenyuan Zhang, Ningzi Wang, et al. Sparsead: Sparse query-centric paradigm for efficient end-to-end autonomous driving. *arXiv preprint arXiv:2404.06892*, 2024.

[3] 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*, pages 8340–8350, 2023.

[4] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22442–22452, 2025.

[5] 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*, 2025.

[6] 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.

[7] Ziang Luo, Kangan Qian, Jiahua Wang, Yuechen Luo, Jinyu Miao, Zheng Fu, Yunlong Wang, Sicong Jiang, Zilin Huang, Yifei Hu, et al. Mtrdrive: Memory-tool synergistic reasoning for robust autonomous driving in corner cases. *arXiv preprint arXiv:2509.20843*, 2025.

[8] Chong He, Yuechen Luo, Fang Li, Shaoqing Xu, and Fuxi Wen. DriveVer: Lightweight trajectory evaluator as test-time verifier for autonomous driving. *arXiv preprint arXiv:2607.00399*, 2026.

[9] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. *arXiv preprint arXiv:2503.19755*, 2025.

[10] 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.

[11] Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-r1: Bridging reasoning and planning in vlms for autonomous driving with reinforcement learning. *arXiv preprint arXiv:2506.18234*, 2025.

[12] Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, et al. Adathinkdrive: Adaptive thinking via reinforcement learning for autonomous driving. *arXiv preprint arXiv:2509.13769*, 2025.

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

[14] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

[15] Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. Sim-cot: Supervised implicit chain-of-thought. *arXiv preprint arXiv:2509.20317*, 2025.

[16] Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Hangjun Ye, Wenyu Liu, et al. Drivelaw: Unifying planning and video generation in a latent driving world. *arXiv preprint arXiv:2512.23421*, 2025.

[17] Shaoqing Xu, Fang Li, Yuechen Luo, Qimao Chen, Yifan Yang, Zhixiang Duan, Long Chen, and Zhi-xin Yang. World-In-Loop: Online correction via event-triggered world models for robust VLA policies. In *Computer Vision – ECCV 2026*, pages 664–681. Springer Nature Switzerland, 2026.

[18] Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

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

[20] 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.

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

[22] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3621–3631, 2023.

[23] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[24] 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*, pages 256–274. Springer, 2024.

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

[26] Yuechen Luo, Qimao Chen, Fang Li, Shaoqing Xu, Jaxin Liu, Ziying Song, Zhi-xin Yang, and Fuxi Wen. Unleashing vla potentials in autonomous driving via explicit learning from failures. *arXiv preprint arXiv:2603.01063*, 2026.

[27] Zhihao Zhu, Jiafeng Liang, Shixin Jiang, Jinlan Fu, Ming Liu, Guanglu Sun, See-Kiong Ng, and Bing Qin. Analyzing reasoning consistency in large multimodal models under cross-modal conflicts. *arXiv preprint arXiv:2601.04073*, 2026.

[28] Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*, 2025.

[29] Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*, 2025.

[30] Ken Deng, Zizheng Zhan, Wen Xiang, Wenqiang Zhu, Weihao Li, Jingxuan Xu, Tianhao Peng, Xinping Lei, Kun Wu, Yifan Yao, et al. HiPO: Hybrid policy optimization for dynamic reasoning in LLMs. *arXiv preprint arXiv:2509.23967*, 2025.

[31] Zizheng Zhan, Ken Deng, Huaixi Tang, Wen Xiang, Kun Wu, Weihao Li, Wenqiang Zhu, Jingxuan Xu, Lecheng Huang, Zongxian Feng, et al. KAT-V1: Kwai-autothink technical report. *arXiv preprint arXiv:2507.08297*, 2025.

[32] Zizheng Zhan, Ken Deng, Jinghui Wang, Xiaojiang Zhang, Huaixi Tang, Minglei Zhang, Zhiyi Lai, Haoyang Huang, Wen Xiang, Kun Wu, et al. KAT-Coder technical report. *arXiv preprint arXiv:2510.18779*, 2025.

[33] Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

[34] Arijit Ray, Ahmed Abdelkader, Chengzhi Mao, Bryan A Plummer, Kate Saenko, Ranjay Krishna, Leonidas Guibas, and Wen-Sheng Chu. Mull-tokens: Modality-agnostic latent thinking. *arXiv preprint arXiv:2512.10941*, 2025.

[35] Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025.

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

[37] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[38] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

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

[40] 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.

[41] Xianda Guo, Ruijun Zhang, Yiqun Duan, Yuhang He, Dujun Nie, Wenke Huang, Chenming Zhang, Shuai Liu, Hao Zhao, and Long Chen. Surds: Benchmarking spatial understanding and reasoning in driving scenarios with vision language models. *arXiv preprint arXiv:2411.13112*, 2024.

[42] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[43] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

[44] 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.

[45] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[46] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12037–12047, 2025.

[47] 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. *arXiv preprint arXiv:2504.01941*, 2025.

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

[49] 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.

[50] 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.

[51] 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.

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

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

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

[55] 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.

[56] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7953–7963, 2023.

[57] Xueyi Liu, Zuodong Zhong, Yuxin Guo, Yun-Fu Liu, Zhiguo Su, Qichao Zhang, Junli Wang, Yinfeng Gao, Yupeng Zheng, Qiao Lin, et al. Reasonplan: Unified scene prediction and decision reasoning for closed-loop autonomous driving. *arXiv preprint arXiv:2505.20024*, 2025.

