OneVL: One-Step Latent Reasoning and Planning with Vision-Language Explanation

Jinghui Lu, Jiayi Guan, Zhijian Huang, Jinlong Li, Guang Li, Lingdong Kong, Yingyan Li, Han Wang, Shaoqing Xu, Yuechen Luo, Fang Li, Chenxu Dang, Junli Wang, Tao Xu, Jing Wu, Jianhua Wu, Xiaoshuai Hao, Wen Zhang, Tianyi Jiang, Lingfeng Zhang, Lei Zhou, Yingbo Tang, Jie Wang, Yinfeng Gao, Xizhou Bu, Haochen Tian, Yihang Qiu, Feiyang Jia, Lin Liu, Yigu Ge, Hanbing Li, Jiahong Chen, Irene Li, Yuannan Shen, Jianwei Cui, Hongwei Xie, Bing Wang, Haiyang Sun, Jingwei Zhao, Jiahui Huang, Pei Liu, Zeyu Zhu, Yuncheng Jiang, Zibin Guo, Hanchao Leng, Chuhong Gong, Kun Ma, Guang Chen, Kuiyuan Yang, Hangjun Ye, Long Chen[†]

Xiaomi Embodied Intelligence Team ([†]Corresponding author)

jinghui.lu2026@gmail.com, guanjiayi@.com, longchen@xiaomi.com

Project page: https://Xiaomi-Embodied-Intelligence.github.io/OneVL

Code: https://github.com/xiaomi-research/OneVL

Abstract

Chain-of-Thought (CoT) reasoning has become a powerful driver of trajectory prediction in VLA-based autonomous driving, yet its autoregressive nature imposes latency prohibitive for real-time deployment. Latent CoT methods attempt to close this gap by compressing reasoning into continuous hidden states, but consistently fall short of their explicit counterparts. We argue that this is because purely linguistic latent representations compress a symbolic abstraction of the world rather than the spatiotemporal dynamics that govern driving. We present **OneVL** (**One**-step latent reasoning and planning with **V**ision-**L**anguage explanations), a unified VLA and world model framework that routes reasoning through compact latent tokens supervised by dual auxiliary decoders. It comprises a language decoder that reconstructs text CoT and a visual world model decoder that predicts future-frame tokens, forcing the latent space to internalize how the scene evolves. A three-stage training pipeline progressively aligns these latents with trajectory, language, and visual objectives. During inference, the decoders are discarded, and all latent tokens are prefilled in a single parallel pass, matching answer-only prediction speed. Across four benchmarks, OneVL consistently surpasses explicit CoT, delivering state-of-the-art accuracy at answer-only latency.

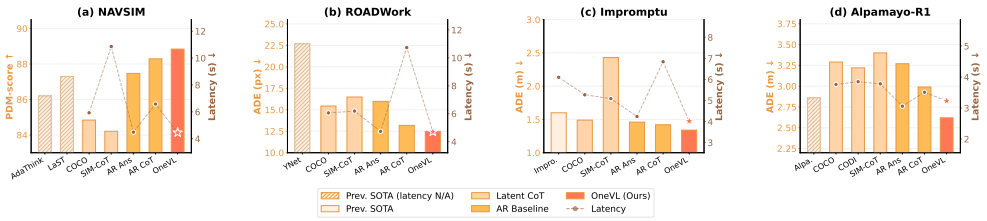


Figure 1: Accuracy and efficiency comparison on four benchmarks. Language-only latent CoT methods underperform explicit CoT. **OneVL** surpasses it while matching answer-only prediction latency.

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

1 Introduction

Vision-Language Models (VLMs) [3, 16, 22, 32] have rapidly become a foundational building block for autonomous driving, unifying scene understanding, natural language reasoning, and end-to-end trajectory planning within a single model [41, 54]. When further extended to produce action outputs such as trajectory waypoints, these models are known as Vision-Language-Action models (VLAs) [6, 8, 36]. A central driver of recent progress in VLA-based driving is Chain-of-Thought (CoT) reasoning [8, 35, 36], where the model articulates intermediate reasoning steps before generating a trajectory, yielding substantial gains in prediction quality [48]. By surfacing scene semantics, anticipated agent behaviors, and driving intent, CoT supervision binds predictions into coherent causal chains and reduces planning errors. A unifying explanation for why CoT works comes from the *compression view of intelligence* [11, 27], that is, under next-token supervision, a model forced to articulate intermediate steps must compress its understanding into structured, generalizable representations rather than memorize shallow input–output mappings.

Yet deploying CoT in real driving systems exposes a sharp tension between interpretability and efficiency. Standard autoregressive (AR) CoT has to output every reasoning token before the trajectory can be produced. This yields latency proportional to chain length, which is far above answer-only prediction. At the same time, for VLA-based driving tasks, explicit CoT chains are highly redundant, suggesting that the essential reasoning content can be compressed into a far more compact form [46] without sacrificing generalization.

Latent CoT and its limitations. A growing line of work pursues this direction by replacing explicit reasoning tokens with compact latent representations. COCONUT [19] introduced curriculum learning over latent thought tokens; CODI [39] extended this with self-distillation; SIM-CoT [49] attached a text-decoding auxiliary decoder for direct text supervision during latent training. However, these methods were designed for language-only reasoning and make no use of the rich visual structure that defines driving scenes. Their purely linguistic latents prove insufficient for multimodal reasoning demanded by trajectory prediction. As shown by Figure 1, every language-only latent CoT method we adapt to VLA tasks (COCONUT, CODI, SIM-CoT) underperforms explicit CoT across all benchmarks. Concurrent work explores latent reasoning with world-model tokens for driving [43] (see Section 2). More fundamentally, natural language descriptions of driving scenes are inherently abstract. They encode semantic labels rather than the spatiotemporal dynamics that determine future states. A latent vector that compresses language is therefore compressing an abstraction of the world, not the underlying scene dynamics.

OneVL: one-step latent reasoning with vision-language explanations. We present OneVL, a framework that overcomes these limitations through two key innovations. First, we introduce *dual-modal auxiliary decoders*, namely a language decoder that reconstructs human-readable CoT reasoning from compact language latent tokens, and a visual decoder that predicts anticipated future frames [9, 40] from visual latent representations. The visual decoder serves as a *world model* auxiliary. By forcing the latents to anticipate future scene appearance, it ensures that the bottleneck encodes scene dynamics (agent motion, road geometry evolution, emerging hazards) rather than abstract symbolic summaries. Together, the dual reconstruction objectives ensure that the compressed latents encode generalizable structure rather than superficial correlations [11, 45].

Second, we design a *prefill inference* mechanism. At inference time, the auxiliary decoders are discarded, and all latent tokens are prefilled into the prompt context as fixed inputs, enabling single-pass generation that matches answer-only AR speed. Here, "one-step" refers to inference only, since reasoning is distilled into the latents during training and at inference they are consumed in one parallel prefill pass rather than decoded autoregressively. Figure 2 contrasts explicit CoT, prior implicit latent CoT, and OneVL. Besides, we propose a principled *three-stage training pipeline* that progressively aligns these latents with trajectory, language, and visual objectives to avoid the learning of superficial shortcuts. The contributions are as follows:

- We introduce the **OneVL framework**, a latent CoT architecture with dual-modal auxiliary decoders. A language decoder and a visual world model decoder that jointly supervise compact latent tokens to encode both linguistic reasoning and future scene dynamics.
- A three-stage training pipeline progressively aligns the latent bottleneck with trajectory, language, and visual objectives. In inference, all latent tokens are prefilled in a single pass, matching answer-only latency while enabling both linguistic and visual interpretability.

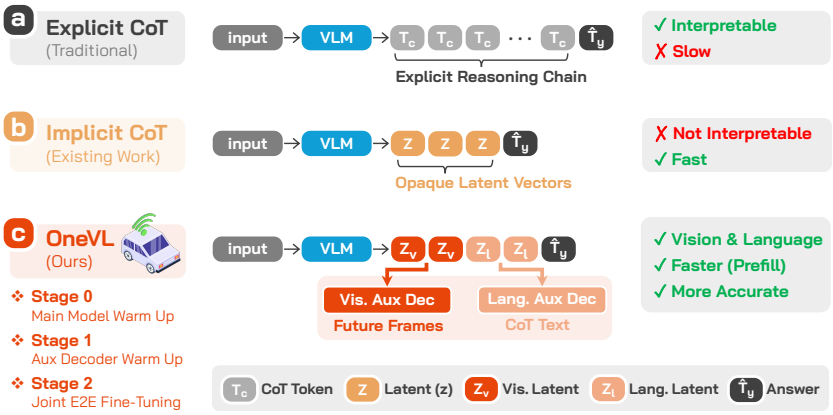


Figure 2: Comparison of three CoT paradigms. (a) Explicit CoT generates a chain of discrete reasoning tokens; (b) Implicit CoT compresses reasoning into opaque latent vectors $\mathcal{Z}$; (c) **OneVL** uses visual ($\mathcal{Z}_v$) and language ($\mathcal{Z}_l$) latent tokens that can be decoded to future status and text CoT. In (c), the latents are supervised only at training time by the decoders of Section 3 (Eqs. 1–6).

- OneVL achieves superior performance across four benchmarks (NAVSIM, ROADWork, Impromptu, Alpamayo-R1), **surpassing explicit AR CoT on every benchmark** consistently across random seeds; to our knowledge, it is the first latent CoT method to do so across four driving benchmarks. Ablation studies confirm each component's contribution. Experimental results show that introducing implicit world model supervision and the staged training recipe are essential to the performance of VLA-based methods in latent reasoning.

2 Related Work

Latent Chain-of-Thought. Since explicit CoT incurs inference overhead proportional to the length of the reasoning chain, a growing line of work internalizes reasoning into continuous latent representations [7, 12, 28, 29, 42, 53]. COCONUT [19] generalizes step-by-step internalization to continuous latent thought tokens through a staged curriculum. Compressed Chain of Thought [7] distills explicit traces into dense summary vectors prepended to the input. CODI [39] adopts sequence-level self-distillation, aligning a student's anchor latent state with a teacher's full CoT. SIM-CoT [49] addresses latent instability with a plug-and-play auxiliary decoder that aligns each implicit token with its explicit counterpart at training time. More recently, LaDiR [24] encodes reasoning steps into blocks of latent thoughts and iteratively refines them with a latent diffusion model, enabling diverse reasoning paths. As discussed, all of these methods were developed for language-only tasks and do not transfer effectively to VLA-based autonomous driving, as they lack visual grounding for spatial-temporal reasoning; in this work we adapt three representative ones (COCONUT, CODI, SIM-CoT) as baselines.

VLMs and VLAs for autonomous driving.

