

# Latent Chain-of-Thought World Modeling for End-to-End Autonomous Driving

Shuhan Tan<sup>1,2\*</sup> Kashyap Chitta<sup>2</sup> Yuxiao Chen<sup>2</sup> Ran Tian<sup>2</sup> Yurong You<sup>2</sup> Yan Wang<sup>2</sup>  
Wenjie Luo<sup>2</sup> Yulong Cao<sup>2</sup> Philipp Krähenbühl<sup>1</sup> Marco Pavone<sup>2,3</sup> Boris Ivanovic<sup>2</sup>

<sup>1</sup>UT Austin    <sup>2</sup>NVIDIA    <sup>3</sup>Stanford University

## Abstract

Recent Vision-Language-Action (VLA) models for autonomous driving explore inference-time reasoning as a way to improve driving performance and safety in challenging scenarios. Most prior work uses natural language to express chain-of-thought (CoT) reasoning before producing driving actions. However, text may not be the most efficient representation for reasoning. In this work, we present **Latent-CoT-Drive (LCDrive)**: a model that expresses CoT in a latent language that captures possible outcomes of the driving actions being considered. Our approach unifies CoT reasoning and decision making by representing both in an action-aligned latent space. Instead of natural language, the model reasons by interleaving (1) action-proposal tokens, which use the same vocabulary as the model’s output actions; and (2) world model tokens, which are grounded in a learned latent world model and express future outcomes of these actions. We cold start latent CoT by supervising the model’s action proposals and world model tokens based on ground-truth future rollouts of the scene. We then post-train with closed-loop reinforcement learning to strengthen reasoning capabilities. On a large-scale end-to-end driving benchmark, LCDrive achieves faster inference, better trajectory quality, and larger improvements from interactive reinforcement learning compared to both non-reasoning and text-reasoning baselines.

## 1. Introduction

End-to-end (E2E) autonomous driving aims to map raw, multi-view camera streams together with ego state, history, and high-level navigation commands *directly* to future trajectories and low-level controls using a single policy [11, 37]. A growing trend is to instantiate this policy as a *Vision-Language–Action (VLA)* foundation model [17], pre-trained on large-scale vision-language data and fine-tuned on driving logs. Building on this trend, recent studies introduce *inference-time reasoning* by generating a text-

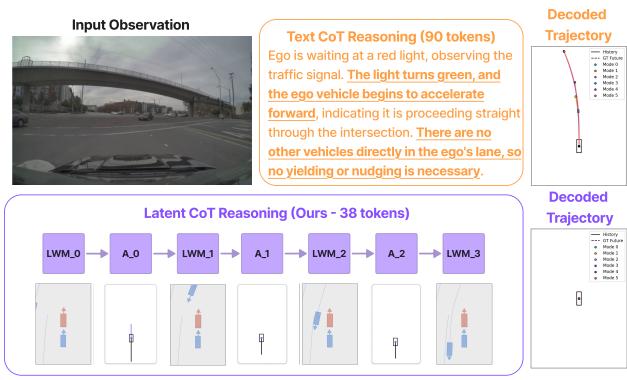


Figure 1. **Latent Chain-of-Thought Reasoning.** Compared to text-based CoT, our proposed Latent CoT provides more efficient and aligned reasoning traces for end-to-end driving VLA models.

based chain-of-thought (CoT) before committing to actions [14, 24, 33, 34, 41]. While this is a natural choice following recent works on reasoning LLMs [36], a textual CoT presents several limitations when applied to driving. First, natural language is ill-suited for representing spatiotemporal geometry and multi-agent interactions, which are central to driving decision-making. Second, autoregressively generating long chains of text introduces nontrivial latency, making real-time deployment challenging. Furthermore, the generated actions may significantly diverge from the preceding language rationales (e.g., the text states “go left” yet the action indicates a right turn) due to weak action–text alignment [24]. Accordingly, we argue that text is not the most suitable substrate in driving VLA models.

In this paper, we propose LCDrive, a Latent Chain-of-Thought framework for Driving VLA models. Instead of relying on textual CoT, LCDrive performs reasoning through vector-space supervised chain-of-thought tokens grounded in a learned *latent world model* (LWM), as shown in Fig. 1. The latent reasoning process alternates between action-proposal tokens and latent world model prediction tokens, thereby simulating counterfactual futures directly in latent space and using those futures to inform the choice of the next action. This interleaved latent CoT forms a structured

\*Work done during an internship at NVIDIA.

and compact reasoning trace grounded in the multi-agent interaction process, yielding both higher dynamical precision and significantly more efficient inference. We train LCDrive through a three-stage pipeline (Fig. 3). Starting from a pretrained non-reasoning VLA, we first cold-start with latent CoT by teacher-forcing the model with ground-truth (GT) world model states and reasoning actions proposed by the model itself. During this process, we simultaneously train a small LWM prediction head to predict LWM embeddings from proposed actions during inference. Next, we apply reinforcement learning (RL) [16] to refine this initial scaffold of latent reasoning and improve final action prediction using trajectory-level rewards.

We evaluate LCDrive on the large-scale PhysicalAI-AV dataset [23], consisting of 1727 hours of driving data across challenging urban scenarios with dense multi-agent interactions. In Tab. 1, we show that LCDrive improves trajectory fidelity and driving success compared to the baseline text-cot VLA models. Qualitative rollouts in Fig. 4 show how coherent latent-cot reasoning could benefit the driving performance over text-cot reasoning. We further include results across different scenario categories as well as extensive ablation experiments to show the superior performance of LCDrive.

**Contributions.** The main contributions of our work are:

- We rethink the representation of reasoning in VLA models for E2E driving with LCDrive, which conducts latent CoT with latent reasoning tokens strongly aligned with driving actions and a latent world model.
- We introduce a training framework combining latent CoT cold-start, world model training, and closed-loop RL, finding it especially effective for latent reasoning models.
- We demonstrate consistent empirical gains on a large, diverse E2E driving benchmark: LCDrive delivers faster inference, improved driving quality, and larger improvements under interactive RL compared to non-reasoning and text-reasoning baselines.

## 2. Related Work

**Driving VLA Models.** E2E driving systems learn a direct mapping from raw sensor inputs to trajectories or controls, aiming to reduce handcrafted components and human bias in the traditional perception–prediction–planning pipeline [11, 37]. Although this has shown effectiveness in common scenarios, classical E2E models struggle in long-tail driving scenarios due to limited world knowledge and weak reasoning structure. With the rise of foundation models, recent work has explored using pre-trained LLMs and multimodal LLMs as core building blocks for end-to-end driving policies. Early approaches incorporate these models primarily as backbones while still directly predicting actions from multimodal inputs [7, 15, 38, 40]. More recent methods introduce textual chain-of-thought [36] before ac-

tion prediction, leveraging the common-sense reasoning capabilities of LLM backbones to improve motion planning, particularly in rare or complex scenarios [14, 24, 33, 34, 41]. Different from previous works, our work departs from text-based CoT in driving VLAs and instead performs reasoning directly in a latent representation space.

**Latent World Models.** An alternative to model-free driving policy learning is to leverage latent world models (LWMs) [8, 30]. LWMs learn a generalized latent dynamics function that predicts the action-conditioned future evolution of the environment given current observations and planned actions. In autonomous driving, LWMs have recently emerged as flexible dynamic models that complement end-to-end policies. Some works jointly learn latent dynamics and the driving policy from expert demonstrations [10, 35], enabling the agent to model multi-agent interactions and future outcomes directly in latent space. Other efforts leverage trained latent world models to generate additional demonstrations for data augmentation [22, 27] or to serve as neural simulators for reinforcement learning–based policy training [13, 20]. These approaches highlight the promise of latent dynamics as a way to introduce structure and interaction-awareness into the learning process.