A Appendix

In this appendix, we provide supplementary materials to support the findings and design choices of the main paper. First, **Section B** reports extended experimental results and ablation studies, covering nuScenes open-loop evaluation, hallucination analysis, reasoning paradigm comparison, alignment loss, loss-weight sensitivity, RL-stage latent alignment, latent token numbers, supervised feature layers, backbone scalability, latent generation order, and inference efficiency. **Section C** describes the data preparation protocols for the nuScenes, Bench2Drive and NuDynamics benchmarks. **Section D** presents implementation details, including model architecture, training hyperparameters, evaluation metrics on NAVSIM, Bench2Drive and nuScenes, as well as the RL reward design. Finally, **Section E** provides qualitative visualizations, including input-output examples, comparisons with textual CoT, failure cases, and supplementary analyses on SURDS and NuDynamics.

B More Results

B.1 Ablation Studies

Ablation on Hallucination Analysis. To quantitatively assess the prevalence of reasoning errors, we evaluated 12,146 samples from the test set. As shown in Table 9, Textual CoT baselines suffer from severe hallucination issues, with hallucination rates of 29.1% and 24.8% for the SFT and RL models, respectively. In this context, a hallucination is defined as a factual contradiction generated during explicit reasoning steps, such as falsely claiming the presence of a vehicle ahead when the road is clear. Our proposed LaST-VLA addresses this issue by operating in continuous latent space, thereby avoiding explicit linguistic reasoning errors. The results also show a clear inverse relationship between hallucination rate and PDMS, indicating that reducing textual reasoning errors is important for improving trajectory planning performance.

Table 9: Analysis of hallucination rates and their impact on PDMS.

Method	Halluc. Rate ↓	PDMS ↑
Textual CoT (SFT)	29.1%	83.9
Textual CoT (RL)	24.8%	87.2

Table 10: Ablation study on the necessity of the masked two-stage SFT strategy.

Method	SFT (PDMS) ↑	RL (PDMS) ↑
One Stage	85.0	90.2
Masked Two Stage	**87.3**	**91.3**

Ablation on the two-stage SFT. We conduct an ablation study to verify the necessity of the proposed masked two-stage SFT strategy. As shown in Table 10, the one-stage training strategy obtains 85.0 PDMS after SFT and 90.2 PDMS after RL. In contrast, the masked two-stage SFT achieves higher performance, reaching 87.3 after SFT and 91.3 after RL, demonstrating the effectiveness of decoupling latent reasoning learning from trajectory prediction.

Ablation on Module Innovation. To verify the efficacy of our paradigm, we conduct a direct comparison between LaST-VLA and baseline methods utilizing Query-based and Spatial Forcing mechanisms under identical physical supervision signals. As illustrated in Table 11, traditional feature distillation approaches encounter significant performance bottlenecks despite being provided with the same teacher priors. These results demonstrate that the performance leap of LaST-VLA originates from the fundamental architectural shift toward autoregressive continuous physical reasoning rather than simple feature injection. Furthermore, ablation results in Table 5 reveal that LaST-VLA achieves highly competitive performance even in the absence of explicit supervision for the Latent CoT, nearly matching the results of supervised Query-based and Spatial Forcing methods. This observation underscores the inherent structural advantage of our latent reasoning framework in capturing complex physical dynamics.

Ablation on Alignment Loss Strategy. Regarding the alignment strategy, we employ a straightforward L_2 loss. To validate this choice, we also evaluate the performance using KL divergence and Cosine similarity losses. As shown in Table 12, the L_2 loss yields slightly more stable and superior results, although the overall performance gaps among these metrics are relatively marginal.

Ablation on Sensitivity of Loss Weights. To verify the rationality of the loss weights, we conducted ablation experiments on the ratio of $\lambda_{WM} : \lambda_{3D} : \lambda_{action}$, as summarized in Table 13. Our empirical analysis reveals that an optimal balance is achieved at $\lambda = 0.01$, which effectively maintains essential cross-stage task coupling while focusing on the core objectives of each stage. In contrast, excessive

Table 11: Quantitative comparison of reasoning paradigms under identical physical supervision.

Method	SFT (PDMS) ↑	RL (PDMS) ↑
Query-based	85.1	90.0
Spatial Forcing	86.0	90.3
LaST-VLA	**87.3**	**91.3**

Table 12: Ablation study on different alignment loss functions.

Alignment Loss	SFT (PDMS) ↑	RL (PDMS) ↑
KL Loss	86.8	90.8
Cosine Loss	**87.3**	91.2
L_2 Loss (Ours)	**87.3**	**91.3**

weights ($\lambda = 0.1$) interfere with main task learning by encouraging the model to take shortcuts in Stage 1 or disrupting trajectory precision in Stage 2. Conversely, insufficient weights ($\lambda = 0.001$) fail to provide adequate regularization, resulting in the decoupling or catastrophic forgetting of critical cross-stage features, such as physical constraints and latent space structures. Ultimately, the choice of 0.01 yields the highest performance across both SFT and RL phases.

Table 13: Ablation study on loss weight settings in the two-stage SFT strategy.