Beyond the foundational VLM works and CoT-augmented driving models discussed in Section 1, a parallel line of research has focused on establishing richer evaluation and supervision signals for language-grounded driving [35, 36, 48]. Typically, recent VLA models pair language reasoning with waypoint or action outputs [33, 34, 56]. DriveVLA-W0 [30] employs world modeling to generate dense self-supervised signals that amplify data scaling laws in VLA-based driving. AdaThinkDrive [35] introduces adaptive CoT for driving decisions, LaST-VLA [36] trains a large vision-language-action model on driving data, and Alpamayo-R1 [48] explicitly bridges reasoning traces with long-tail action prediction. Closest to our work, the concurrent LCDrive [43] also reasons in a latent space for end-to-end driving, interleaving action-proposal and world-model tokens and refining the policy with closed-loop RL; it likewise reports latent CoT outperforming text CoT on some metrics of a different benchmark. OneVL differs in that its latents are shaped by *training-only* decoders (decodable into text CoT and future frames) and are prefilled in a single pass, so no latent tokens are generated at inference. OneVL thus addresses the latency cost of explicit CoT brought by existing VLA-based methods through dual-modal latent supervision and prefill inference.

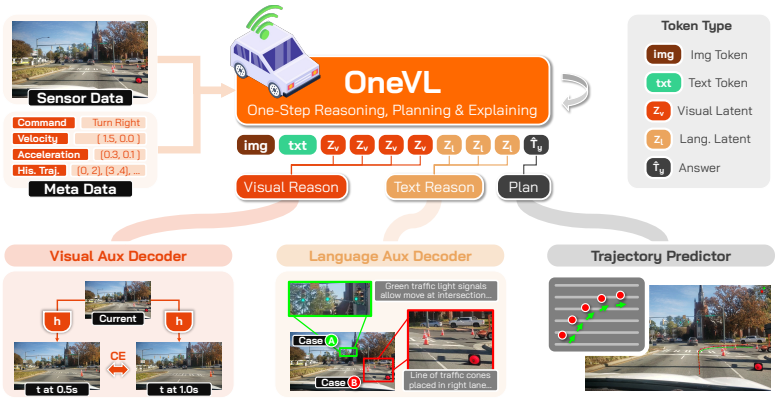


Figure 3: **OneVL** feeds an image and text prompt into a VLM. Its hidden states contain image ($\mathcal{T}_v$), text ($\mathcal{T}_l$), visual latent ($\mathcal{Z}_v$), language latent ($\mathcal{Z}_l$), and trajectory answer ($\hat{\mathcal{T}}_y$) tokens. During training, latent-position hidden states are routed to a visual aux. decoder (predicting future-frame tokens at +0.5s/+1.0s) and a language aux. decoder (predicting CoT reasoning). During inference, both decoders are dropped, and latent tokens are prefilled, preserving answer-only AR latency.

World models for autonomous driving. The concept of the *world model* originates from model-based reinforcement learning, where it seeks to emulate human cognitive processes and predict the effects of actions on environmental evolution [17, 18], particularly in 3D and 4D spaces [4, 31, 52]. To further enhance spatial reasoning, several approaches incorporate advanced perception frameworks [47, 50, 51, 57, 58] to achieve a more robust understanding of 3D environments. With advances in video generation and the introduction of the Joint Embedding Predictive Architecture by Assran et al. [1], LeCun [26], the scope and applications of world models have broadened considerably [13, 20, 21, 44]. In autonomous driving, world models serve data generation [2], closed-loop evaluation [14, 55], and representation learning [30, 38]. In contrast, OneVL uses short-horizon future visual token prediction as a *training-only* world model auxiliary paired with compressed latent CoT inside a single VLA. This auxiliary steers the bottleneck toward scene dynamics where language-only implicit CoT falls short, and is dropped at inference so prefilled latents deliver answer-only latency.

3 Methodology

This section elaborates on model architecture of **OneVL** and the principled staged training pipeline.

3.1 Model Architecture

OneVL augments a pretrained VLM with compact latent tokens and dual auxiliary decoders for explanation. Figure 3 gives a complete overview. The model consists of the following components.

Main Vision-Language Model. The backbone of OneVL is Qwen3-VL-4B-Instruct [3], consisting of a Vision Encoder (ViT), Visual Projector (MLP Aligner), and Large Language Model (LLM), all initialized from the provided checkpoint and fully trainable in Stages 0 and 2. The backbone is optimized via a standard next-token prediction cross-entropy loss $\mathcal{L}_c$ on both trajectory answers and latent reasoning tokens.

One innovation of OneVL is introducing *latent tokens* to replace the explicit CoT text before outputting the trajectory. In this work, we define two classes of latent tokens embedded in the assistant response, placed before the trajectory answer where explicit CoT would normally appear. **(1) Visual latent tokens** ($\langle|\texttt{latent-vis}|\rangle$): $\mathcal{C}_v = 4$ tokens (realized as 35 original vocabulary tokens) encoding spatial and temporal visual reasoning about the future scene state. **(2) Language latent tokens** ($\langle|\texttt{latent}|\rangle$): $\mathcal{C}_t = 2$ tokens (realized as 20 original vocabulary tokens) encoding the model's implicit language-grounded reasoning. Both sets serve as reasoning carriers whose hidden states at the VLM output layer are fed into auxiliary decoders. Under causal attention, the latents attend to everything preceding them (system prompt, image, and the ego-state/intent fields of the query, *i.e.*, command, velocity, and history) but not to the trajectory answer generated after them.

Language Auxiliary Decoder. The language auxiliary decoder $\mathcal{D}_l$ recovers human-readable CoT reasoning from the compact language latent hidden states $\mathcal{H}_l \in \mathbb{R}^{\mathcal{C}_t \times d}$. We additionally supply the current-frame ViT patch embeddings $\mathcal{V} \in \mathbb{R}^{N_v \times d}$ and form the input by projection and concatenation:

$$\mathcal{Z}_l = \Big[\mathrm{W}_l(\mathcal{V}),\ \mathrm{W}_l(\mathcal{H}_l)\Big], \tag{1}$$

where $\mathrm{W}_l \in \mathbb{R}^{d \times d}$ is an MLP layer. The language auxiliary decoder is trained with cross-entropy on the ground-truth CoT text $\mathcal{T}_{y_t}$:

$$\mathcal{L}_l = -\sum_{i=1}^{|\mathcal{T}_{y_t}|} \log P_{\mathcal{D}_l}\left(\mathcal{T}_{y_t,i} \mid \mathcal{Z}_l, \mathcal{T}_{y_t,<i}\right) . \tag{2}$$

Visual Auxiliary Decoder. The visual auxiliary decoder $\mathcal{D}_v$ predicts anticipated future-frame visual tokens. Future-frame prediction is a natural world model objective. To minimize reconstruction loss on unseen configurations, the visual latent tokens have to encode how the scene evolves, not just its current appearance. This objective is *predictive* rather than interventional (Section 5). The decoder receives:

$$\mathcal{Z}_v = \Big[\mathrm{W}_v(\mathcal{V}),\ \mathrm{W}_v(\mathcal{H}_v)\Big] , \tag{3}$$

where $\mathcal{H}_v \in \mathbb{R}^{\mathcal{C}_v \times d}$ are the visual latent hidden states and $\mathrm{W}_v \in \mathbb{R}^{d \times d}$ is the MLP layer. We adopt the IBQ visual tokenizer [40] from Emu3.5 [9] with a codebook of 131,072 discrete visual codebook to encode the continuous image pixels into discrete visual tokens. The Qwen3-VL vocabulary is extended accordingly. The visual token sequences are constructed offline. Let $\mathcal{T}_{y_v} = [\mathcal{T}_{y_v,1},\ \mathcal{T}_{y_v,2}]$ be the concatenated target for future frames at $+0.5$s and $+1.0$s, the loss is:

$$\mathcal{L}_v = -\sum_{t=1}^{|\mathcal{T}_{y_v}|} \log P_{\mathcal{D}_v}(\mathcal{T}_{y_v,t} \mid \mathcal{Z}_v, \mathcal{T}_{y_v,<t}) . \tag{4}$$

Section D verifies that the latents, not $\mathcal{V}$ alone, carry the decoded information.

Combined Training Objective. The total loss is a weighted sum of three components:

$$\mathcal{L} = \mathcal{L}_c + \lambda_l \mathcal{L}_l + \lambda_v \mathcal{L}_v , \tag{5}$$

where $\lambda_l = 1.0$ and $\lambda_v = 0.1$. The lower weight on $\mathcal{L}_v$ reflects that visual token reconstruction is harder and prevents it from dominating the training signal.

Prefill Inference. At inference time, the auxiliary decoders are discarded. Both visual and language latent tokens are prefilled into the prompt context as fixed token sequences:

$$[\text{System, User query, } \langle|\texttt{start-latent-vis}|\rangle, \underbrace{\langle|\texttt{latent-vis}|\rangle \cdots}_{\mathcal{C}_v}, \langle|\texttt{end-latent-vis}|\rangle, \langle|\texttt{start-latent}|\rangle, \underbrace{\langle|\texttt{latent}|\rangle \cdots}_{\mathcal{C}_t}, \langle|\texttt{end-latent}|\rangle] . \tag{6}$$

Since transformers process the prefill in parallel, these additional tokens add negligible overhead. The model then generates only trajectory tokens autoregressively, yielding latency nearly identical to answer-only AR prediction while still activating the reasoning pathways learned during training. This single parallel prefill pass is what "one-step" latent reasoning refers to.

3.2 Three-Stage Training Pipeline

Training OneVL presents a unique optimization challenge. The main VLM, the language auxiliary decoder, and the visual auxiliary decoder must all be jointly optimized, yet they have fundamentally different learning objectives and start from different relative states of alignment. We address this challenge through a training pipeline consisting of a preliminary self-supervised pretraining step followed by three main stages, each with a clear purpose. The configuration is summarized in Table 11.

Preliminary: Visual Auxiliary Decoder Self-Supervised Pretraining.

Before integrating the visual auxiliary decoder into the full OneVL pipeline, we pretrain it independently as a future-frame generator. Given the current-frame ViT embeddings, the decoder learns to predict the next two frames without any latent conditioning that analogous to self-supervised video

prediction. This establishes a robust visual prior, after which visual latent tokens $\mathcal{H}_v$ are introduced as conditional signals alongside ViT embeddings, transitioning the model from unconditioned frame prediction to ego-state-conditioned rollouts and grounding $\mathcal{H}_v$ as a compact, actionable representation that steers the world model's dynamics. ("Conditioned" refers to the ego-state/intent fields encoded into $\mathcal{H}_v$, not to the output trajectory $\hat{\mathcal{T}}_y$.)