**Language-Free Paradigms for Reasoning.** While textual CoT has become a popular strategy for eliciting reasoning in multimodal models, it is not always an ideal medium for tasks that require geometry understanding and dynamics modeling. In addition, textual CoT often contains many non-essential tokens that do not contribute to the underlying reasoning process, inflating token usage and slowing inference without proportional improvements in decision quality [3, 6]. Recently, a line of work has begun to explore latent reasoning in LLMs, where intermediate computations are performed directly in latent space rather than in natural language. This paradigm enables more compact and informed reasoning [4, 5, 9], often with a more cost-effective inference budget. Building on these ideas, subsequent works extend latent reasoning to vision-language models, achieving latent spatial reasoning [19, 32]. In this work, we adopt this emerging paradigm within driving foundation models and perform reasoning entirely in latent space, demonstrating that latent reasoning is both more effective and substantially more efficient than textual reasoning for autonomous driving.

## 3. LCDrive: Driving with Latent CoT

### 3.1. Preliminaries

In this section, we formally define the task, followed by the concepts required to enable latent reasoning.

**Task.** We aim to design a policy that maps sensor streams and ego state inputs to future trajectories. Following previous works on reasoning VLA for driving [24], we regard

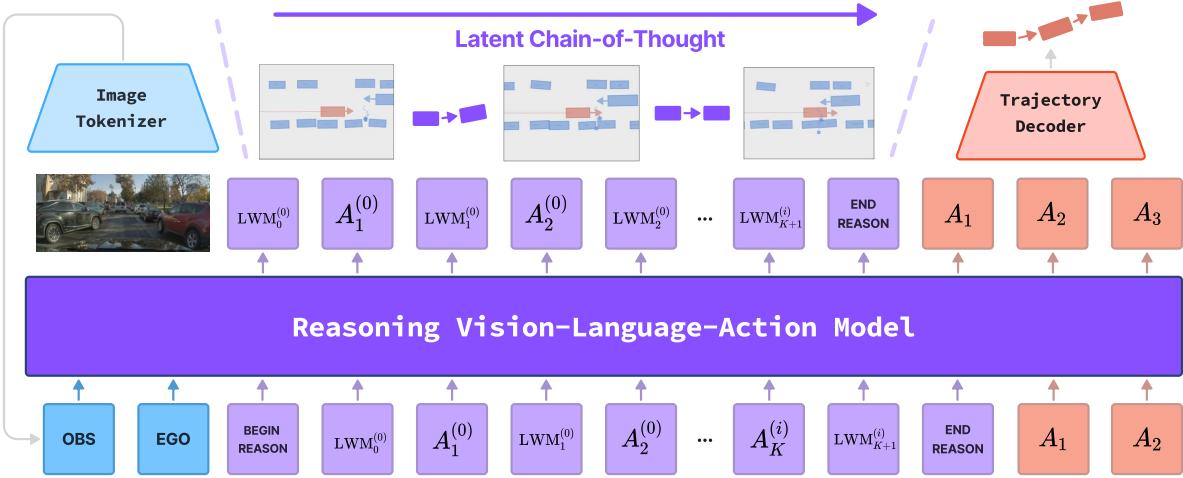


Figure 2. **Architecture.** Overview of our proposed latent reasoning framework.

E2E driving as modeling an autoregressive distribution over a token sequence that concatenates input information, (optional) reasoning trace, and the future trajectory of the ego vehicle  $\tau$ :

$$[o_{\text{image}}, o_{\text{ego}}, \text{REASON}, \tau], \quad (1)$$

where each component conditions on all previous ones. The inputs of the model include  $o_{\text{image}}$ ,  $M$  front-view (or multi-camera) frames over the last  $L$  steps; and  $o_{\text{ego}}$ , egomotion history. Given these inputs, the model produces (optional) REASON tokens followed by the future trajectory of the ego vehicle  $\tau$ . We parameterize  $\tau$  as the full 6.4 s future at 10 Hz, yielding a sequence of 64 future waypoints:

$$\tau = \{(x^i, y^i, \theta_{\text{yaw}}^i)\}_{i=1}^{64}. \quad (2)$$

**Input Tokenizers.** *Image tokenizer:* Following standard VLM practice, each frame in  $o_{\text{image}}$  is tokenized independently using a ViT-based encoder (e.g., [1, 28]), producing a sequence of visual tokens  $o_{\text{img}} = \text{Tok}_{\text{img}}(V_{t-L:t}^{1:M})$ . Tokens from different camera views and timestamps are concatenated to form the full visual token sequence. *Egomotion tokenizer:* The ego vehicle’s historical kinematics (speed, yaw rate, past  $k$  control actions) are embedded into a compact set of tokens  $o_{\text{ego}} = \text{Tok}_{\text{ego}}(e_t)$  with learned positional encoding.

**Trajectory Tokenizer.** The 6.4 s future trajectory at 10 Hz is represented using 64 discrete trajectory tokens  $\tau = a_{1:64}$ , one token per time step. Each  $a_i$  indexes a motion-primitive bin corresponding to the ego-frame  $\Delta$ -pose ( $\Delta x, \Delta y, \Delta \psi$ ). We build a 1024-code vocabulary via  $k$ -means on training

$\Delta$ -poses. We encode continuous trajectories by quantifying them to indices  $a_{1:64}$  with nearest-code assignment. We decode discrete indices back to  $\Delta$ -poses via codebook lookup and integrate them over time to recover continuous trajectories  $\hat{\tau}$ .

**Latent World Model (LWM).** We introduce an ego-centric latent world model state  $LWM_t$  that captures vectorized agent boxes and poses from online perception. Each  $LWM_t$  summarizes a fixed 1.0 s window at 10 Hz (10 frames) as a fixed-size set of vectorized representations (ego +  $K_{\text{agents}}$  nearest agents).  $LWM_0$  encodes the most recent history window up to the current time, which *starts* the reasoning process. It can be *given* from online perception (detection, tracking) or  by the VLA model itself.  $LWM_1, LWM_2, \dots$  represent future 1.0 s windows produced during latent reasoning, conditioned on proposal actions. We encode each LWM into a small set of latent worldmodel tokens  $LWM_0$  via a light Transformer module.

**Reasoning Tokens.** The presence of REASON is optional and used differently across different models. For the *non-reasoning* baseline model, we set REASON =  $\emptyset$ . For a fair comparison, the baseline may *optionally* condition on *only* REASON =  $[LWM_0]$  as context. For *text-based CoT* models (e.g., AR1 [24]), REASON consists of a sequence of natural-language tokens that verbally describe intermediate reasoning before action prediction. In this paper, we propose *latent CoT*, where REASON is instantiated as a short interleaved sequence of latent tokens composed of *action-proposal* tokens and counterfactual latent world-model tokens, initialized from the latent state  $LWM_0$ . By default,

$LWM_0$  is predicted by the VLA model itself given the sensor inputs as context. We detail the construction of latent REASON tokens in the following section.

### 3.2. Latent Chain-of-Thought Reasoning

We aim to design a compact, action-aligned reasoning process that performs latent counterfactual rollouts in the latent world model state, and keeps the CoT in the same vocabulary as the final trajectory output.

**Token Scheme.** We represent each reasoning branch as an interleaved action and latent world model trace  $R^{(i)}$ :

$$R^{(i)} = [A_0^{(i)}, LWM_1^{(i)}, A_1^{(i)}, LWM_2^{(i)}, \dots, A_{K-1}^{(i)}, LWM_K^{(i)}]. \quad (3)$$

Here  $A_t^{(i)}$  are *action-proposal* tokens drawn from the same action vocabulary as the final output, but grouped as a 1.0s block of 10 stepwise tokens:

$$A_t^{(i)} := (a_{10(t-1)+1}, \dots, a_{10t}),$$

which makes proposals easy to produce and interpret.  $LWM_{t+1}^{(i)}$  is the ego-centric latent world state summarizing the *same* 1.0s window at 10Hz. Reasoning is seeded by the history anchor  $LWM_0$ , after which we interleave  $(A_t^{(i)}, LWM_{t+1}^{(i)})$  for  $t = 1 \dots K$  to form  $R^{(i)}$ .