ID	Stage 1	Stage 2	SFT (PDMS) ↑	RL (PDMS) ↑
1	1 : 1 : 0.1	0.1 : 0.1 : 1	86.9	91.0
2	1 : 1 : 0.001	0.001 : 0.001 : 1	86.5	90.9
Ours	1 : 1 : 0.01	0.01 : 0.01 : 1	**87.3**	**91.3**

Ablation on Latent Alignment Quality in RL Stage. During the RL stage, Latent CoT is propagated as continuous hidden features without discrete decoding or sampling. Therefore, we remove the explicit WM/3D supervision and only use reward signals to optimize the subsequent trajectory generation. To verify whether the latent alignment is preserved during RL, we monitor the Alignment Loss, defined as the weighted sum of $\mathcal{L}_{WM}$ and $\mathcal{L}_{3D}$, together with PDMS. As shown in Table 14, the Alignment Loss drops from 2.0 to 0.0303 after SFT, while PDMS increases from 55.0 to 87.3. During RL, although no explicit alignment loss is applied, the Alignment Loss remains stable around 0.03, while PDMS further improves to 91.3. This indicates that the latent physical alignment learned in SFT is largely preserved during RL, allowing reward optimization to focus on improving trajectory quality.

Table 14: Ablation on latent alignment quality in RL stage.

Metric	Before SFT	After SFT	Step 100	Step 300	Step 600	Step 772
Alignment Loss ↓	2.0000	0.0303	0.0324	0.0315	0.0318	0.0311
PDMS ↑	55.0	87.3	88.5	90.3	91.2	**91.3**

Ablation on the Number of Latent Tokens. Table 15 analyzes the sensitivity of the model to the granularity of latent representations. We observe that the configuration with $N_{\text{3D}} = 12$ and $N_{\text{WM}} = 3 \times 12$ yields the optimal performance of 91.3 PDMS. Reducing the token count (e.g., setting $N_{\text{3D}} = 6$ or $N_{\text{WM}} = 3 \times 6$) leads to performance degradation, likely due to an information bottleneck where the overly compressed latent space fails to retain sufficient fine-grained spatial details. Conversely, increasing the token count (e.g., to $N_{\text{3D}} = 24$ or $N_{\text{WM}} = 3 \times 24$) also results in inferior outcomes. This suggests that excessive tokens introduce redundancy, which complicates the optimization landscape and hinders the planner from distilling essential causal features efficiently. Thus, the selected configuration represents the optimal trade-off between representational capacity and reasoning efficiency.

Ablation on Geometric Supervision Layers. Table 16 investigates the efficacy of supervising latent thoughts with different layers of the VGGT aggregator. We observe that supervising shallow layers (Features 4 and 11) yields suboptimal performance. This is likely because these early layers primarily encode low-level visual textures rather than the explicit 3D depth and geometric structures essential for spatial planning. While performance improves with increasing depth (Feature 17), aggregating features across all layers (ID 4) fails to outperform the single final layer. We attribute this diminishing return to increased learning difficulty due to information redundancy and the need for a longer latent CoT sequence, which complicates optimization. In contrast, supervising solely with the final

Table 15: Ablation study on the number of geometric (3D) and dynamic (WM) latent tokens after RL training phase.

ID	WM/N	3D/N	NC↑	DAC↑	TTC↑	CF↑	EP↑	PDMS↑
1	3×12	6	98.6	97.5	95.6	**100**	86.5	90.9
2	3×12	24	98.4	97.7	95.2	**100**	86.5	90.8
3	3×6	12	98.6	97.5	95.4	**100**	86.5	90.8
4	3×24	12	**98.9**	97.6	**96.0**	**100**	85.9	90.9
5	3×12	**12**	98.7	**97.9**	95.6	**100**	**86.7**	**91.3**

layer (Feature 23) achieves the peak performance of 91.0 PDMS, confirming that the terminal layer encapsulates the most refined and compact 3D geometric priors for efficient reasoning.

Table 16: Ablation study on the effectiveness of supervising features from different layers of the VGGT aggregator. The Feature (23) is the last layer. **Note: These methods only have geometric (3D) latent CoT, without dynamic (WM) latent CoT.**

ID	Supervised Features	NC↑	DAC↑	TTC↑	CF↑	EP↑	PDMS↑
1	Feature (4)	98.3	97.3	94.8	**100**	86.5	90.4
2	Feature (11)	98.2	97.3	94.6	**100**	86.5	90.3
3	Feature (17)	98.4	97.5	95.3	**100**	86.4	90.7
4	Feature (4, 11, 17, 23)	98.4	**97.7**	94.8	**100**	**86.8**	90.8
5	**Feature (23) (Ours)**	**98.6**	**97.7**	**95.6**	**100**	86.4	**91.0**

Ablation on Base Model and Model Size. Table 17 evaluates the scalability and generalization of our framework across different foundation model architectures and sizes. We observe that the InternVL3 series demonstrates exceptional compatibility, where even the 2B variant achieves a high score of 91.1 PDMS, closely approaching the 8B model's peak performance of 91.3. Furthermore, while the Qwen2.5-VL series yields slightly lower metrics than the InternVL counterparts, they still exhibit substantial improvements over the standard baseline (89.0 PDMS), with the 3B and 7B models reaching 90.5 and 90.8 PDMS, respectively. This consistent superiority across diverse architectures and parameter scales confirms that our latent supervision strategy serves as a generalized and robust mechanism for enhancing driving performance, independent of the specific backbone.

Table 17: Ablation study on the impact of different base models and model scales on the performance of LaST-VLA.