Using the same concatenated target $\mathcal{T}_{y_v} = [\mathcal{T}_{y_v,1},\ \mathcal{T}_{y_v,2}]$, the loss is:

$$\mathcal{L}_p = -\sum_{t=1}^{|\mathcal{T}_{y_v}|} \log P_{\mathcal{D}_v}(\mathcal{T}_{y_v,t} \mid \mathcal{V},\ \mathcal{T}_{y_v,<t}) \ . \tag{7}$$

Stage 0: Main Model Warmup. The prerequisite for auxiliary decoders to provide meaningful supervision is that the main model's latent tokens (*i.e.*, <|latent-vis|> or <|latent|>) carry information that is semantically aligned with reasoning content. Without a targeted warmup, these tokens would not produce meaningful hidden states that auxiliary decoders can decode. Stage 0 addresses this by training the main VLM end-to-end solely on the trajectory prediction task, with latent tokens embedded in each training sample's assistant response. Thus, the model can develop meaningful latent representations by contextualizing the latent tokens within the prompt-response structure alongside trajectory targets. The model naturally learns to use the latent positions to encode intermediate representations useful for trajectory prediction. Besides, the attention mechanism allows trajectory tokens to attend to latent token positions, establishing the information routing pathways that the decoders will later exploit.

Stage 1: Auxiliary Decoder Warmup. With the main model producing stable, meaningful latent representations, Stage 1 focuses exclusively on training the auxiliary decoders to align with these representations. We freeze the main model during this stage, which ensures that the auxiliary decoders optimize against a consistent semantic distribution. To conclude, Stage 1 trains a Language auxiliary decoder $\mathcal{D}_l$ to decode the language CoT reasoning text with $\mathcal{L}_l$ against the ground-truth reasoning annotations and a Visual auxiliary decoder $\mathcal{D}_v$ to predict two future frames with $\mathcal{L}_v$.

Stage 2: Joint End-to-End Fine-Tuning. Finally, Stage 2 jointly fine-tunes all three model components with the combined loss $\mathcal{L}$ (Eq. 5). The gradients from $\mathcal{L}_l$ and $\mathcal{L}_v$ now flow back into the main model, directly shaping the latent representations to simultaneously serve trajectory prediction, language explanation, and visual prediction objectives. This joint optimization is possible in Stage 2 precisely because both the main model and the auxiliary decoders are already well-initialized. Ablation studies (Section 4.3) show that skipping staged training leads to degraded performance, concentrated in Stage 0.

4 Experiments

4.1 Experimental Setups

Datasets. We evaluate on four benchmarks. **NAVSIM** [10], a large-scale planning benchmark from nuPlan [25]; **ROADWork** [15], targeting navigation in road construction zones with temporary signage, cones, and non-standard lanes. **Impromptu** [8], a VLA benchmark covering unstructured corner-case scenarios distilled from eight open driving datasets and **Alpamayo-R1** [48], featuring Chain-of-Causation annotations for interpretable driving. CoT annotations are sourced from AdaThinkDrive [35] for NAVSIM. For ROADWork and Impromptu, we built an in-house pipeline to construct CoT annotations. For Alpamayo-R1, we reproduced the CoC annotation via released checkpoints (details in Appendix H).

Metrics. On NAVSIM, we use the **PDM-score** (Predictive Driver Model), a composite metric assessing trajectory safety, comfort, and progress. On ROADWork, Impromptu, and Alpamayo-R1, we report **ADE** (Average Displacement Error) and **FDE** (Final Displacement Error). We additionally report average inference **latency** across all benchmarks.

Baselines. All baselines are built on Qwen3-VL-4B-Instruct [3]. *AR baselines*: **AR Answer** (direct trajectory prediction, latency lower bound) and **AR CoT+Answer** (full CoT then trajectory, performance upper bound for explicit reasoning). *Latent CoT baselines*: **COCONUT** [19], **CODI** [39], and **SIM-CoT** [49], adapted for VLA-based driving. We also compare against prior SOTA methods under the *supervised-finetuning* setting, including **AdaThinkDrive** [35] and **LaST-VLA** [36] on

Table 1: **NAVSIM** results on PDM-score (higher is better). *Result from original papers.

Method	Size	PDM ↑	Latency (s) ↓	Interpretability
Previous State-of-the-Art				
○ AdaThinkDrive [35]	8B	86.20*	–	Lang.
○ LaST-VLA [36]	8B	87.30*	–	–
AR Baselines (4B, Qwen3-VL)				
○ AR Answer	4B	87.47	<u>4.49</u>	–
○ AR CoT+Answer	4B	<u>88.29</u>	6.58	Lang.
Latent CoT Baselines (4B, Qwen3-VL)				
○ COCONUT [19]	4B	84.84	5.93	–
○ CODI [39]	4B	83.92	8.62	–
○ SIM-CoT [49]	4B	84.21	10.86	Lang.
• **OneVL**	4B	**88.84**	**4.46**	Vis. + Lang.

Table 2: **ROADWork** results on ADE & FDE in pixels (lower is better). *Result from original papers.

Method	Size	ADE (px) ↓	FDE (px) ↓	Latency (s) ↓	Interpretability
Previous State-of-the-Art					
○ YNet [15]	–	22.68*	80.78*	–	–
AR Baselines (4B, Qwen3-VL)					
○ AR Answer	4B	15.98	40.29	<u>4.74</u>	–
○ AR CoT+Answer	4B	<u>13.18</u>	<u>29.98</u>	10.74	Lang.
Latent CoT Baselines (4B, Qwen3-VL)					
○ COCONUT [19]	4B	15.44	38.60	6.06	–
○ CODI [39]	4B	16.45	44.28	6.73	–
○ SIM-CoT [49]	4B	16.49	44.32	6.19	Lang.
• **OneVL**	4B	**12.49**	**28.80**	**4.71**	Vis. + Lang.

NAVSIM; **YNet** [15] on ROADWork; **Impromptu VLA** [8] on Impromptu; and **Cosmos-Reason** on Alpamayo-R1.

4.2 Main Results

OneVL achieves state-of-the-art across all benchmarks. Tables 1–3 present results across all four benchmarks. Figure 1 provides a visual overview of accuracy and latency. We can observe that on **NAVSIM**, OneVL achieves 88.84 PDM-score, surpassing the previous SOTA AdaThinkDrive (86.20) and LaST-VLA (87.30) by +2.64 and +1.54, respectively. On **ROADWork**, OneVL achieves 12.49 / 28.80 ADE/FDE, improving substantially over YNet (22.68 / 80.78). On **Impromptu**, OneVL reaches 1.34 / 3.70 ADE/FDE, outperforming Impromptu-VLA (1.60 / 4.28). On **Alpamayo-R1**, OneVL achieves 2.62 / 7.53 ADE/FDE, competitive with Cosmos-Reason (2.86 / 7.42) despite using a 4B (OneVL, ours) *vs*. 10B (Cosmos-Reason) in model sizes.

Moreover, OneVL outperforms not only all latent CoT baselines but also the explicit AR CoT+Answer baseline across all benchmarks; to our knowledge, it is the first latent CoT method to do so consistently across four driving benchmarks (see Section 2 for a discussion of concurrent work [43]). Three-seed retraining on the close-margin benchmarks confirms the gap over explicit CoT (88.81 ± 0.12 *vs*. 88.07 ± 0.20 PDM on NAVSIM and 1.35 ± 0.02 *vs*. 1.44 ± 0.02 ADE on Impromptu; Section C). The qualitative examples are shown in Figures 4 and 5. Due to page limits, additional visualizations are placed in **Appendix** A.

Prefill inference matches answer-only speed. On NAVSIM, OneVL achieves 4.46s latency, essentially identical to AR Answer (4.49s) and 1.5× faster than AR CoT+Answer (6.58s). Similar speedups hold on ROADWork (2.3× faster than explicit CoT), as well as on Impromptu, and Alpamayo-R1. By prefilling latent tokens in a single parallel pass, we incur negligible overhead compared to sequential autoregressive generation, resulting in competitive inference latency for answer-only prediction while significantly outperforming explicit CoT methods in speed. Note that

Table 3: **Impromptu** and **Alpamayo-R1** results. Both benchmarks report ADE and FDE in meters (lower is better). OneVL achieves state-of-the-art on both metrics compared to existing models.

Method	Size	Impromptu			Alpamayo-R1			Interpretability
		ADE ↓	FDE ↓	Lat.(s) ↓	ADE ↓	FDE ↓	Lat.(s)	
Previous State-of-the-Art								
○ Impromptu VLA [8]	3B	1.60	4.28	6.10	–	–	–	–
○ Cosmos-Reason [48]	10B	–	–	–	2.86	**7.42**	**1.71**	Lang.
AR Baselines (4B, Qwen3-VL)								
○ AR Answer	4B	1.46	4.03	4.24	3.27	9.59	3.06	–
○ AR CoT+Answer	4B	1.42	3.96	6.84	2.99	8.54	3.51	Lang.
Latent CoT Baselines (4B, Qwen3-VL)								
○ COCONUT [19]	4B	1.49	4.07	5.27	3.29	9.48	3.76	–
○ CODI [39]	4B	1.86	5.24	5.18	3.22	9.25	3.85	–
○ SIM-CoT [49]	4B	2.43	5.09	6.10	3.40	9.85	3.78	Lang.
• **OneVL**	4B	**1.34**	**3.70**	**4.02**	**2.62**	7.53	3.23	Vis. + Lang.

Table 4: **(a)** Ablations on the **NAVSIM** benchmark (PDM-score). In the *Staged* column, ✓ denotes the full three-stage recipe; "0+2" / "1+2" denote step-matched variants that skip Stage 1 / Stage 0, respectively; "cur." denotes replacing the future-frame target with current-frame reconstruction. **(b)** Deployment variants on NAVSIM; latency is reported here, where the autoregressive (AR) trajectory head and the MLP regression head are compared directly.

(a) Variant	Lang.	Vis.	Staged	PDM ↑
○ *w/o* vis. decoder	✓	–	✓	87.97
○ *w/o* lang. decoder	–	✓	✓	88.53
○ vis. decoder w/ current-frame target	✓	cur.	✓	88.14
○ *w/o* staged training	✓	✓	–	67.13
○ skip Stage 0 (main-model warmup)	✓	✓	1+2	67.28
○ skip Stage 1 (decoder warmup)	✓	✓	0+2	86.84
• **OneVL** (full)	✓	✓	✓	**88.84**

(b) Variant	Traj. head	PDM ↑	Latency (s) ↓
○ AR CoT+Answer	AR	88.29	6.58
• **OneVL** (full)	AR	**88.84**	4.46
• **OneVL** (MLP regression)	MLP	86.83	**0.24**

on Alpamayo-R1, Cosmos-Reason uses a diffusion-based sampler in a different generation regime, resulting in a lower inference latency that is not directly comparable to autoregressive decoding.