**Action Proposal.** At step  $t$ , the VLA proposes  $A_t^{(i)}$  conditioned on sensor tokens, the current world state, and the prior reasoning token sequence:

$$A_t^{(i)} \sim \pi_\theta(\cdot | o_{\text{image}}, o_{\text{ego}}, LWM_0, R_{<t}^{(i)}).$$

Note that  $A_t^{(i)}$  uses the same token vocabulary as the final trajectory prediction  $\tau$ . These proposals are only used as reasoning context and do *not* commit to a specific final plan.

**LWM Prediction.** Given the proposal as context, we predict the next latent world state:

$$LWM_{t+1}^{(i)} \sim q_\phi(\cdot | o_{\text{image}}, o_{\text{ego}}, LWM_0, R_{<t}^{(i)}, A_t^{(i)}).$$

In practice, we compute it with  $f_\phi(h_t^{\text{VLA}})$ , where  $h_t^{\text{VLA}}$  is the VLA hidden embedding after taking  $A_t^{(i)}$  as input and  $f_\phi$  is a lightweight MLP that outputs LWM tokens.

**Multi-Branch Reasoning.** To allow the model to spend more reasoning tokens on diverse strategies and paths, we enable autoregressive generation of a fixed number of branches  $B$  (default  $B = 2$ ). All branches share the history anchor  $LWM_0$  and are generated sequentially: for  $i = 1 \dots B$ , we produce  $R^{(i)}$  while conditioning on previously formed traces  $R^{(<i)}$ . This lets the model refer to prior latent reasoning when proposing the next branch, promoting diversity and yielding more plausible, complementary counterfactual futures under a bounded token budget.

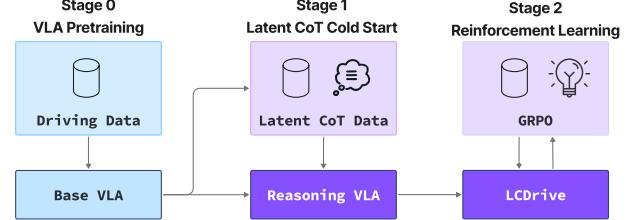


Figure 3. **Training strategy.** We first use a base non-reasoning VLA to create latent CoT data, and cold start LCDrive by supervised learning. Then, we conduct reinforcement learning to activate useful reasoning capacity of LCDrive.

In this paper, we fix both  $K$  and  $B$  at training and evaluation for simplicity.

**Action Prediction.** The complete reasoning context is

$$\text{REASON} = [LWM_0, R^{(1)}, \dots, R^{(B)}].$$

Conditioned on the sensor input and REASON in Eq. (1), the model predicts the 64 stepwise trajectory tokens  $a_{1:64}$  and decodes the final trajectory  $\hat{\tau}$ . The final actions attend to *all* proposals and their associated latent world model rollouts, forming rich counterfactual context that we will show yields higher-fidelity, safer, and more stable trajectories.

### 3.3. Training Strategy

We train LCDrive in three training stages (Fig. 3).

#### 3.3.1. Stage 0 - Non-reasoning Pretraining

We start from a *non-reasoning* VLA ( $\text{REASON} = \emptyset$ ) trained via supervised fine-tuning to predict trajectory tokens from driving data. We keep two copies of this model: (1) one serves as the initialization for LCDrive in the later fine-tuning stage; (2) the other is *frozen* and used solely to generate action-proposal tokens for latent reasoning.

#### 3.3.2. Stage 1 - CoT Cold Start

In this step, we aim to teach the VLA model the format and structure of latent CoT with teacher forcing. To this end, we construct supervision data for latent CoT REASON tokens through the following steps.

**Action Proposals.** Given sensor inputs, we use the *frozen* non-reasoning VLA  $\pi_0$  to sample  $B$  different trajectories  $\{\tilde{a}_{1:64}^{(i)}\}_{i=1}^B$  in random order. Each sample is sliced into  $K$  1.0s action blocks:  $\tilde{A}_t^{(i)} := (\tilde{a}_{10t+1}^{(i)}, \dots, \tilde{a}_{10t+10}^{(i)})$ .

**Action-conditioned LWM targets.** For each block  $\tilde{A}_t^{(i)}$ , we integrate its ego-frame  $\Delta$ -poses to obtain the ego pose for that 1.0s window, re-center the *GT* future tracked agent bounding boxes into this ego frame, and encode them to produce a target latent world state:  $\tilde{LWM}_{t+1}^{(i)}$ . This yields branch-specific world tokens  $\{\tilde{LWM}_{t+1}^{(i)}\}$  that reflect the *consequences* of each proposal window.

**Supervision sequence.** Action proposals and targets are interleaved to form  $B$  reasoning traces  $R^{(i)}$  (Eq. (3)). The full training sequence in Eq. (1) thus becomes

$$[o_{\text{image}}, o_{\text{ego}}, \underbrace{\text{LWM}_0, R^{(1)}, \dots, R^{(B)}}_{\text{REASON}}, a_{1:64}].$$

We input this full sequence to LCDrive during training.

**Objective.** We train LCDrive to minimize a standard cross-entropy loss over proposals and the final action plan:

$$\mathcal{L}_{\text{token}} = \sum_{i=1}^B \sum_{t=0}^{K-1} \text{CE}(A_t^{(i)}, \tilde{A}_t^{(i)}) + \text{CE}(a_{1:64}, a_{1:64}^*).$$

Additionally, we train the LWM prediction module to predict the corresponding ground-truth LWM embedding during reasoning as well as the initial  $\text{LWM}_0$ :

$$\mathcal{L}_{\text{lwm}} = \|\text{LWM}_0 - \tilde{\text{LWM}}_0\|_2^2 + \sum_{i,t} \|\text{LWM}_{t+1}^{(i)} - \tilde{\text{LWM}}_{t+1}^{(i)}\|_2^2.$$

The overall objective of LCDrive in Stage 1 is:

$$\mathcal{L}_{\text{stage-1}} = \mathcal{L}_{\text{token}} + \lambda \mathcal{L}_{\text{lwm}}. \quad (4)$$

### 3.3.3. Stage 2 - Reinforcement Learning

The second stage post-trains LCDrive to actively produce useful latent reasoning and output better actions. By directly encourage the model to improve the feasibility of the final action *conditioned* the latent reasoning process, the model learns how to produce reasoning tokens beyond imitating the frozen model in Stage 1.

**Rollout.** For each training input, we keep the fixed reasoning budget ( $K, B$ ) and generate a group of  $G$  stochastic *completions*: the policy autoregressively generates *action-proposal* blocks interleaved with latent world states to form branch traces  $R^{(i)}$ , and concatenates them into  $\text{REASON} = [\text{LWM}_0, R^{(1)}, \dots, R^{(B)}]$ . Conditioned on  $\text{REASON}$  and the sensor tokens, the policy then produces the 64 trajectory tokens  $a_{1:64}$  and decodes  $\hat{\tau}$ .

**Reward.** We use a single trajectory-accuracy signal: *Average Displacement Error (ADE)* in meters between the predicted and expert trajectories over the 6.4 s horizon:

$$\text{ADE}(\hat{\tau}, \tau^*) = \frac{1}{64} \sum_{i=1}^{64} \|\hat{\mathbf{p}}_i - \mathbf{p}_i^*\|_2,$$

where  $\mathbf{p}_i$  is the  $i$ -th 2D ego location along the trajectory. The reward for completion  $j$  is  $R^{(j)} = -\text{ADE}(\hat{\tau}^{(j)}, \tau^*)$ .