ID	VLM Model	NC↑	DAC↑	TTC↑	CF↑	EP↑	PDMS↑
1	Qwen2.5-VL-3B	98.1	97.4	94.5	**100**	86.7	90.5
2	Qwen2.5-VL-7B	98.6	97.3	**95.8**	**100**	86.3	90.8
3	InternVL3-2B	98.6	97.7	95.6	**100**	86.5	91.1
4	InternVL3-8B	**98.7**	**97.9**	95.6	**100**	**86.8**	**91.3**

Ablation on Latent Generation Order. We further investigate the generation sequence of the two latent reasoning components: Dynamic tokens (WM) and Geometric tokens (3D). The results shown in Table 18, indicate that the performance fluctuation caused by swapping the generation order is negligible. Specifically, the PDMS scores differ by only 0.1 (91.3 vs. 91.2). Given this marginal stability and the slightly superior performance, we ultimately adopt the *Dynamic* $\rightarrow$ *Geometric* generation strategy in our final architecture.

Inference Efficiency Analysis. Table 19 quantifies the computational overhead across different inference modalities using the InternVL3-8B backbone. By compressing the complex reasoning process into just 48 latent tokens, LaST-VLA fundamentally bypasses the high forward-pass computation and memory I/O burdens of lengthy autoregressive decoding. Consequently, LaST-VLA achieves an end-to-end latency of 98ms, representing a 21.6% reduction compared to the explicit Textual CoT baseline. Moreover, LaST-VLA only adds 3ms of overhead compared to the reasoning-free No-CoT baseline. These results demonstrate that our latent paradigm drastically cuts token generation

Table 18: Ablation study on the generation order of latent reasoning tokens. "Dyn." denotes Dynamic tokens (WM) and "Geo." denotes Geometric tokens (3D).

ID	Generation Order	NC↑	DAC↑	TTC↑	CF↑	EP↑	PDMS↑
1	Geo. (3D) → Dyn. (WM)	**98.7**	97.6	95.5	**100**	**86.8**	91.2
2	Dyn. (WM) → Geo. (3D)	98.6	**97.9**	**95.6**	**100**	86.7	**91.3**

Table 19: Comparison of inference latency and token generation costs across different inference modalities for trajectory generation.

Method	**VLM Model**	**Inference Latency**	**Input Tokens**	**Output CoT Tokens**	**Output Traj Tokens**
No Textual CoT	InternVL3-8B	95ms	∼1220	-	∼152
Textual CoT	InternVL3-8B	125ms	∼1220	∼200	∼152
LaST-VLA	InternVL3-8B	98ms	∼1220	∼48	∼152

costs while achieving an optimal balance between advanced reasoning capabilities and real-time deployment efficiency.

C Data Preparation Details

C.1 nuScenes

To further validate the generalization and effectiveness of our proposed Spatio-Temporal Latent CoT, we extend our evaluation to the **nuScenes** dataset [42]. This large-scale benchmark comprises 1,000 complex driving scenarios, each spanning approximately 20 seconds. The data is captured via a comprehensive sensor suite, featuring a 32-beam LiDAR and six surrounding cameras that provide a 360-degree field of view. For our experiments, we utilize the standard split containing 28,130 samples for training and 6,019 samples for validation.

C.2 Bench2Drive

Bench2Drive is a closed-loop evaluation benchmark for end-to-end (E2E) autonomous driving. It consists of 2 million fully annotated frames as official training data, collected from 13638 short clips, uniformly distributed under 44 interactive scenarios (cut-in, overtaking, detour, etc), 23 weathers, and 12 towns in CARLA Leaderboard v2. For closed-loop evaluation, it requires E2E models to pass those 44 interactive scenarios under different locations which sums up to 220 routes and thus provide a comprehensive and disentangled assessment about their driving capability under different situations.

C.3 NuDynamics

NuDynamics is constructed upon the SURDS dataset with the goal of evaluating dynamic scene understanding capabilities. We select clearly visible targets from the NuScenes dataset and utilize Qwen2.5-VL-32B [51] to generate concise descriptions. Behavioral labels, including stopped crossing, moving in the same direction, moving in the opposite direction, and diagonal movement, are derived by mapping native annotations from the nuScenes dataset. The dataset comprises 4K training samples and 1K evaluation samples, and a prompt example is illustrated in Fig. 16.

D Experimental Details

Model Architecture. We leverage InternVL3-8B [45] as our vision-language backbone, bridging a 300M-parameter InternViT encoder with the Qwen2.5-7B language model. Distinguished by its resolution-adaptive architecture, the model dynamically modulates feature extraction granularity according to image complexity. By prioritizing fine-grained processing for information-rich regions while applying coarser abstraction to simpler areas, it effectively strikes an optimal balance between visual fidelity and computational overhead.

Training Parameters and Hardware Configuration. We utilize InternVL3-8B [45] as the foundation model, training across three sequential stages on 8 NVIDIA H20 GPUs. The first SFT stage fine-tunes on the navtrain-hard-24k subset for 2 epochs with a learning rate of 1×10^{-5}, batch size of 2, and 4 gradient accumulation steps; we enable the Structured Causal Mask and set loss weights to $\lambda_{WM} = \lambda_{3D} = 1.00$ and $\lambda_{action} = 0.01$. Subsequently, the second stage trains on the full 85k navtrain dataset for 2 epochs using 4 gradient accumulation steps, where the mask is disabled and weights are adjusted to $\lambda_{action} = 1.00$ with $\lambda_{WM} = \lambda_{3D} = 0.01$. Finally, the RL phase employs GRPO for 2 iterations using a learning rate of 2×10^{-6}, batch size of 2, 16 gradient accumulation steps, 8 generations, and a sampling temperature of 2.0, with reward coefficients set to $\lambda_{traj} = 8, \lambda_{fmt} = 1, \lambda_{goal} = 1$ and a KL penalty $\beta = 0.1$. The training duration for the three stages is approximately 1 hour, 3.5 hours, and 48 hours, respectively. Detailed hyperparameters are provided in Table 20.