Prior latent CoT methods fail on driving. We observe that COCONUT, CODI, and SIM-CoT all perform substantially worse than the explicit CoT method, or even the answer-only AR prediction. We believe that this is because these methods compress language descriptions, which are a symbolic abstraction, rather than scene dynamics. Without visual world model supervision and staged training, their latent representations collapse into degenerate forms that are insufficient for geometric reasoning.

4.3 Ablation Study

Table 4(a) isolates each component's contribution to NAVSIM. Overall, the visual auxiliary decoder contributes $+0.87$ PDM-score while the language auxiliary decoder adds merely $+0.31$, confirming that world model supervision, which encodes anticipated future scene content, provides complementary benefit beyond language reasoning. Latent-removal and current-frame-reconstruction controls (Sections D and E) show that this gain is carried by the latents and comes from predicting *change* rather than appearance (the current-frame target reaches only 88.14).

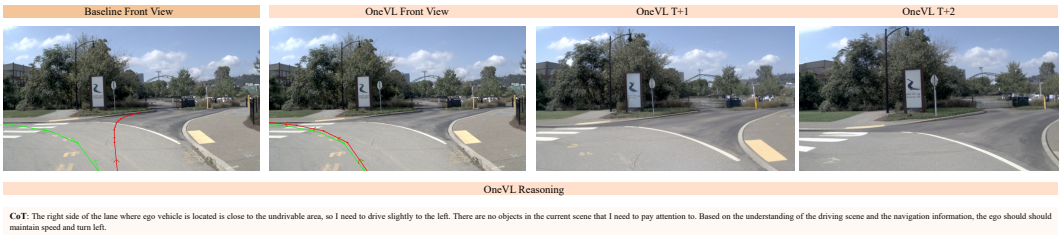


Figure 4: Visualizations of predictions on **NAVSIM**. Each plot overlays ground truth (green) and predicted (red) trajectories on the front camera view. Additional examples are in **Appendix**.



Figure 5: Visualizations of predictions on **ROADWork**. Additional examples are in **Appendix**.

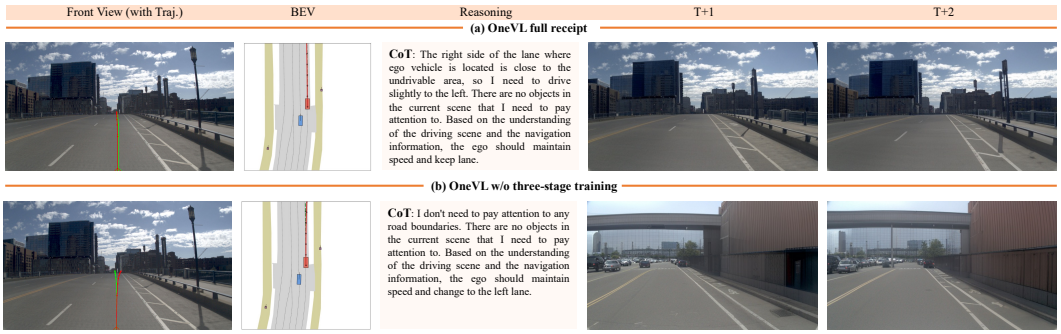


Figure 6: Visual CoT under full training *vs.* an overfitting visual auxiliary decoder. **(a)** Top row. With the full training recipe, latent visual tokens decode to future frames that remain scene-consistent and usable as spatial-temporal reasoning supervision. **(b)** Bottom row. Without three-stage training, decoding from the same input collapses to memorized artifacts rather than generalizing.

Why visual supervision helps more than language. The visual decoder's $+0.87$ contribution *vs.* language's $+0.31$ reflects the importance of world model supervision. Trajectory prediction is fundamentally a spatial task, and future-frame prediction is inherently aligned with geometric reasoning. Language CoT describes the scene in abstract, symbolic terms, which is valuable for semantic grounding instead of the physical dynamics that drive trajectory outcomes.

On the efficacy of three-stage training. We observe that the three-stage training is essential. Skipping it causes a catastrophic 21.71-point drop ($88.84 \rightarrow 67.13$). To find out the reason, we screened the training procedure and discovered that direct end-to-end training suffers from gradient shock (norm 378.22 *vs.* stable 0.28). Second, the end-to-end method causes catastrophic task interference. The backbone struggles to optimize conflicting objectives simultaneously, resulting in a much higher final trajectory prediction loss (0.186 *vs.* 0.136), showing that the model converges into a local minimum and consequently has poor driving performance. A step-matched ablation (Table 4) shows the collapse is concentrated in Stage 0, since skipping it yields 67.28 (reproducing the full collapse) whereas skipping Stage 1 costs only 2.00 PDM; that is, Stage 0 makes the latents decodable

while Stage 1 mainly initializes the decoders. Accordingly, the advantage of our language-only variant over SIM-CoT stems jointly from the recipe, latent-token construction, and backbone fine-tuning.

Furthermore, we evaluate the visual quality of future frame predictions in Figure 6. The full OneVL recipe (a) generates spatially coherent previews for $t+1$ and $t+2$, aligning with plausible ego motion and scene layout. In contrast, the ablated model (b) produces irrelevant frames and erroneous language reasoning. Training dynamics further support this. The full training receipt shows a stable visual explain loss ($\mathcal{L}_v$), while the ablation exhibits an early spike, indicating overfitting and memorization rather than learning real-world dynamics. Consequently, the visual decoder fails to provide useful supervision, introducing noise that degrades final trajectory performance. These results demonstrate that the three-stage curriculum is essential for learning generalizable scene dynamics, which suggests that the quality of compression dictates the quality of intelligence.

Real-World Deployment. Appending a compact MLP regression head on the same backbone yields 86.83 PDM-score at 0.24s latency (≈4.16 Hz), just 5.4% of the latency of full OneVL with autoregressive trajectory decoding (Table 4(b)). Few-Hz operation thus currently costs 2.01 PDM via the MLP head; our headline result remains the accuracy–latency point at answer-only AR speed. Due to page limits, additional details are placed in **Appendix** I.

4.4 Quality of Visual Explanations

Beyond trajectory accuracy, OneVL provides human-interpretable explanations in both language and vision. We evaluate text CoT quality on 500 NAVSIM test samples using three metrics, namely Meta Action Accuracy (exact match of predicted driving decision), Semantic Textual Similarity (cross-encoder score [5]), and LLM-as-Judge Score (gemini-3.1-flash-lite-preview evaluation on perception, motion, and decision correctness). OneVL achieves 71.00 / 78.26 / 79.13 across these metrics, consistently outperforming SIM-CoT (67.20 / 76.25 / 78.73) and approaching the AR CoT+Answer upper bound (73.20 / 79.75 / 81.86). Due to page limits, the full results and the definition of different evaluation metrics used in this work are detailed in **Appendix** B.

5 Conclusion

We presented **OneVL**, a latent CoT framework for autonomous driving built on the hypothesis that *compression drives generalization*. By identifying that prior latent CoT methods fail because they compress language rather than scene dynamics, we introduced a visual world model decoder that predicts future-frame visual tokens alongside a language decoder, forcing the latent bottleneck to encode both semantic intent and anticipated scene evolution. A three-stage training pipeline ensures stable optimization, and prefill inference matches answer-only latency. OneVL achieves superior performance results on NAVSIM, ROADWork, Impromptu, and Alpamayo-R1, surpassing explicit autoregressive CoT on every benchmark.

Potential Limitations. The system requires 3× memory during training (three model instances), mitigated by ZeRO-2, but is still nontrivial. The latent token count and prediction horizons were chosen empirically (Section G), which is left for future work. Our future-frame objective is *predictive* (conditioned on the scene and ego intent, not on the selected trajectory); without action-conditioned or counterfactual rollouts, we do not claim causality in an interventional sense. Finally, few-Hz deployment currently requires the MLP head (−2.01 PDM), and closed-loop evaluation remains future work. Additional analyses are placed in the **Appendix**.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and the area chair for their constructive feedback, which helped us clarify our claims and add several analyses to this version. This work was supported by Xiaomi EV (Xiaomi Embodied Intelligence Team). The authors are employees of Xiaomi, which develops autonomous driving products; they declare no other competing interests.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[2] Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

[3] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[4] Hengwei Bian, Lingdong Kong, Haozhe Xie, Liang Pan, Yu Qiao, and Ziwei Liu. DynamicCity: Large-scale 4D occupancy generation from dynamic scenes. In *International Conference on Learning Representations*, 2025.

[5] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

[6] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with LLMs: Fusing object-level vector modality for explainable autonomous driving. In *IEEE International Conference on Robotics and Automation*, pages 14093–14100, 2024.

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

[8] Haohan Chi, Huan ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, Leichen Wang, Xingtao Hu, Hao Sun, Hang Zhao, and Hao Zhao. Impromptu VLA: Open weights and open data for driving vision-language-action models. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 38, 2025.

[9] Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, Wenxuan Wang, et al. Emu3. 5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

[10] 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.

[11] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al. Language modeling is compression. *arXiv preprint arXiv:2309.10668*, 2023.

[12] Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024.

[13] Yifei Dong, Fengyi Wu, Yilong Dai, Lingdong Kong, Guangyu Chen, Xu Zhu, Qiyu Hu, Tianyu Wang, Johnalbert Garnica, Feng Liu, et al. Language-conditioned world modeling for visual navigation. *arXiv preprint arXiv:2603.26741*, 2026.

[14] Jiaheng Geng, Jiatong Du, Xinyu Zhang, Ye Li, Panqu Wang, and Yanjun Huang. Driving in corner case: A real-world adversarial closed-loop evaluation platform for end-to-end autonomous driving. *arXiv preprint arXiv:2512.16055*, 2025.

[15] Anurag Ghosh, Shen Zheng, Robert Tamburo, Khiem Vuong, Juan Alvarez-Padilla, Hailiang Zhu, Michael Cardei, Nicholas Dunn, Christoph Mertz, and Srinivasa G. Narasimhan. ROAD-Work: A dataset and benchmark for learning to recognize, observe, analyze and drive through work zones. In *IEEE/CVF International Conference on Computer Vision*, pages 6132–6142, 2025.

[16] Google. Gemini 2.5 Pro preview: even better coding performance. https://developers.googleblog.com/en/gemini-2-5-pro-io-improved-coding-performance, 2025.

[17] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[18] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[19] 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.