**Learning Algorithm.** We use Group Relative Policy Optimization (GRPO) [31] for RL training. Specifically, for each training example, we sample a group of  $G$  completions  $\{\hat{\tau}^{(j)}\}_{j=1}^G$ , compute a trajectory-centric reward  $R^{(j)}$ , and construct centered advantages for each completion:  $A^{(j)} =$

$R^{(j)} - \frac{1}{G} \sum_k R^{(k)}$ . We then maximize the advantage-weighted log-probability of the *generated* tokens, including both proposal and final action tokens:

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{G} \sum_{j=1}^G A^{(j)} \sum_t \log \pi_\theta(x_t^{(j)} \mid \text{context}_t^{(j)}). \quad (5)$$

Empirically, we found that GRPO performs best without KL regularization, so we omit the KL term in the final objective. Note that Stage 2 can also be applied to a non-reasoning baseline with  $\text{REASON} = \emptyset$ . We will show in Sec. 4.2 that RL yields substantially larger gains for LCDrive than the baseline.

## 4. Experiments

### 4.1. Setup

**Dataset.** We conduct our experiments on the recently released PhysicalAI-AV dataset [23]. It provides large-scale (1700+ hours) real-world multi-camera driving logs with precise ego trajectories and dense multi-agent annotations, enabling realistic end-to-end driving evaluation. In coordination with the dataset authors, we obtained a *scenario-balanced* subset that maintains consistency with the official public splits of the full dataset: 39,072 training clips (87 hours) and 23,758 validation clips (53 hours). For each clip, we consider 1.6 s of history and 6.4 s of future ego and surrounding-agent trajectories at 10 Hz.

As summarized in Tab. 2, the subset is constructed to balance nominal and eventful scenes: 30% of clips are *General Driving* and the remaining 70% are evenly distributed across 14 specific scenarios (e.g., lane keeping, intersection navigation, merges, cut-ins), with 5% of the data per category. In addition to its significantly larger scale compared to prior E2E driving validation benchmarks (e.g. nuScenes [2] with only 150 validation clips, less than 1 hour), this split provides a near-uniform scenario distribution. It avoids dominance by easy cases (e.g., 73.9% straight driving in nuScenes [21]) and enables a fair, per-scenario evaluation of driving models.

**Metrics.** For each input clip, we randomly sample 6 trajectories from the evaluated model. Metrics are then computed for each sample, and the average over all samples is taken to be the overall score of the clip.

To measure the similarity of the model output with the expert driving behaviors, we report ADE (meters) as the mean  $\ell_2$  error between the predicted ego positions and expert positions at 10 Hz over the  $T = 64$  steps. We also measure the safety of the model driving behavior: OffRoad<sub>2.5</sub> and OffRoad<sub>5.0</sub> (%) are the fraction of clips for which *any* point in the predicted ego footprint leaves the drivable area within the first  $T \in \{2.5, 5.0\}$  seconds. Coll@2.5 and Coll@5.0 (%) are the fraction of clips that experience *any*

intersection between the ego polygon and any other agent polygon within the same  $S \in \{2.5, 5.0\}$  s window. Corner Dist (m) measures the mean Euclidean distance between corresponding corners of the predicted and expert ego boxes (with fixed vehicle dimensions) over the 64 steps at 10 Hz, capturing both translation and heading errors. More detailed metrics can be found in the supplementary material.

**Baselines.** All variants share the same non-reasoning backbone, trajectory tokenizer, and decoder. Unless noted, training uses the PhysicalAI split mentioned above. All models receive identical inputs and differ only in the format of the REASON tokens. We compare 1) **No CoT** ( $\emptyset$ ): VLA without any reasoning tokens; 2) **LWM<sub>0</sub>-only**: the model conditions on the history latent world model state LWM<sub>0</sub> but performs no interleaved rollout; 3) **Latent CoT**: our interleaved action-proposal and latent world-model tokens, initialized from LWM<sub>0</sub>; 4) **Text CoT**: a language-reasoning baseline that uses English text for reasoning. We mainly compare methods that *predict* all the LWM tokens needed in the reasoning stage. To show performance upper-bounds, we also compare with methods that take *GT* LWM tokens within the reasoning space, marked with \*. Our model, LCDrive, is **Latent CoT with Predicted LWM**; we also report performance with and without the RL training stage.

**Text CoT baseline.** Since obtaining Text-CoT labels for the PhysicalAI-AV dataset [23] is non-trivial, we use model weights provided by the AR1 team [24]. The model shares the same AR1 architecture, and is pretrained on a large proprietary dataset of driving logs that is an over  $100\times$  larger superset of our training set, followed by finetuning on a smaller set of Text-CoT-paired data (though still  $\sim 10\times$  larger than our training set). Given its substantially larger training corpus and direct supervision on carefully-curated text CoT dataset, this baseline is expected to perform better than models trained only on PhysicalAI-AV.

**Implementation.** We adopt a Qwen3-0.5B [39] LLM as the language-action module and a DINOv2 [25] ViT as the image encoder, following the AR1 architecture design [24]. Each input clip uses two front-view cameras (wide 120° and telephoto 30°) with 320×512 resolution visual inputs. The encoded image tokens are concatenated with ego tokens and REASON tokens before being fed into the decoder.

*Stage-0 non-reasoning pretrain:* We first train a non-reasoning model for 100k steps on the PhysicalAI-AV training split using batch size 128, learning rate 4e-5, and cosine annealing. *Stage-1 CoT cold start:* We then enable latent reasoning and train for 10k steps with the same optimizer settings. Action proposals are generated from the frozen non-reasoning model using temperature 0.6 and top-p = 0.98. The loss in Eq. (4) is weighted by  $\lambda = 0.1$ . *Stage-2 GRPO:* We finally apply RL post-training with GRPO for 3k steps using group size 8, effective batch size 32 sampled completions per update, and a learning rate of 1e-6. We

set the reasoning depth  $K = 5$  and branch factor  $B = 2$  through our experiments unless otherwise specified.

For all approaches, we use temperature 0.6 and top-p = 0.98 during sampling of the 6 trajectories per input.

## 4.2. Main Results

**PhysicalAI-AV evaluation.** We show the main result in Table 1. We first compare the oracle models that use the LWM states (GT LWM). When provided with ground-truth LWM, Latent CoT\* substantially outperforms simply conditioning on the history state (LWM<sub>0</sub>-only\*): ADE improves from 1.393 to 1.268, and RL further reduces it to 1.197 while also improving safety (e.g., reducing Coll<sub>5.0</sub> from 0.905 to 0.867). These results indicate that counterfactual reasoning with LWM tokens is an effective substrate for planning with an accurate world state.

Note that RL is beneficial *only* when the model conducts reasoning. The first two rows show that adding RL to LWM<sub>0</sub>-only\* yields no gain in ADE and worsens OffRoad<sub>5</sub>, whereas RL on Latent CoT\* consistently improves both accuracy and safety. This suggests that RL *activates* a useful latent CoT process and enables closed-loop interactive policy optimization with internal latent rollouts.

In the practical (non-oracle) setting, our model LCDrive remains strong. LCDrive outperforms the non-reasoning baseline by a clear margin (ADE 1.626 vs. 1.762; OffRoad<sub>2.5</sub> 1.219 vs. 1.753; Coll<sub>5</sub> 0.836 vs. 2.207), indicating that learned LWM tokens are highly informative at inference time. Notably, the latent CoT process is *robust* to noise in the predicted LWM. Despite errors during model prediction, the interleaved Latent CoT yields consistent gains over the non-reasoning policy. Moreover, adding RL on top of predicted LWM further improves accuracy and safety, delivering a clear additional gain. This demonstrates that RL remains beneficial even when the world model is learned, and that it helps the policy exploit the latent CoT interface more effectively.