Table 20: Training hyperparameters and time of LaST-VLA on NAVSIM benchmark (InternVL3-8B).

Stage	Hyper-parameter	Value	Training Time
SFT-Stage1	Epochs	2	1h
	Batch size	2	
	Learning rate	1×10^{-5}	
	Gradient accumulation steps	4	
	Weight decay	0.05	
	Weight ratio	0.05	
	Weight λ_{action}	0.01	
	Weight λ_{WM}	1.00	
	Weight λ_{3D}	1.00	
	Structured Causal Mask	True	
SFT-Stage2	Epochs	2	3.5h
	Batch size	4	
	Learning rate	1×10^{-5}	
	Gradient accumulation steps	2	
	Weight decay	0.05	
	Weight ratio	0.05	
	Weight λ_{action}	1.00	
	Weight λ_{WM}	0.01	
	Weight λ_{3D}	0.01	
	Structured Causal Mask	False	
RL	Epochs	2	48h
	Batch size	2	
	Learning rate	2×10^{-6}	
	Gradient accumulation steps	16	
	Weight decay	0.05	
	Weight ratio	0.05	
	Number of generations	8	
	Number of iterations	2	
	KL penalty $\beta = 0.1$	0.1	
	Temperature	2.0	
	Weight λ_{traj}	8	
	Weight λ_{fmt}	1	
	Weight λ_{goal}	1	

Metric. To comprehensively evaluate planning performance, we employ specific metrics tailored to each benchmark. For closed-loop planning within the NAVSIM suite, we adopt the Predictive Driver Model Score (PDMS) for **NAVSIMv1** [40] and the Extended Predictive Driver Model Score (EPDMS) for **NAVSIMv2** [44]. Additionally, on the nuScenes dataset, we follow the protocol of UniAD [1] to assess open-loop trajectory quality using L2 metrics and collision rate.

For NAVSIMv1, PDMS integrates five sub-metrics: No At-Fault Collision (NC), Drivable Area Compliance (DAC), Time-to-Collision (TTC), Comfort (C), and Ego Progress (EP) to produce a

comprehensive closed-loop planning score. Its calculation formula is defined as follows:

$$PDMS = NC \times DAC \times \left(\frac{5 \times EP + 5 \times TTC + 2 \times C}{12} \right), \tag{10}$$

For NAVSIMv2, the EPDMS metric includes several components categorized as penalties or weighted subscores. Its key metrics are No at-Fault Collision (NC), Drivable Area Compliance (DAC), Driving Direction Compliance (DDC), Traffic Light Compliance (TLC), Ego Progress (EP), Time to Collision (TTC), Lane Keeping (LK), History Comfort (HC), and Extended Comfort (EC). Its calculation formula is defined as follows:

$$EPDMS = NC \times DAC \times DDC \times TLC \times \left(\frac{5 \times EP + 2 \times LK + 2 \times HC + 5 \times TTC + 2 \times EC}{16} \right). \tag{11}$$

For Bench2Drive, we evaluate interactive closed-loop driving performance using Driving Score (DS), Success Rate (SR), Efficiency, and Comfort. DS jointly reflects route progress and safety by combining the completed route percentage with multiplicative penalties for infractions, while SR measures the fraction of routes that are successfully completed within the time limit and without traffic violations:

$$\text{Success Rate} = \frac{n_{\text{success}}}{n_{\text{total}}}, \quad \text{Driving Score} = \frac{1}{n_{\text{total}}} \sum_{i=1}^{n_{\text{total}}} \text{RC}_i \prod_{j=1}^{n_{i,\text{penalty}}} p_{i,j}. \tag{12}$$

Here, n_{success} and n_{total} denote the number of successful routes and the total number of evaluated routes, respectively; RC_i is the route completion ratio of the i-th route, and $p_{i,j}$ denotes the j-th infraction penalty applied to that route. Efficiency measures whether the ego vehicle maintains reasonable progress by comparing its speed with the average speed of nearby vehicles at each checked frame, and the final score is obtained by averaging these speed percentages over all checks. Comfort evaluates the smoothness of the driving behavior by dividing the trajectory into segments with a timestep interval of $n = 20$ and checking whether the kinematic properties within each segment exceed predefined thresholds; the final metric is the proportion of smooth segments among all trajectory segments.

For nuScenes, we report the L2 error between the planned trajectory and the expert trajectory, together with the collision rate, as the planning evaluation metrics.

Detailed Rewards Design in RL. To effectively guide the policy optimization during the Reinforcement Learning stage, we formulate a composite reward function comprising three distinct components: trajectory quality (R_{traj}), structural validity (R_{fmt}), and goal alignment (R_{goal}).

For the trajectory reward r_{traj}, we directly leverage the feedback from the NAVSIM simulator. Instead of using a surrogate loss, we utilize the continuous PDMS (or EPDMS for NAVSIMv2) score, ranging from 0 to 1, as the primary reward signal. This encourages the model to generate paths that maximize safety and drivability metrics as defined above.