[20] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[21] Tianshuai Hu, Zeying Gong, Lingdong Kong, Xiaodong Mei, Yiyi Ding, Qi Zeng, Ao Liang, Rong Li, Yangyi Zhong, and Junwei Liang. NavThinker: Action-conditioned world models for coupled prediction and planning in social navigation. *arXiv preprint arXiv:2603.15359*, 2026.

[22] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[23] Ayesha Ishaq, Jean Lahoud, Ketan More, Omkar Thawakar, Ritesh Thawkar, Dinura Dissanayake, Noor Ahsan, Yuhao Li, Fahad Shahbaz Khan, Hisham Cholakkal, Ivan Laptev, Rao Muhammad Anwer, and Salman Khan. DriveLMM-o1: A step-by-step reasoning dataset and large multimodal model for driving scenario understanding. *arXiv preprint arXiv:2503.10621*, 2025.

[24] Haoqiang Kang, Yizhe Zhang, Nikki Lijing Kuang, Nicklas Majamaki, Navdeep Jaitly, Yi-An Ma, and Lianhui Qin. LaDiR: Latent diffusion enhances LLMs for text reasoning. *arXiv preprint arXiv:2510.04573*, 2025.

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

[26] Yann LeCun. A path towards autonomous machine intelligence. Technical report, Meta AI, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf.

[27] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4):391–444, 2007.

[28] Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *International Conference on Learning Representations*, 2025.

[29] Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via BEV world model. In *IEEE/CVF International Conference on Computer Vision*, pages 27137–27146, 2025.

[30] Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-W0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, 2026.

[31] Alan Liang, Youquan Liu, Yu Yang, Dongyue Lu, Linfeng Li, Lingdong Kong, Huaici Zhao, and Wei Tsang Ooi. LiDARCrafter: Dynamic 4D world modeling from lidar sequences. *AAAI Conference on Artificial Intelligence*, 40(22):18406–18414, 2026.

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916, 2023.

[33] Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, Junqiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yadan Luo. GuideFlow: Constraint-guided flow matching for planning in end-to-end autonomous driving. *arXiv preprint arXiv:2511.18729*, 2025.

[34] Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, and Long Chen. DriveWorld-VLA: Unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026.

[35] 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.

[36] Yuechen Luo, Fang Li, Shaoqing Xu, Yang Ji, Zehan Zhang, Bing Wang, Yuannan Shen, Jianwei Cui, Long Chen, Guang Chen, et al. Last-vla: Thinking in latent spatio-temporal space for vision-language-action in autonomous driving. *arXiv preprint arXiv:2603.01928*, 2026.

[37] Ziang Luo, Kangan Qian, Jiahua Wang, Yuechen Luo, Jinyu Miao, Zheng Fu, Yunlong Wang, Sicong Jiang, Zilin Huang, Yifei Hu, Yuhao Yang, Hao Ye, Mengmeng Yang, Xiaojian Dong, Kun Jiang, and Diange Yang. MTRDrive: Memory-tool synergistic reasoning for robust autonomous driving in corner cases. *arXiv preprint arXiv:2509.20843*, 2025.

[38] Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, et al. DynVLA: Learning world dynamics for action reasoning in autonomous driving. *arXiv preprint arXiv:2603.11041*, 2026.

[39] Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Conference on Empirical Methods in Natural Language Processing*, pages 677–693, 2025.

[40] Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable image tokenization with index backpropagation quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16037–16046, 2025.

[41] 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.

[42] Dijia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv preprint arXiv:2502.03275*, 2025.

[43] Shuhan Tan, Kashyap Chitta, Yuxiao Chen, Ran Tian, Yurong You, Yan Wang, Wenjie Luo, Yulong Cao, Philipp Krähenbühl, Marco Pavone, and Boris Ivanovic. Latent chain-of-thought world modeling for end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[44] Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *arXiv preprint arXiv:2512.24497*, 2025.

[45] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop*, pages 1–5. IEEE, 2015.

[46] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[47] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5294–5306, 2025.

[48] Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, Liang Feng, Greg Heinrich, Jack Huang, Peter Karkus, Boyi Li, Pinyi Li, Tsung-Yi Lin, Dongran Liu, Ming-Yu Liu, Langechuan Liu, Zhijian Liu, Jason Lu, Yunxiang Mao, Pavlo Molchanov, Lindsey Pavao, Zhenghao Peng, Mike Ranzinger, Ed Schmerling, Shida Shen, Yunfei Shi, Sarah Tariq, Ran Tian, Tilman Wekel, Xinshuo Weng, Tianjun Xiao, Eric Yang, Xiaodong Yang, Yurong You, Xiaohui Zeng, Wenyuan Zhang, Boris Ivanovic, and Marco Pavone. Alpamayo-R1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

[49] 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.

[50] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Benchmarking and improving bird's eye view perception robustness in autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3878–3894, 2025.

[51] Shaoqing Xu, Fang Li, Shengyin Jiang, Ziying Song, Li Liu, and Zhi-xin Yang. Gaussian-Pretrain: A simple unified 3D Gaussian representation for visual pre-training in autonomous driving. *arXiv preprint arXiv:2411.12452*, 2024.

[52] Xiang Xu, Ao Liang, Youquan Liu, Linfeng Li, Lingdong Kong, Ziwei Liu, and Qingshan Liu. U4D: Uncertainty-aware 4D world modeling from LiDAR sequences. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[53] Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Annual Meeting of the Association for Computational Linguistics*, pages 23336–23351, 2025.

[54] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. DriveGPT4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024.

[55] Tianyi Yan, Tao Tang, Xingtai Gui, Yongkang Li, Jiasen Zheng, Weiyao Huang, Lingdong Kong, Wencheng Han, Xia Zhou, Xueyang Zhang, et al. AD-R1: Closed-loop reinforcement learning for end-to-end autonomous driving with impartial world models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[56] Lingfeng Zhang, Xiaoshuai Hao, Yingbo Tang, Haoxiang Fu, Xinyu Zheng, Pengwei Wang, Zhongyuan Wang, Wenbo Ding, and Shanghang Zhang. $nava^3$: Understanding any instruction, navigating anywhere, finding anything. *arXiv preprint arXiv:2508.04598*, 2025.

[57] Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Shengyin Jiang, Long Chen, Zhi-Xin Yang, and Jiwen Lu. Dvgt: Driving visual geometry transformer. *arXiv preprint arXiv:2512.16919*, 2025.

[58] Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Hanbing Li, Long Chen, Zhi-Xin Yang, and Jiwen Lu. Dvgt-2: Vision-geometry-action model for autonomous driving at scale. *arXiv preprint arXiv:2604.00813*, 2026.

Appendix

A Qualitative Results **15**

B Explanation Quality **15**

B.1 Text CoT Quality 15

B.2 LLM-as-Judge Evaluation Prompt 18

C Seed Variance on Close-Margin Benchmarks **19**

D Are the Latent Tokens Load-Bearing? **20**

E Future-Frame Prediction Quality **21**

F Impromptu L2 Error **22**

G Training Configuration **22**

H CoT Annotation Construction **22**

I Towards Real-World Deployment **24**

A Qualitative Results

Beyond trajectory prediction accuracy, OneVL is unique in providing human-interpretable explanations in both language and vision. Figure 7, Figure 8, Figure 9 and Figure 10 provide NAVSIM, ROADWork, Impromptu and Alpamayo-R1 qualitative examples, including a side-by-side trajectory prediction comparison (*vs.* AR Answer baseline), two future frames from the visual auxiliary decoder, and the text CoT from the language auxiliary decoder.

B Explanation Quality

We also quantitatively evaluate the quality of text explanations.

B.1 Text CoT Quality

We evaluate the language explanations produced by the language auxiliary decoder against the ground-truth CoT annotations on the 500 NAVSIM test set. We compare against the AR CoT+Answer baseline (which generates explicit CoT autoregressively) and SIM-CoT (the only prior latent CoT method with language interpretability). We report three complementary metrics that capture different dimensions of explanation quality:

- **Meta Action Accuracy**: Following Luo et al. [37], we report the meta action accuracy. On NAVSIM, each CoT concludes with a high-level driving decision (*e.g.*, "the ego should maintain speed and keep lane"). We extract this meta-action clause from both the ground-truth and the predicted CoT and compute exact string match accuracy. This metric directly measures whether the model's reasoning arrives at the correct driving intent, which is the most safety-critical aspect of CoT quality.
- **STS Score (Semantic Textual Similarity Score)**: We compute a neural semantic similarity score between each predicted CoT and its ground-truth reference using a cross-encoder reranker (BGE-reranker-v2-m3 [5]). This evaluator is particularly suitable for computing a similarity score for templated CoTs where most of the words are identical. A cross-encoder concatenates the ground truth and the prediction, processing them simultaneously through full token-by-token