Compared with the Text CoT baseline, LCDrive is comparable without RL and clearly better with RL. Before RL, LCDrive (ADE 1.668) is on par with Text CoT (1.650). After RL, LCDrive achieves 1.626 ADE and lower risk (OffRoad<sub>2.5</sub> 1.219 vs. 1.391; Coll<sub>5</sub> 0.836 vs. 0.905), despite Text CoT being trained on a *much* larger, CoT-annotated dataset.

Overall, we conclude that (1) LWM tokens provide a more effective reasoning medium than text; (2) RL is especially impactful when paired with latent CoT, reliably translating internal rollouts into better final actions, and (3) introducing latent CoT consistently improves driving quality over its non-reasoning counterpart for driving VLAs.

**Scenario breakdown.** We further evaluate LCDrive across diverse driving scenarios. As shown in Tab. 2, LCDrive achieves consistent improvements over both non-reasoning

REASON	GT LWM	RL	ADE $\downarrow$	OffRoad <sub>2.5</sub> $\downarrow$	OffRoad <sub>5.0</sub> $\downarrow$	Coll <sub>2.5</sub> $\downarrow$	Coll <sub>5.0</sub> $\downarrow$	Corner Dist. $\downarrow$
LWM <sub>0</sub> -only*	✓		1.393	<b>1.250</b>	<b>3.104</b>	<b>0.259</b>	1.198	0.835
	✓	✓	1.397	1.430	4.218	0.326	1.706	0.948
Latent CoT*	✓		1.268	1.309	3.408	0.327	0.905	<b>0.691</b>
	✓	✓	<b>1.197</b>	1.303	3.443	0.318	<b>0.867</b>	0.739
∅ (None)			1.762	1.753	5.279	0.348	2.207	0.986
Text CoT			1.650	1.391	<b>3.005</b>	<b>0.276</b>	0.905	<b>0.642</b>
Latent CoT			1.668	1.268	3.536	0.322	1.591	0.904
Latent CoT (LCDdrive)	✓		<b>1.626</b>	<b>1.219</b>	3.292	0.289	<b>0.836</b>	0.880

Table 1. **Main evaluation results** on the PhysicalAI-AV dataset [23]. Lower is better for all metrics, bold is best.

Scenario Category	ADE @ 6.4 s (in meters, lower is better)					
	LWM <sub>0</sub> -only*	Latent CoT*	Latent CoT + RL*	No CoT	Text CoT	LCDdrive
General Driving	1.015	0.899	0.838	1.268	1.434	<b>1.166</b>
Stop for Vehicle	0.760	0.542	0.514	0.995	<b>0.919</b>	0.942
Speed Control	1.109	1.004	1.675	1.573	2.037	<b>1.376</b>
Nudge Static Obstacle Maneuver	1.226	1.085	1.518	1.575	1.855	<b>1.387</b>
Traffic Control Compliance	1.248	1.087	0.870	1.627	<b>1.312</b>	1.431
Vulnerable Road Users (VRU)	1.369	1.215	1.246	1.850	<b>1.655</b>	1.707
Lead Vehicle Following	1.421	1.305	1.112	1.861	<b>1.455</b>	1.708
Intersection Navigation	1.456	1.300	1.277	1.887	<b>1.725</b>	1.730
Lane Keeping	1.535	1.484	1.420	2.021	<b>1.783</b>	1.828
Nudge Maneuver	1.541	1.453	1.554	1.966	1.909	<b>1.824</b>
Lane Keeping Curve	1.675	1.553	1.857	2.162	2.002	<b>1.986</b>
Merging	1.716	1.571	1.375	2.215	2.169	<b>2.089</b>
Turning Maneuver	1.839	1.652	2.041	2.347	<b>1.990</b>	2.085
Cut-In	2.076	1.913	1.220	2.583	<b>1.884</b>	2.385
Lane Change	2.053	1.922	1.897	2.579	<b>2.167</b>	2.423
<b>Overall</b>	1.397	1.268	1.197	1.762	1.650	<b>1.626</b>

Table 2. **ADE split by scenario.** Columns are ordered with methods using GT LWM (marked with \*) shown first. Bold is best.

and text-reasoning baselines in nearly all categories. Compared with the non-reasoning model, LCDdrive reduces ADE by 7–15% on most complex maneuvers such as *Intersection Navigation*, *Turning Maneuver*, and *Merging*, which require anticipating multi-agent interactions. The largest relative gains appear in *Traffic Control Compliance*, *Speed Control*, and *Nudge Static Obstacle Maneuver*, demonstrating the effectiveness of reasoning with LWM which predicts other agents states into the future.

Compared with the Text CoT model, LCDdrive achieves lower ADE in every scenario, despite Text CoT being trained on a much larger CoT-annotated corpus. Notably, the gaps are largest in interaction-heavy settings such as *Lead Vehicle Following* (1.708 vs. 1.455) and *Stop for Vehicle* (0.942 vs. 0.919) indicating that latent reasoning grounded in the LWM space generalizes better to diverse multi-agent behaviors.

The oracle results (Latent CoT\*) further illustrate the potential of latent reasoning. When supplied with perfect

LWM, latent CoT reduces the ADE by large margins across nearly all categories (e.g., 1.300 in *Intersection Navigation* and 0.542 in *Stop for Vehicle*). Adding RL on top of oracle LWM yields even stronger results in difficult scenarios such as *Cut-In* (1.220) and *Lane Change* (1.897), demonstrating that latent reasoning becomes especially powerful when accurate multi-agent futures are available.

Overall, the per-scenario analysis shows that latent CoT provides broad, uniform improvements across the full spectrum of driving tasks. Reasoning in the latent world-model space leads to better anticipation, more stable long-horizon predictions, and improved performance on categories that require understanding interactions, maneuvers, and compliance with traffic rules. These results highlight that latent chain-of-thought is an effective and generalizable reasoning mechanism for VLA-based driving models.