For the **format reward** r_{fmt}, we assign a total of 1.0 point to enforce structural integrity. This is equally distributed: 0.5 points are awarded for the correct usage of structural tags (<latent_start_wm>...<latent_end_wm>, <latent_start_3d>...<latent_end_3d>, and <answer>...</answer>), and the remaining 0.5 points validate the syntax of the trajectory waypoints to ensure the output is well-formed and machine-parsable.

For the **goal reward** r_{goal}, we employ a piecewise function based on the L1 distance to encourage precise alignment between the predicted endpoint and the ground truth. The specific calculation is defined as follows:

$$R_{goal} = \begin{cases} 1 & \text{if } 0 < dis < 2 \\ 0.8 & \text{if } 2 \leq dis < 4 \\ 0.6 & \text{if } 4 \leq dis < 6 \\ 0.4 & \text{if } 6 \leq dis < 10 \\ 0.2 & \text{if } 10 \leq dis < 15 \\ 0 & \text{if } dis > 15 \end{cases} \tag{13}$$

E Visualization Analysis

Input and Output Visualization on NAVSIM. Fig. 6 and Fig. 7 illustrate representative examples of the input and output prompts, respectively, employed by LaST-VLA on the NAVSIM benchmark.

Extended qualitative comparison between Textual CoT and LaST-VLA. Serving as a supplement to Fig. 5, Fig. 8 presents additional scenarios comparing the textual baseline (Red) with LaST-VLA (Green). These cases further corroborate the superiority of our approach in maintaining geometric compliance (e.g., lane adherence) and ensuring dynamic safety (e.g., timely braking), reinforcing the efficacy of the physically grounded latent reasoning paradigm.

Failure Analysis and Limitations. We analyzed failure cases where the planning performance drops significantly, such as when the PDMS is 0. As shown in Fig. 9, the main failure pattern occurs when the target trajectory extends beyond the Field of View (FOV) of the front-view camera. This typically happens in scenarios involving sharp turns or complex intersections, where the model lacks sufficient visual information about the peripheral environment. Consequently, the planner may generate paths that deviate from the drivable area. This issue is primarily due to our current reliance on a single front-view camera. In future work, we plan to address this by incorporating surround-view inputs and temporal information to expand the perceptual range for more robust planning.

Qualitative Visualization on SURDS and NuDynamics. To further evaluate the comprehensive reasoning capabilities of LaST-VLA, we present qualitative visualizations on the SURDS and NuDynamics benchmarks. Fig. 10 through 15 illustrate representative input queries and corresponding model responses across diverse SURDS tasks, such as Yaw Angle Determination and Pixel Location Estimation. These examples confirm that our model can accurately interpret complex 3D geometric constraints from 2D images. Furthermore, the visualizations on the NuDynamics benchmark (Fig. 16) highlight the model's proficiency in analyzing scene evolution and motion trends. Collectively, these results demonstrate that our latent reasoning paradigm effectively equips the agent with robust 3D spatial awareness and dynamic scene understanding.

You are an autonomous driving agent. You have access to the front-view camera image of a vehicle, the drive command, the current velocity and acceleration of the vehicle, and the ego vehicle's historical trajectory.\n The drive command represents the desired control command that the system wants the vehicle to maintain. The vehicle's velocity and acceleration are represented as lists [x, y], where x and y correspond to the respective components in the x and y directions. The ego vehicle's historical trajectory consists of past positions in the format [x, y, heading]. It may include up to 3 points, each representing a timestep of 0.5 seconds, with a maximum history of 1.5 seconds. \n

Your task is to generate the corresponding 8-step trajectory in the <answer> </answer> tags. The trajectory must be consistent with and accurately reflect the decisions described in the reasoning process. For the content of answer, you need predict the vehicle's future trajectory over the next 8 timesteps (covering 4 seconds total), based on the surrounding traffic environment observed in the provided images. Each timestep represents 0.5 seconds. Each trajectory point should be a 3D vector: [x, y, heading], where x and y are the 2D coordinates and heading is the orientation angle. The positive y-axis points to the left side of the vehicle, and the negative y-axis points to the right. The positive x-axis points in the forward driving direction. If you fail to provide a response, it may lead to potential harm or injury. \n

Prior to the <think> block, two sets of latent priors are embedded to ground reasoning: 36 world-model tokens (wrapped in <|latent_start_wm|> and <|latent_end_wm|>) encoding the front-view visual dynamics and 8-frame future evolution, and 12 spatial tokens (wrapped in <|latent_start_3d|> and <|latent_end_3d|>) encoding the 3D scene geometry. Utilize these spatiotemporal features to mentally reconstruct the scene context and guide subsequent trajectory prediction. \n

Figure 6: A simplified representation of LaST-VLA input on NAVSIM benchmark.

<|latent_start_wm|>
<|latent_pad_wm_1|><|latent_pad_wm_1|>…<|latent_pad_wm_1|><|latent_pad_wm_1|><|latent_pad_wm_1|>
<|latent_pad_wm_2|><|latent_pad_wm_2|>…<|latent_pad_wm_2|><|latent_pad_wm_2|><|latent_pad_wm_2|>
<|latent_pad_wm_3|><|latent_pad_wm_3|>…<|latent_pad_wm_3|><|latent_pad_wm_3|><|latent_pad_wm_3|>
<|latent_end_wm|>
<|latent_start_3d|>
<|latent_pad_3d|><|latent_pad_3d|><|latent_pad_3d|>…<|latent_pad_3d|><|latent_pad_3d|><|latent_pad_3d|>
<|latent_end_3d>
<answer>
[2.32, 0.01, 0.01], [5.08, 0.05, 0.01], [8.22, 0.1, 0.01], … , [19.83, 0.14, 0.0], [24.34, 0.12, -0.0], [29.13, 0.08, -0.01]
</answer>

Figure 7: A simplified representation of LaST-VLA output on NAVSIM benchmark.