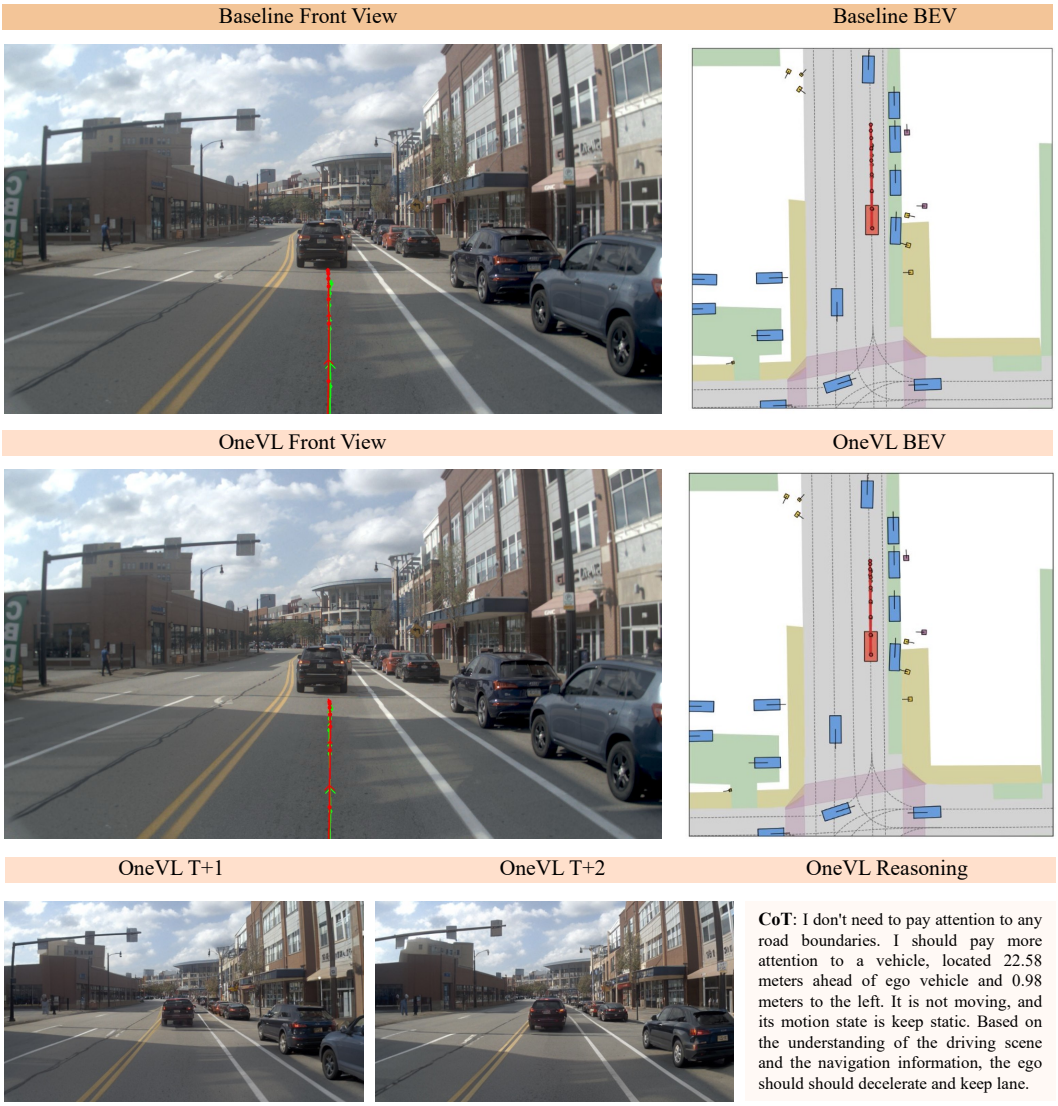
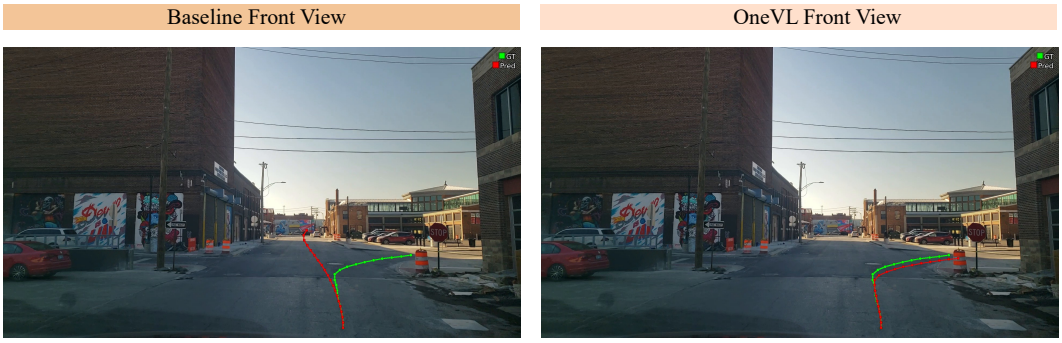


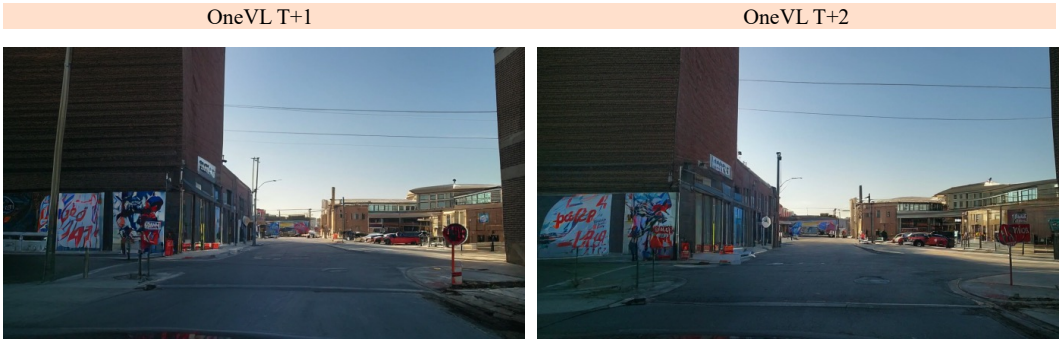
Figure 7: Visualizations of prediction on NAVSIM. Each plot overlays ground-truth (green) and predicted (red) trajectories on the front camera view.

cross-attention. This mechanism allows the model to perform a deep, comparative analysis, making it highly sensitive to critical localized contradictions—such as predicting "slowly" instead of "fastly", or outputting hazardous numerical distances. For each ground-truth-prediction pair, the cross-encoder outputs a raw relevance logit. We apply global min-max normalization across all methods and examples to obtain scores in $[0, 1]$. This metric captures fine-grained semantic alignment beyond surface-level n-gram overlap.

- **LLM-as-Judge Score**: Following Ishaq et al. [23], Luo et al. [37], we also employ a state-of-the-art proprietary VLM, gemini-3.1-flash-lite-preview, as an automated evaluator. Given the front-camera image, the ground-truth CoT, and the predicted CoT, the judge model scores each prediction on a 0-100 scale based on four criteria: (1) perception accuracy; (2) motion state prediction; (3) ego decision correctness, and (4) language fluency. Scores are normalized to $[0, 1]$. This metric provides a holistic evaluation that accounts for both visual grounding and reasoning quality. The full evaluation prompt is provided in Appendix B.2.

Quantitative results are shown in Table 5. OneVL consistently outperforms SIM-CoT across all evaluation dimensions. Specifically, OneVL achieves a Meta Action Accuracy of 71.00, a significant





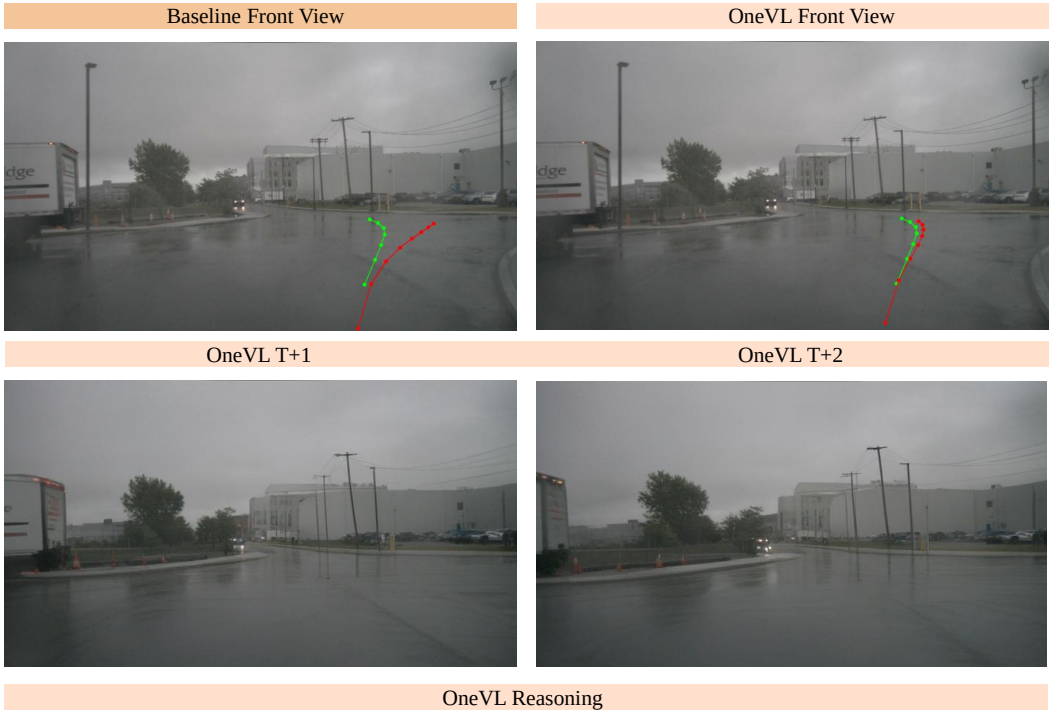
OneVL Reasoning

CoT: <critical_factors>Orange construction barrels/cones along the right edge of the roadway narrow/define the usable lane and encourage the ego vehicle to keep left of them while proceeding forward..
A stop sign at the right-side corner of the upcoming intersection requires the ego vehicle to approach cautiously and prepare to stop/yield at the intersection.
Parked vehicles along the right curb near the intersection reduce lateral clearance and constrain the approach path, reinforcing a centered/leftward trajectory.</critical_factors>
<cause>Stop at the control point while turning right because the stop sign at the upcoming intersection requires a controlled approach and full stop.</cause>
<decision>longitudinal:Stop at control point lateral:Turn right</decision>

Figure 8: Visualizations of prediction on ROADWork. Each plot overlays ground-truth (green) and predicted (red) trajectories on the front camera view.

3.8 improvement over SIM-CoT (67.20). This indicates that our latent representation more effectively captures and decodes the critical high-level driving intents. In terms of semantic alignment, OneVL reaches an STS score of 78.26 and an LLM Judge score of 79.13, narrowing the gap to the autoregressive baseline (AR CoT+Answer). While the AR model maintains the highest scores across all metrics, it requires explicit sequential text generation. This results in higher inference latency as demonstrated in Table 1. In contrast, OneVL achieves competitive explanation quality while benefiting from the efficiency of a latent reasoning framework. These results demonstrate that the language auxiliary decoder in OneVL successfully enables faithful, human-readable explanations of the model's reasoning.

Alignment between decoded reasoning and predicted trajectories. We further inspected cases where the decoded CoT contains errors (*e.g.*, the examples in Figures 4 and 5) to understand how such errors relate to the predicted trajectory. We observe that the effect depends on the error type. Errors in scene-description details (*e.g.*, a misjudged parked car or a miscounted cone) have a small impact, as they alter the narrative content of the explanation without changing the planned motion. In contrast, errors in the meta-action (*e.g.*, turn left *vs*. right, speed up *vs*. slow down) significantly affect the trajectory, since the meta-action directly determines the maneuver. This matches the design intent of the latent bottleneck, in which the meta-action is the decision-level content the latents must carry, whereas fine-grained descriptions are largely explanatory. The decoded reasoning is therefore aligned with the trajectory where it matters most—decision errors propagate, cosmetic ones do not—which is also consistent with Meta Action Accuracy being the metric on which OneVL is closest to the explicit CoT upper bound.