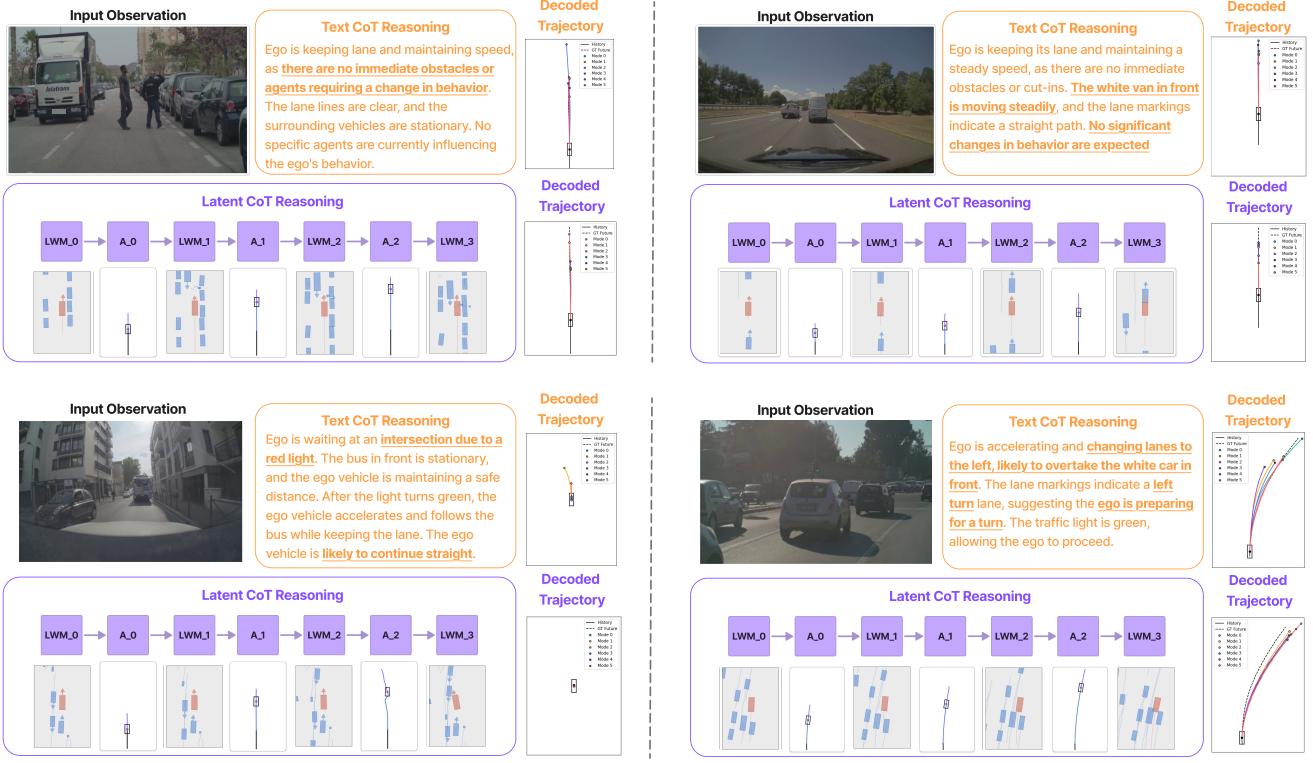


Figure 4. **Qualitative Results.** Qualitative comparison of textual and latent reasoning in driving VLA models. Latent CoT captures fine-grained spatial relationships and multi-agent interactions while using a smaller inference budget, leading to more stable and accurate trajectory predictions. In each case, we highlight the main misalignment of the Text CoT reasoning with the final trajectory.

### 4.3. Qualitative Results

In Fig. 4, we analyze several textual and latent reasoning traces output by the Text CoT baseline and LCDdrive respectively. In each example, textual CoT provides a high-level narrative of the environment, but the descriptions remain generic and fail to capture the fine-grained spatial relationships and multi-agent interactions needed for precise driving decisions. Moreover, these textual rationales often contain numerous non-essential tokens (e.g., stylistic or filler words), which increase inference latency without improving the underlying reasoning. In contrast, LCDdrive produces a compact sequence of interleaved action-proposal tokens and latent world-model predictions that encode informative scene dynamics allowing the model to perform multi-step reasoning using only a few compact vector tokens. Across all examples, LCDdrive produces motion plan predictions that align closer with the ground truth demonstration while using a significantly lower inference budget.

For each scene, we show one latent world model reasoning trace, selecting the one with the most similar action tokens to the final decoded trajectory. While LCDdrive is capable of predicting LWM tokens, it does not require a decoder that reconstructs these tokens back into a human-

interpretable visualization. Therefore, for this comparison, we use the Latent CoT\* model from Tab. 1 that accesses GT LWM tokens, which we visualize with the corresponding action tokens interleaved.

### 5. Conclusion

In conclusion, we present LCDdrive: a model that replaces natural language CoT reasoning with a compact, action-aligned latent reasoning space for autonomous driving. By interleaving action-proposal tokens and world-model tokens, our approach unifies inference-time reasoning and decision making within a single latent world modeling process. This design enables LCDdrive to reason about the effects of candidate actions via their predicted future outcomes, while avoiding the inefficiencies and potential misalignment of text-based explanations. Experiments on large-scale real-world driving data demonstrates that latent CoT not only accelerates inference, but also leads to higher-quality trajectories and enables further improvements from closed-loop RL compared to both non-reasoning and text-reasoning baselines.

While these results are encouraging, there are a few limitations that motivate future work: First, training latent CoT

currently requires a source of supervision (e.g., GT agent bounding boxes) to ground the representation, which may be difficult to obtain at scale (though recent efforts in auto-labeling are addressing this [12, 18, 26, 29]). Second, our current model does not support easy recovery of a human-interpretable representation from a latent CoT token (e.g., for in-car visualization). Accordingly, building a deeper understanding of the efficiency-interpretability spectrum is an exciting area of future work. Finally, our model does not yet support flexible reasoning lengths adjusting to different task difficulties, which would make it even more efficient.

## References

- [1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. 3
- [2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liang, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multi-modal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 5
- [3] Xinghao Chen, Zhijing Sun, Guo Wenjin, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, et al. Unveiling the key factors for distilling chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15094–15119, 2025. 2
- [4] Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*, 2025. 2
- [5] 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. 2
- [6] Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*, 2025. 2
- [7] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. *arXiv preprint arXiv:2503.19755*, 2025. 2
- [8] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018. 2
- [9] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhitong Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024. 2
- [10] Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zachary Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. *Advances in Neural Information Processing Systems*, 35:20703–20716, 2022. 2
- [11] Hanxue Hu, Ye Yuan, Hongyang Xu, Zhaoyang Chen, Ming Liang, Zhiding Li, Yuexin Ma, Xiaodong Shen, Yuning Chai, Xiaoqing Tan, et al. UniAD: Unified perception and prediction for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2
- [12] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. ViPE: Video pose engine for 3D geometric perception. In *NVIDIA Research Whitepapers arXiv:2508.10934*, 2025. 9
- [13] Weidong Huang, Jiaming Ji, Chunhe Xia, Borong Zhang, and Yaodong Yang. Safedreamer: Safe reinforcement learning with world models. *arXiv preprint arXiv:2307.07176*, 2023. 2
- [14] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. EMMA: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024. 1, 2
- [15] Anqing Jiang, Yu Gao, Yiru Wang, Zhigang Sun, Shuo Wang, Yuwen Heng, Hao Sun, Shichen Tang, Lijuan Zhu, Jinzhao Chai, et al. Irl-vla: Training an vision-language-action policy via reward world model. *arXiv preprint arXiv:2508.06571*, 2025. 2
- [16] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996. 2
- [17] Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 13:162467–162504, 2025. 1
- [18] In-Jae Lee, Mungyeom Kim, Kwonyoung Ryu, Pierre Musacchio, and Jaesik Park. OpenBox: Annotate any bounding boxes in 3d. In *Proceedings of the Int. Conf. on Neural Information Processing Systems (NeurIPS)*, 2025. 9
- [19] Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muham Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025. 2
- [20] Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in carla-v2). In *European Conference on Computer Vision*, pages 142–158. Springer, 2024. 2
- [21] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahua Li, Jan Kautz, Tong Lu, and Jose M. Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 5
- [22] Jiageng Mao, Boyi Li, Boris Ivanovic, Yuxiao Chen, Yan Wang, Yurong You, Chaowei Xiao, Danfei Xu, Marco Pavone, and Yue Wang. Dreamdrive: Generative 4d scene modeling from street view images. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 367–374. IEEE, 2025. 2