Figure 8: Qualitative visualization comparing the **Textual CoT baseline (Red)** and **LaST-VLA (Green)** on NAVSIM benchmark. (a) **Drivable Area Compliance (DAC):** Our method maintains precise lane adherence, whereas the baseline violates spatial boundaries. (b) **Time-to-Collision (TTC):** Our method accurately anticipates dynamics to avoid rear-end collisions, while the baseline fails to brake effectively.



Figure 9: **Qualitative visualization of failure cases on NAVSIM benchmark.** The red line represents the trajectory predicted by LaST-VLA, while the green line denotes the Ground Truth. We observe that failures predominantly occur when the planned path extends beyond the front-view camera's field of view (FOV). Lacking sufficient visual context in these peripheral regions, the model struggles with precise spatial grounding, leading to potential collisions or deviations from the drivable area.



Question: <image>\nTask Description: \nThe primary goal of this task is to identify the direction that the specified object is facing in the given image. The camera in the image is facing South, and you need to analyze the object's orientation based on this reference.\n\nQuestion: \nWhich direction is the white car facing in the image?\n\nOptions: \n- South\n- West\n- East\n- North\n\n

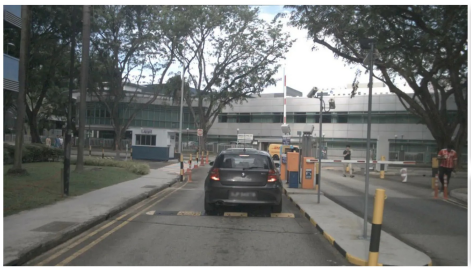
<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>** **<|latent_start_3d|>** <|latent_pad_3d|>... **<|latent_end_3d>** **<answer>** North **</answer>**



Question: <image>\nTask Description: \nThe primary goal of this task is to identify the direction that the specified object is facing in the given image. The camera in the image is facing South, and you need to analyze the object's orientation based on this reference.\n\nQuestion: \nWhich direction is the orange rigid bus facing in the image?\n\nOptions: \n- East\n- West\n- North\n- South\n\n

<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>** **<|latent_start_3d|>** <|latent_pad_3d|>... **<|latent_end_3d>** **<answer>** South **</answer>**

Figure 10: Examples of the Yaw Angle Determination task on the SURDS benchmark.



Question: <image>\nTask Description:\nThe primary goal of this task is to accurately identify and provide the coordinates of a specified object within a given image. Your task is to analyze the image, locate the object, and return its position in the form of coordinates [x, y].\n\nQuestion:\nWhere is the black car located in the image?\n\n

<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>** **<|latent_start_3d|>** <|latent_pad_3d|>... **<|latent_end_3d>** **<answer>** [837, 620] **</answer>**



Question: <image>\nTask Description:\nThe primary goal of this task is to accurately identify and provide the coordinates of a specified object within a given image. Your task is to analyze the image, locate the object, and return its position in the form of coordinates [x, y].\n\nQuestion:\nWhere is the white rigid bus located in the image?\n\n

<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>** **<|latent_start_3d|>** <|latent_pad_3d|>... **<|latent_end_3d>** **<answer>** [919, 477] **</answer>**

Figure 11: Examples of the Pixel Location Estimation task on the SURDS benchmark.



Question: <image>\nTask Description:\nThe primary goal of this task is to estimate the vertical distance of the specified object in the image from the camera, which is positioned at the origin. You need to analyze the image and choose the correct range of distance from the camera based on the visual cues provided.\n\nQuestion:\nHow far is the vertical distance of the orange rigid bus in the picture from the camera?\n\nOptions:\n- Between 32 meters and 39 meters\n- Between 27 meters and 30 meters\n- Between 22 meters and 25 meters\n\n

<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>** **<|latent_start_3d|>** <|latent_pad_3d|>... **<|latent_end_3d>** **<answer>** Between 32 meters and 39 meters **</answer>**



Question: <image>\nTask Description:\nThe primary goal of this task is to estimate the vertical distance of the specified object in the image from the camera, which is positioned at the origin. You need to analyze the image and choose the correct range of distance from the camera based on the visual cues provided.\n\nQuestion:\nHow far is the vertical distance of the black truck in the picture from the camera?\n\nOptions:\n- Between 14 meters and 17 meters\n- Between 9 meters and 12 meters\n- Between 19 meters and 25 meters\n\n

<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>** **<|latent_start_3d|>** <|latent_pad_3d|>... **<|latent_end_3d>** **<answer>** Between 19 meters and 25 meters **</answer>**

Figure 12: Examples of the Depth Range Determination on the SURDS benchmark.