CoT: <decision>longitudinal decision: keep speed, lateral decision: straight</decision>\n<cause> The decision to maintain current speed and go straight is likely due to the presence of a large building and parked cars on the left, which would obstruct a left turn. Additionally, the wet road conditions suggest caution, making a straight path the safer option.</cause>

Figure 9: Visualizations of prediction on Impromptu. Each plot overlays ground-truth (green) and predicted (red) trajectories on the front camera view.

Table 5: Text CoT quality on the **NAVSIM** test set (500 samples). *Meta Act.* measures exact match of the predicted driving decision. *STS* reports mean semantic similarity (cross-encoder, normalized). *LLM Judge* reports VLM evaluation score (normalized).

Method	Meta Act. ↑	STS ↑	LLM Judge ↑	Avg. ↑
○ AR CoT+Answer	**73.20**	**79.75**	**81.86**	**78.27**
○ SIM-CoT	67.20	76.25	78.73	74.06
• **OneVL** (lang. aux.)	<u>71.00</u>	<u>78.26</u>	<u>79.13</u>	<u>76.13</u>

B.2 LLM-as-Judge Evaluation Prompt

We employ a state-of-the-art proprietary VLM, gemini-3.1-flash-lite-preview[1] as an automated evaluator for text CoT quality. For each sample, the judge receives the front-camera image alongside the following prompt (with the ground-truth and predicted CoT filled in):

[1] https://ai.google.dev/gemini-api/docs/models/gemini-3.1-flash-live-preview



Figure 10: Visualizations of prediction on Alpamayo-R1. Each plot overlays ground-truth (green) and predicted (red) trajectories on the front camera view.

You are an expert evaluator for Autonomous Driving Systems.
Your task is to evaluate the quality of a predicted driving Chain-of-Thought (CoT) against the Ground Truth CoT. You are also provided with the front-camera image for driving context.

Ground Truth CoT: "{gt_cot}"
Predicted CoT: "{pred_cot}"

Evaluation Criteria:
1. Perception (Distance & Location): Minor numerical differences (e.g., 29.3m vs 27.5m) are acceptable and should only incur slight penalties.
2. Motion State (Prediction): Identifying the correct state (e.g., 'moving fastly', 'moving slowly', 'keep static') is highly critical. A mismatch here is a severe safety hazard and must heavily reduce the score.
3. Ego Decision (Planning): The final action (e.g., 'accelerate', 'decelerate', 'maintain speed', 'keep lane') MUST match the Ground Truth. Any deviation is a critical safety error and should result in a massive penalty.
4. Language Fluency: Penalize minor grammatical errors.

Score the Predicted CoT on a scale of 0 to 100 based on the above criteria.

Output STRICTLY in the following JSON format without any markdown blocks, backticks, or extra text:
{"reasoning": "Briefly explain the penalties applied based on the criteria.", "score": <integer_between_0_and_100>}

The judge's raw score (0–100) is normalized to $[0, 1]$ by dividing by 100. We set the temperature to 0.1 to ensure scoring stability.

C Seed Variance on Close-Margin Benchmarks

The margins of OneVL over explicit CoT are narrow on NAVSIM ($+0.55$ PDM-score) and Impromptu (0.08 m ADE), whereas they are large on ROADWork and Alpamayo-R1. To verify that the close-

Table 6: Three-seed results of OneVL and AR CoT+Answer on the two close-margin benchmarks. PDM-score (higher is better) on **NAVSIM**; ADE/FDE in meters (lower is better) on **Impromptu**.

Benchmark / Metric	**Method**	**Seed 1**	**Seed 2**	**Seed 3**	**Mean $\pm$ std**
NAVSIM / PDM $\uparrow$	AR CoT+Answer	88.29	88.02	87.90	88.07 ± 0.20
	OneVL	88.84	88.68	88.91	$\mathbf{88.81 \pm 0.12}$
Impromptu / ADE $\downarrow$	AR CoT+Answer	1.42	1.46	1.44	1.44 ± 0.02
	OneVL	1.34	1.33	1.37	$\mathbf{1.35 \pm 0.02}$
Impromptu / FDE $\downarrow$	AR CoT+Answer	3.96	4.07	3.99	4.01 ± 0.06
	OneVL	3.70	3.69	3.79	$\mathbf{3.73 \pm 0.06}$

Table 7: Removing latent tokens at inference on **NAVSIM**.

Variant	**OneVL (full)**	*w/o* **lang. latents**	*w/o* **vis. latents**	*w/o* **all latents**
PDM-score $\uparrow$	**88.84**	88.60	88.26	87.97

margin results are not seed artifacts, we retrained both OneVL (full) and AR CoT+Answer with three random seeds on NAVSIM and Impromptu, keeping all other settings identical. Results are reported in Table 6. On both benchmarks the advantage of OneVL is consistent across all three seeds, and the mean gap exceeds one standard deviation with no overlap between the two methods (on NAVSIM, $88.81 - 0.12 = 88.69$ *vs.* $88.07 + 0.20 = 88.27$; on Impromptu, $1.35 + 0.02 = 1.37$ *vs.* $1.44 - 0.02 = 1.42$ ADE, and the FDE gap is larger still). The main-text tables report the seed-1 numbers, which are the same runs used for all other analyses.

D Are the Latent Tokens Load-Bearing?

Since both auxiliary decoders receive the current-frame ViT features $\mathcal{V}$ in addition to the latent hidden states ($\mathcal{H}_l$, $\mathcal{H}_v$; Eqs. 1 and 3), and since the trajectory tokens can attend directly to image and text tokens, one may ask whether the latents are the true information carriers or are bypassed. We find that both $\mathcal{V}$ and the latents are necessary, and that neither alone is sufficient. All experiments are on NAVSIM.

Latent-removal at inference. We remove different types of latent tokens from the prefilled prompt at inference while keeping the trained model fixed (Table 7). Both latent types affect trajectory prediction (visual latents more so, -0.58 *vs.* -0.24), and removing both returns the PDM-score to the no-latent level (87.97). The trajectory head thus consumes the latents rather than routing around them.

Attention distribution during trajectory decoding. On 500 test samples we aggregate the attention mass that trajectory tokens place on each category of prefilled tokens, summed over all layers and heads (excluding the first sink token); see Table 8. The prompt text carries most of the total mass (75.7%), but among the remaining categories the latents receive by far the highest per-token weight ($5.7\times$ that of image tokens for language latents and $4.5\times$ for visual latents), consistent with the latents acting as compact, high-density summaries. The latent attention share is also stable across scenarios ($< 0.2\%$ variation), whereas the image attention share varies widely (14.8%–27.5%), indicating that inference does not bypass the latents in a scenario-dependent manner.

Decoder-input ablation. We retrain a variant in which the decoders receive only the latents ($\mathcal{H}_l$ / $\mathcal{H}_v$) without $\mathcal{V}$. The trajectory PDM-score drops to 86.87, well below full OneVL (88.84), and the decoder outputs collapse into hallucination (text full of errors and frames unrelated to the scene, matching the failure mode in Figure 6b). Without $\mathcal{V}$ to decode jointly against, the decoders produce no useful training signal, the latents receive no shaping gradient, and the trajectory head treats them as noise. Conversely, the "w/o vis. decoder" and current-frame controls in Table 4 show that $\mathcal{V}$ alone does not account for the gain.

Table 8: Attention mass placed by trajectory tokens on each category of prefilled tokens (**NAVSIM**, 500 samples, all layers and heads). *Share/Token* is the mass share divided by the number of tokens, normalized by the prompt-text value.

Category	**#Tokens**	**Mass share**	**Share/Token**
Prompt text	232	0.757	0.732
Image	2040	0.213	0.024
Language latent	20	0.012	0.137
Visual latent	35	0.017	0.108

Table 9: Future-frame prediction quality on **NAVSIM** (500 samples). *GT-Recon* denotes the tokenize-then-decode reconstruction of the ground-truth frame, which serves as the quality ceiling.

Horizon	**Comparison**	**SSIM ↑**	**LPIPS ↓**	**FID ↓**
$+0.5$s	GT *vs*. GT-Recon (ceiling)	0.759 ± 0.064	0.249 ± 0.038	20.91
$+0.5$s	GT *vs*. Prediction	0.641 ± 0.085	0.398 ± 0.081	29.72
$+1.0$s	GT *vs*. GT-Recon (ceiling)	0.759 ± 0.065	0.249 ± 0.037	20.98
$+1.0$s	GT *vs*. Prediction	0.620 ± 0.087	0.446 ± 0.085	32.49

Staged convergence of the visual decoder loss. With $\mathcal{V}$ and the decoder capacity fixed throughout, the visual decoder's cross-entropy loss tightens monotonically as $\mathcal{H}_v$ becomes available and is then shaped, from ≈4–5 in the preliminary pretraining (no $\mathcal{H}_v$, $\mathcal{V}$ only) to ≈0.4 in Stage 1 ($\mathcal{H}_v$ introduced, main model frozen) and ≈0.1 in Stage 2 (joint fine-tuning). This ≈40–50× reduction tracks exactly the points where the latents gain and then exploit access to the ego-state/intent context; it would not occur if the latents were redundant and $\mathcal{V}$ alone drove the prediction. In summary, the latents are not a passive pass-through for $\mathcal{V}$; they are progressively shaped by training-time supervision, but only through the joint $(\mathcal{V}, \mathcal{H})$ coupling.

E Future-Frame Prediction Quality

Quantitative metrics. On 500 held-out NAVSIM samples, we quantify the quality of the predicted future frames at $+0.5$s and $+1.0$s with SSIM, LPIPS, and FID against the ground-truth frames (Table 9). As a reference ceiling we also report the *GT reconstruction*, *i.e*., the ground-truth frame encoded into visual tokens by the IBQ tokenizer and decoded back, which bounds the attainable quality given the information lost by quantization. The predictions sit below the ceiling but in the same regime, and degrade with horizon ($+0.5$s $\rightarrow$ $+1.0$s) as expected. We also measured token exact-match accuracy (1.88% at $+0.5$s, 0.97% at $+1.0$s) but note that it is uninformative in isolation, because with a large codebook minor pixel variation (shadows, lighting, texture) flips token IDs with no visible change, while the decoded frames remain visually plausible (Figure 6).

Predicting change *vs*. reconstructing appearance. To isolate the world-model effect from generic visual regularization, we train a control in which the visual auxiliary decoder reconstructs the *current* frame instead of predicting the $+0.5$s/$+1.0$s frames (the current frame is duplicated so that the number of supervised visual tokens is identical). If the visual decoder merely acted as an appearance regularizer, this control should recover most of the $+0.87$ gain. It does not, as the current-frame control reaches 88.14 PDM-score, compared with 88.84 for full OneVL and 87.97 without any visual decoder (Table 4). Together with the systematic degradation of frame quality with horizon in Table 9 (a static, horizon-agnostic reconstruction would track the flat ceiling), this indicates that the latents are sensitive to how far ahead they predict and that the benefit comes from predicting change rather than reconstructing appearance. We note again that this is evidence for *predictive* temporal structure; establishing interventional causality would require action-conditioned or counterfactual rollouts, which we leave for future work (Section 5).