- [23] NVIDIA. Physical AI autonomous vehicles dataset. <https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles>, 2025. 2, 5, 6, 7
- [24] NVIDIA, 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. 1, 2, 3, 6
- [25] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023. 6
- [26] Aljosa Osep, Tim Meinhardt, Francesco Ferroni, Neehar Peri, Deva Ramanan, and Laura Leal-Taixé. Better Call SAL: Towards learning to segment anything in lidar. In *European Conference on Computer Vision (ECCV)*, 2024. 9
- [27] Alexander Popov, Alperen Degirmenci, David Wehr, Shashank Hegde, Ryan Oldja, Alexey Kamenev, Bertrand Douillard, David Nistér, Urs Muller, Ruchi Bhargava, et al. Mitigating covariate shift in imitation learning for autonomous vehicles using latent space generative world models. *arXiv preprint arXiv:2409.16663*, 2024. 2
- [28] Qwen Team. Qwen3-VL: Sharper vision, deeper thought, broader action. <https://qwen.ai/blog?id=99f0335c4ad9ff6153e517418d48535ab6d8afe&from=research.latest-advancements-list>, 2025. 3
- [29] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädl, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 9
- [30] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, et al. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019. 2
- [31] Zihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 5
- [32] Guohao Sun, Hang Hua, Jian Wang, Jiebo Luo, Sohail Di-anat, Majid Rabbani, Raghuveer Rao, and Zhiqiang Tao. Latent chain-of-thought for visual reasoning. *arXiv preprint arXiv:2510.23925*, 2025. 2
- [33] Ran Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone.
- Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. In *Conference on Robot Learning*, 2024. 1, 2
- [34] Tianqi Wang, Enze Xie, Ruihang Chu, Zhenguo Li, and Ping Luo. DriveCoT: Integrating chain-of-thought reasoning with end-to-end driving. *arXiv preprint arXiv:2403.16996*, 2024. 1, 2
- [35] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pages 55–72. Springer, 2024. 2
- [36] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. 1, 2
- [37] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. PARA-Drive: Parallelized Architecture for Real-time Autonomous Driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024. 1, 2
- [38] Yichen Xie, Runsheng Xu, Tong He, Jyh-Jing Hwang, Katie Luo, Jingwei Ji, Hubert Lin, Letian Chen, Yiren Lu, Zhaoqi Leng, et al. S4-driver: Scalable self-supervised driving multimodal large language model with spatio-temporal visual representation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1622–1632, 2025. 2
- [39] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 6
- [40] Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C Knoll. OpenDriveVLA: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*, 2025. 2
- [41] Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. AutoVLA: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*, 2025. 1, 2

## A. Additional Implementation Details

### A.1. Latent World Model Encoder

Our latent world model (LWM) encodes the surrounding agents around the ego vehicle (*excluding* the ego vehicle) into a compact set of tokens for latent chain-of-thought reasoning. Concretely, each LWM state summarizes a fixed 1.0 s window at 10 Hz in an ego-centric frame.

**Per-agent temporal encoder.** For each clip, we select the  $N$  nearest agents (based on distance in the current frame). The raw per-timestep state of each agent includes position, heading, dimensions, velocity, and other kinematic attributes. We stack these over a 1.0 s window (10 frames) to obtain

$$\text{agent\_state} \in \mathbb{R}^{B \times N \times T \times F},$$

where  $B$  is the batch size,  $N$  the number of agents,  $T=10$  the number of timesteps, and  $F$  the number of input features. We first augment the state with 4 oriented corner points of the 3D bounding box (projected to BEV), resulting in 8 additional normalized features per timestep. A linear layer projects the concatenated features from dimension  $(F+8)$  to a latent dimension  $d_{\text{lwm}}$ , after which we apply: 1) a learned timestep embedding added along the temporal axis; 2) an agent-type embedding (shared over timesteps) added per agent; 3) a stack of MLP residual blocks along the feature dimension. This produces a sequence of per-agent, per-timestep features of shape  $\mathbb{R}^{B \times N \times T \times d_{\text{lwm}}}$ .

**Temporal pooling per agent.** To summarize the  $T=10$  timesteps into a single feature per agent, we use a learnable query vector and a cross-attention layer along the time axis. The query attends to the  $T$  timestep features with an attention mask that ignores invalid timesteps, yielding one vector per agent:

$$\text{LWM\_agent} \in \mathbb{R}^{B \times N \times d_{\text{lwm}}}.$$

**Two-token LWM summarization.** The latent world model state  $\text{LWM}_t$  used in LCDrive is a compact summary of all agents in the 1.0 s window. We train an additional attention layer with  $M << N$  learnable query tokens, each of dimension  $d_{\text{lwm}}$ , to attend over the  $N$  agent features:

$$\text{LWM}_t = \text{Attn}(Q_M, \text{LWM\_agent}) \in \mathbb{R}^{B \times M \times d_{\text{lwm}}}.$$

These  $M$  tokens keep the LWM interface extremely compact for latent reasoning. In this paper, we use  $N = 64$  and  $M = 2$ , maintaining a compact representation of LWM while capturing rich agent state information.

### A.2. Stage 1: CoT Cold Start

In Stage 1, we teach the model the structure of latent chain-of-thought (CoT) by *teacher forcing* both the action-proposal tokens and the corresponding latent world model

(LWM) tokens. Here we focus on how we construct the supervised reasoning sequence.

**Action proposals from a frozen GT-LWM model.** We start from the  $\text{LWM}_0$ -only model with ground-truth LWM inputs (Row 1 of Tab. 1 in the main paper). This model is trained without latent reasoning and serves as a strong teacher that produces full 6.4 s trajectories. Given sensor inputs  $(o_{\text{image}}, o_{\text{ego}})$  and the history latent state  $\text{LWM}_0$ , the frozen teacher  $\pi_0$  autoregressively samples discrete trajectory tokens

$$a_{1:64} \sim \pi_0(\cdot | o_{\text{image}}, o_{\text{ego}}, \text{LWM}_0).$$

For each training clip, we draw  $B$  such trajectories  $\{a_{1:64}^{(i)}\}_{i=1}^B$  using top- $p$  sampling (temperature 0.6,  $p = 0.98$ ). Each sampled trajectory is then sliced into  $K$  non-overlapping 1.0 s action blocks of length 10:

$$A_t^{(i)} := (a_{10t+1}^{(i)}, \dots, a_{10(t+1)}^{(i)}), \quad t = 0, \dots, K-1.$$

These blocks define the *target* action-proposal tokens that our latent CoT policy imitates during cold start.

**Action-conditioned LWM supervision.** For each branch  $i$  and block index  $t$ , we construct an LWM supervision token  $\text{LWM}_{t+1}^{(i)}$  that encodes the *future world state conditioned on the proposal  $A_t^{(i)}$* .

Starting from the ground-truth ego pose at the beginning of the window, we integrate the sequence of 10 motion-primitive codes in  $A_t^{(i)}$  to obtain the ego pose trajectory over that 1.0 s interval. At each timestep, we 1) take the ground-truth bounding boxes of all tracked agents from the PhysicalAI-AV dataset; 2) transform these boxes into the ego-centric frame defined by the integrated ego pose (translation and rotation); 3) feed the resulting agent states into the LWM encoder described in the last subsection. The encoder yields a compact latent world-model summary for that 1.0 s window, which we store as the target token  $\text{LWM}_{t+1}^{(i)}$ . Repeating this for all blocks  $t = 0, \dots, K-1$  produces an interleaved supervision trace

$$R^{(i)} = [A_0^{(i)}, \text{LWM}_1^{(i)}, \dots, A_{K-1}^{(i)}, \text{LWM}_K^{(i)}].$$

### A.3. Stage 2: Reinforcement Learning

For the reinforcement learning stage of LCDrive, we adopt the `cosmos-rl` framework<sup>1</sup> as our RL backbone. All RL experiments are conducted on a single 8-GPU node. We allocate 6 GPUs as rollout actors, each running an independent sampler replica of LCDrive in inference mode; 2 GPUs as learners, jointly performing GRPO optimization and broadcasting updated parameters to all actors. This partitioning enables high-throughput rollout while keeping optimization stable and fully GPU-resident.

<sup>1</sup><https://github.com/nvidia-cosmos/cosmos-rl>

Metric	No RL	With RL
Reasoning Diversity	0.412	0.353
Reasoning–Action Alignment	0.614	0.581
Reasoning Quality	0.976	0.961
Final-Action Quality	0.784	0.749

Table 3. Reasoning action analysis of LCDrive with/without RL training, using GT LWM. All values are ADE (m).