Question: <image>\nTask Description: \nThe primary goal of this task is to determine which of the two objects is closer to the camera that captured the image below. You need to assess the relative distance between the two objects based on the camera's perspective.\n\nQuestion: \nWhich object, the white car or the orange rigid bus, is closer to the camera?\n\nOptions:\n- The white car\n- The orange rigid bus\n- Almost the same\n\n

<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>** **<|latent_start_3d|>** <|latent_pad_3d|>... **<|latent_end_3d>** **<answer>** The white car **</answer>**



Question: <image>\nTask Description: \nThe primary goal of this task is to determine which of the two objects is closer to the camera that captured the image below. You need to assess the relative distance between the two objects based on the camera's perspective.\n\nQuestion: \nWhich object, the silver car or the yellow bicycle, is farther to the camera?\n\nOptions:\n- The silver car\n- The yellow bicycle\n- Almost the same\n\n

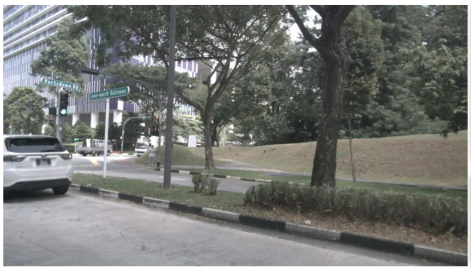
<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>** **<|latent_start_3d|>** <|latent_pad_3d|>... **<|latent_end_3d>** **<answer>** The silver car **</answer>**

Figure 13: Examples of the Distance Estimation task on the SURDS benchmark.



Question: *<image>\nTask Description:\nThe primary goal of this task is to determine the relative left-right positioning of the two objects from the camera's perspective.\n\nQuestion:\nWhich is further left, the adult wearing a blue shirt or the white car?\n\nOptions:\n- The adult wearing a blue shirt\n- The white car\n- Almost the same\n\n*

<\|latent_start_wm\|> *<\|latent_pad_wm_1\|>...* ***<\|latent_end_wm\|>***
<\|latent_start_3d\|> *<\|latent_pad_3d\|>...* ***<\|latent_end_3d>***
<answer> *The adult wearing a blue shirt* ***</answer>***



Question: *<image>\nTask Description:\nThe primary goal of this task is to determine the relative left-right positioning of the two objects from the camera's perspective.\n\nQuestion:\nWhich is further left, the white car or the gray truck?\n\nOptions:\n- The white car\n- The gray truck\n- Almost the same\n\n*

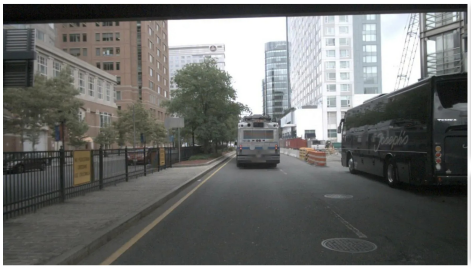
<\|latent_start_wm\|> *<\|latent_pad_wm_1\|>...* ***<\|latent_end_wm\|>***
<\|latent_start_3d\|> *<\|latent_pad_3d\|>...* ***<\|latent_end_3d>***
<answer> *The white car* ***</answer>***

Figure 14: Examples of the Left/Right Determination task on the SURDS benchmark.



Question: *Task Description:\nThe primary goal of this task is to determine the relative front-back positioning of the two objects from the camera's perspective, where the object farther from the camera is considered to be more forward.\n\nQuestion:\nIs the white car in front of the adult wearing red and orange clothes?\n\nOptions:\n- Yes\n- No\n- Almost the same in terms of front-back position\n\n*

<\|latent_start_wm\|> *<\|latent_pad_wm_1\|>...* ***<\|latent_end_wm\|>***
<\|latent_start_3d\|> *<\|latent_pad_3d\|>...* ***<\|latent_end_3d>***
<answer> *Yes* ***</answer>***



Question: *<image>\nTask Description:\nThe primary goal of this task is to determine the relative front-back positioning of the two objects from the camera's perspective, where the object farther from the camera is considered to be more forward.\n\nQuestion:\nIs the silver bendy bus behind the black rigid bus?\n\nOptions:\n- Yes\n- No\n- Almost the same in terms of front-back position\n\n*

<\|latent_start_wm\|> *<\|latent_pad_wm_1\|>...* ***<\|latent_end_wm\|>***
<\|latent_start_3d\|> *<\|latent_pad_3d\|>...* ***<\|latent_end_3d>***
<answer> *No* ***</answer>***

Figure 15: Examples of the Front/Behind Determination task on the SURDS benchmark.



Question: <image>\nTask Description: \nThe primary goal of this task is to identify the motion state of the specified object in the given image.\n\nQuestion: \nWhat is the state of the white truck?\n\nOptions: \n- Crossing\n- Go straight in opposite direction\n- Stopped\n- Go straight in same direction\n\n

<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>**
<|latent_start_3d|> <|latent_pad_3d|>... **<|latent_end_3d>**
<answer> Stopped **</answer>**



Question: <image>\nTask Description: \nThe primary goal of this task is to identify the motion state of the specified object in the given image.\n\nQuestion: \nWhat is the state of the black car?\n\nOptions: \n- Stopped\n- Go straight in same direction\n- Crossing\n- Go straight in opposite direction\n\n

<|latent_start_wm|> <|latent_pad_wm_1|>... **<|latent_end_wm|>**
<|latent_start_3d|> <|latent_pad_3d|>... **<|latent_end_3d>**
<answer> Go straight in same direction **</answer>**

Figure 16: Examples of the Motion State Estimation task on the NuDynamics benchmark.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the papers contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper discuss the limitations of the work performed by the authors.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: N/A

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully disclose all the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data will be provided.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specify all the training and test details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: No error bars needed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provide sufficient information on the computer resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: No societal impact of the work is performed.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets introduced in the paper are well documented and the documentation is provided alongside the assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.864

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper describe the usage of LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.