Table 10: Impromptu L2 error (meters; lower is better). *Result from the checkpoint provided by the original paper.

Method	Traj. Pred. L2 Error (m) 1s ↓	2s ↓	3s ↓	4s ↓	Avg. ↓
○ Impromptu VLA [8]*	0.90	2.80	3.75	5.89	3.16
○ Impromptu VLA [8]	0.14	0.60	1.45	2.67	1.22
• **OneVL**	**0.13**	**0.48**	**1.18**	**2.25**	**1.01**

F Impromptu L2 Error

To reproduce the optimal performance of the baseline model, we follow the official settings to conduct evaluations using both the training script released by Impromptu and the pre-trained checkpoint uploaded to Hugging Face, with the corresponding test results presented in Table 10. Concretely, the first row marked with an asterisk (*) reports the results evaluated on the author-provided checkpoint[2], and the second row shows the outcomes of the model trained from scratch via the official training script. As neither of these results matches the performance claimed in the original paper, we adopt the better-performing one as the representative result of Impromptu for comparative analysis in the main experiments.

G Training Configuration

We begin with self-supervised pretraining of the visual auxiliary decoder, in which the model learns to predict the next frame from ViT features alone, without the autoregressive backbone in the loop. This phase uses 13 040 optimizer steps and a global batch size of 256, rather than a fixed epoch count. We then move to *Stage 0*, which warms up the main VLM while introducing latent tokens, followed by *Stage 1*, where the auxiliary decoders train with the main model frozen and the visual decoder becomes conditioned on the ego state through visual latent tokens. Finally, *Stage 2* performs joint end-to-end fine-tuning with all parts of the stack updated together; details are shown in Table 11. Training is conducted on 32 NVIDIA GPUs. Evaluations of all models were performed on the same NVIDIA GPU.

Data splits. NAVSIM, ROADWork, and Impromptu are used with their existing official train/test splits, and all baselines are trained and evaluated on exactly the same splits as OneVL. For Alpamayo-R1, whose official test set is not released, we subsample 700 examples from the available video clips as a held-out test set that is disjoint from the training examples (see Appendix H). The CoT annotations and data lists are released together with the code to facilitate reproduction.

Sensitivity to latent-token count and prediction horizon. The latent-token counts ($\mathcal{C}_v = 4$, $\mathcal{C}_t = 2$) and the prediction horizons (+0.5s/+1.0s) were chosen empirically to balance accuracy and prefill overhead. In a preliminary sweep over alternative latent-token counts on NAVSIM, the PDM-score varied between approximately 83 and 87, below the chosen configuration (88.84), while larger counts increase the prefill overhead; a systematic sweep over prediction horizons (*e.g.*, 0.3s or 2s) is left for future work.

H CoT Annotation Construction

The chain-of-thought reasoning annotations for the ROADWork dataset are constructed using an in-house annotation pipeline developed specifically for work-zone driving scenarios. Unlike the NAVSIM annotations (sourced from AdaThinkDrive [35]), ROADWork CoT reasoning requires domain-specific annotation of:

- Work-zone hazard identification: Detection and description of cones, barriers, temporary signage, and worker presence.

[2] https://huggingface.co/aaaaaap/ImpromptuVLAModel/tree/main/7B_AD

Table 11: Training hyperparameters for OneVL.

Hyperparameter	**Pre-train**	**Stage 0**	**Stage 1**	**Stage 2**
Steps / Epochs	13 040	2 ep	1 ep	5 ep
Batch (global)	256	64	64	64
Learning rate	1×10^{-4}	4×10^{-5}	1×10^{-4}	1×10^{-4}
LR schedule	Cosine	Cosine	Cosine	Cosine
Optimizer	AdamW	AdamW	AdamW	AdamW
Precision	BF16	BF16	BF16	BF16
Parallelism	ZeRO-2	ZeRO-2	ZeRO-2	ZeRO-2
Trainable	Vis aux dec	ViT, LLM, aligner	Aux decoders	All
Frozen	–	–	Main VLM	–
λ_l	–	–	1.0	1.0
λ_v	1.0	–	0.1	0.1

- Non-standard lane interpretation: Reasoning about temporary lane markings, reduced lane widths, and merge configurations.
- Speed and clearance reasoning: Justification of appropriate speed reduction and lateral clearance decisions specific to work zones.

The CoT reasoning annotations for the Impromptu VLA Dataset are constructed using a VLM-centric annotation pipeline with Chain-of-Thought prompting developed specifically for unstructured driving scenarios (the core corner cases for autonomous driving). Unlike annotations for conventional structured driving datasets, Impromptu VLA CoT reasoning requires domain-specific annotation of:

- Unstructured scenario classification: Identification and categorization of four core challenging unstructured scenarios, including roads with unclear boundaries, temporary traffic rule changes, unconventional dynamic obstacles, and challenging road conditions.
- Complex scene element perception: Detection and description of ambiguous road boundaries, temporary traffic facilities, non-standard dynamic obstacles, and adverse road/environmental conditions.
- Non-standard driving behavior reasoning: Interpretation of temporary traffic rules, obstacle avoidance strategies, and justification of speed adjustment and lateral clearance decisions in unstructured environments.
- Planning-oriented decision reasoning: Generation of ego-vehicle meta-action plans, end-to-end trajectory predictions, and textual rationales for driving maneuvers.

Since the CoT reasoning labels for Alpamayo-R1 are not released, we replicate these annotations ourselves. Specifically, we use the publicly released model checkpoint[3] to reproduce the CoT labels for all training examples. Regarding the waypoint prediction, we applied a heuristic subsampling strategy to reduce the sequence from 64 to 8 points while ensuring the last points are retained, as 64 waypoints are too dense for autoregressive modeling (the original Alpamayo-R1 model employs flow matching). Furthermore, as the original paper does not release the official test set, we subsample 700 examples from the available video clips to construct our test set.

The following are four CoT examples:

NAVSIM Example.

> The right side of the lane where ego vehicle is located is close to the undrivable area, so I need to drive slightly to the left. I should pay more attention to a vehicle, located 18.42 meters ahead of ego vehicle and 0.49 meters to the left. It is driving forward in the same direction, and its motion state is moving fastly. Based on the understanding of the driving scene and the navigation information, the ego should maintain speed and keep lane.

[3] https://huggingface.co/nvidia/Alpamayo-R1-10B

ROADWork Example.

critical_factors: An orange work-zone warning sign placed in the roadway ahead indicates construction activity and prompts caution and adjustment in path. A line of cones along the left side of the roadway creates a temporary channelization/taper that narrows the usable lane and guides vehicles away from the left edge. Parked vehicles tightly line the right curb, reducing lateral clearance and constraining the ego vehicle available path through the work area. Scaffolding/sidewalk construction along the right side suggests active work frontage and encourages keeping distance from the curbside area.
cause:Taper or channelization speed adjustment and channelized lane keeping because cones taper the left side and parked vehicles along the right curb narrow the usable lane.
decision:longitudinal:Taper or channelization speed adjustment lateral:Channelized lane keeping

Impromptu Example.

<decision>longitudinal decision: stop, lateral decision: straight</decision><cause> The decision to stop the car is likely due to the presence of construction barriers and a pedestrian crossing the road, indicating a need to ensure safety and comply with traffic regulations.</cause>

Alpamayo-R1 Example.

Turn right at the intersection since the right-turn traffic light is green.

I Towards Real-World Deployment

Closed-loop on-vehicle testing favors extremely low-latency trajectory prediction. However, autoregressive decoding waypoints still dominate the budget even when latent reasoning is prefilled. To explore a deployment-oriented variant, we append a compact MLP head on top of the same Qwen3-VL-4B-Instruct backbone (using the hidden state of the last latent token as the input). Therefore, the model can autoregressively predict the trajectory using the LLM head as well as predict the trajectory with a single feed-forward pass using the MLP head. This variant can still exploit multimodal latent supervision during training.

Table 4(b) compares this MLP variant against full OneVL on NAVSIM. The MLP head reaches 86.83 PDM-score with 0.24s inference latency, whereas full OneVL keeps 88.84 with 4.46s inference latency, achieving a 5.4% of the AR model's latency. At $1/0.2405 \approx 4.16$ Hz end-to-end, the MLP variant meets a typical on-vehicle few-Hz budget while remaining in a competitive performance relative to strong prior models (*e.g.*, LaST-VLA). We emphasize the trade-off involved, as reaching a few-Hz rate currently requires replacing autoregressive waypoint decoding with the MLP head, which costs 2.01 PDM-score relative to full OneVL. We also note that the lower latency of Cosmos-Reason on Alpamayo-R1 (Table 3) stems from a diffusion-based sampler in a different generation regime and is therefore not directly comparable. The MLP head should thus be regarded as a secondary deployment option rather than as evidence of real-time readiness; closed-loop and on-vehicle evaluation of OneVL is an important direction for future work.

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: We have detailed the contributions accurately in the abstract and introduction

   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: As discussed in Section 5, we have listed some limitations of our work.

   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: This work does not include theoretical results.

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: In Section 4.3, Appendix G and H, we have described the details of implementing and training the proposed model to ensure the reproducibility of our work.

   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: Code, model checkpoints, and the CoT annotations are released at https://github.com/xiaomi-research/OneVL (project page: https://Xiaomi-Embodied-Intelligence.github.io/OneVL).

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: In Section 4.3, Appendix G and H, we have described the details of implementing and training the proposed model to ensure the reproducibility of our work.

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: [Yes]

Justification: For the two benchmarks where the margin over explicit CoT is small (NAVSIM and Impromptu), we retrained OneVL and the AR CoT+Answer baseline with three random seeds and report mean and standard deviation (Section 4.2 and Appendix C); the gap exceeds one standard deviation without overlap. On the remaining benchmarks (ROADWork, Alpamayo-R1), where margins are large, we report single-seed results due to the substantial training cost.

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: We have reported the needed computer resources in Appendix G.

   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 in this paper conforms with the NeurIPS Code of Ethics in every respect.

   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: There is no societal impact of the work 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: All the datasets used in this paper are publicly available and they contain no unsafe images.

    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: For the used datasets and pre-trained models, we have cited their corresponding works and licenses.

    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: [N/A]

    Justification: This paper does not release new 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: This 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.
    - 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: This 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: [N/A]

    Justification: The LLM is used only for writing, editing, or formatting purposes.

    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.