The learning objective is the GRPO loss described in the main paper, but applied to *all* latent CoT tokens. This allows RL to restructure and refine the latent reasoning process itself, beyond imitation from Stage 1. Empirically, we observe that latent reasoning benefits significantly more from RL than non-reasoning baselines, highlighting the importance of closed-loop optimization through the latent world-model interface.

## B. Reasoning Action Analysis

To better understand the behavior of latent chain-of-thought reasoning before/after reinforcement learning, we analyze the relationship between the *proposal actions* generated during the reasoning stage and the *final action* output by the policy. For each validation clip, LCDrive generates  $B=2$  reasoning branches, each producing a 50-step rollout trajectory, decoded from the action proposal tokens  $A_t^{(i)}$ . The final decoded trajectory has 64 steps; we truncate it to the first 50 steps for consistent comparison.

Let  $\hat{\tau}_0$  and  $\hat{\tau}_1$  denote the two proposal rollouts,  $\hat{\tau}_{\text{final}}$  the final action trajectory (trimmed to 50 steps), and  $\tau^*$  the ground-truth future trajectory. We define four metrics as below. All metrics are reported as Average Displacement Error (ADE) in meters.

### 1. Reasoning Diversity:

$$\text{Diversity} = \text{ADE}(\hat{\tau}_0, \hat{\tau}_1),$$

measures how different the two proposal branches are.

### 2. Reasoning–Action Alignment:

$$\text{Alignment} = \min_{k \in \{0,1\}} \text{ADE}(\hat{\tau}_{\text{final}}, \hat{\tau}_k),$$

measures how closely the final action aligns with at least one proposal.

### 3. Reasoning Quality:

$$\text{Quality} = \frac{1}{2} \sum_{k \in \{0,1\}} \text{ADE}(\hat{\tau}_k, \tau^*),$$

measures how good the proposals are with respect to the ground-truth trajectory.

### 4. Final-Action Quality:

$$\text{Final-Action} = \text{ADE}(\hat{\tau}_{\text{final}}, \tau^*),$$

the standard ADE of the final action relative to ground truth.

We evaluate LCDrive using GT LWM and compare the result with and without RL, and show the result in Tab. 3. We summarize two key aspects of the reasoning behavior: (i) how latent reasoning behaves in general, and (ii) how reinforcement learning further improves it. Together, these results reveal the functional role of latent chain-of-thought reasoning in LCDrive. We have the following observations:

**1) Final actions improve upon the reasoning proposals.** In both settings, we observe that Final-Action Quality < Reasoning Quality. This means that even though the reasoning branches provide two candidate future plans, the decoder does not simply copy a branch. Instead, it selects the more promising proposal and further *refines* it to produce a more accurate final trajectory. This refinement effect becomes even stronger after RL.

**2) Strong alignment between reasoning proposals and the final action.** Across both models, the Reasoning–Action Alignment score remains small, indicating that the final trajectory lies close to at least one of the proposal branches. This shows that the proposal actions are actively used. After RL, the alignment improves ( $0.614 \rightarrow 0.581$ ), indicating that RL strengthens the integration between proposals and the final action. Note that the Reasoning–Action Alignment score is consistently lower than the Reasoning Quality score. This means that the final action lies *closer to one of the reasoning proposals* than either proposal lies to the ground truth. Thus, the final plan is strongly aligned with the latent reasoning process, showing that LCDrive relies on and refines the reasoning rollouts when producing its final trajectory.

**3) Reasoning branches maintain meaningful diversity.** The Diversity score for both models indicates the two branches represent distinct motion hypotheses. This is essential in multi-agent driving scenarios with inherent uncertainty. RL slightly reduces diversity ( $0.412 \rightarrow 0.353$ ), but the branches remain significantly different. In other words, RL makes exploration more targeted towards better proposal quality ( $0.976 \rightarrow 0.961$ ).

Overall, we find that the final action trajectory is tightly aligned with the latent reasoning proposals, yet still achieves clearly lower ADE to the ground truth than the proposals themselves, showing that the model both uses and refines the proposed futures. Compared to the latent CoT model without RL, closed-loop RL further reduces both proposal and final-action errors and strengthens the alignment between proposals and the final decision.

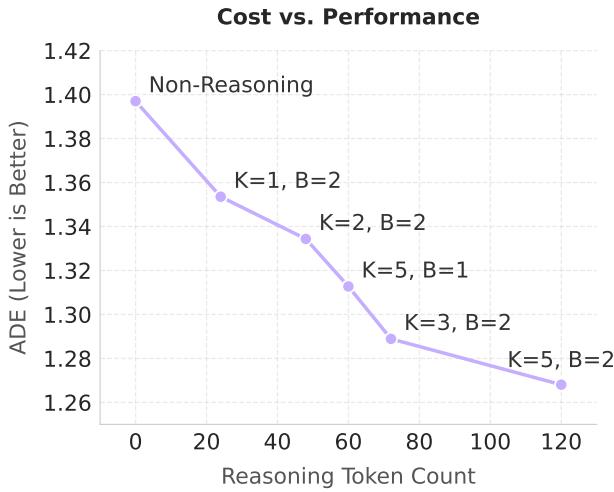


Figure 5. **Efficiency Curve.** We train different variants of LCDrive with different reasoning depth  $K$  and branch factor  $B$ .

## C. Inference Efficiency Study

### C.1. Ablation Study on Reasoning Depth

In this section, we study the trade-off between the reasoning token budget and trajectory accuracy by varying the reasoning depth  $K$  and branch factor  $B$  of LCDrive (GT LWM, Non-RL). For each variant, we construct the CoT supervision target in Stage 1 CoT Cold start stage with different settings of  $K$  and  $B$ . Then, we train the model with teacher forcing with different reasoning depths and branch factors, keeping all other components and hyperparameters fixed across runs. Importantly, we *do not* apply RL fine-tuning and we *do not* use predicted LWM tokens in this study, since our goal here is to quantify the tradeoff of reasoning cost and final action performance of latent CoT.

We then evaluate each model on the validation dataset and compare the performance in Fig. 5. We also compare them with the non-reasoning baseline (LWM<sub>0</sub>-only with GT LWM). The horizontal axis plots the number of reasoning tokens generated per input clip, and the vertical axis shows the resulting ADE (lower is better). We have the following observations:

**1) Latent CoT provides consistent improvements over the baseline** The leftmost point corresponds to the non-reasoning model. Introducing even a minimal amount of latent reasoning (e.g.,  $K=1, B=2$  with 24 tokens) produces a clear reduction in ADE. This demonstrates that a small number of interleaved action-proposal and latent world-model tokens already provides useful counterfactual context for the final trajectory prediction.

**2) Increasing reasoning budget yields meaningful gains** As we increase  $(K, B)$ , performance improves smoothly, indicating that deeper latent reasoning enables

the model to explore more steps into the future and produce better action plans based on that. The largest gains are obtained when moving from shallow reasoning (e.g.,  $K=1, 2$ ) to larger reasoning depth ( $K=3-5$ ). Beyond this range, improvements are smaller but still positive, showing that LCDrive remains effective with different levels of token budgets.

**3) Branching ( $B$ ) leads to complementary improvements to depth ( $K$ )** Branches encourage diverse counterfactual futures. Models with multiple branches (e.g.,  $K=5, B=2$ ) outperform the one with the same depth but fewer branches (e.g.,  $K=5, B=1$ ). This aligns with our diversity analysis: exploring alternative counterfactual futures provides richer reasoning signals for the final policy.

Overall, this curve indicate that latent reasoning offers a highly effective cost-performance tradeoff: a modest reasoning budget (120 tokens) achieves strong trajectory accuracy while remaining relatively cheap. These results demonstrate that LCDrive can flexibly trade inference cost for planning quality. Even lightweight latent CoT substantially enhances the end-to-end driving performance.

### C.2. Inference Cost Analysis

We next compare the inference cost of latent chain-of-thought (Latent CoT) reasoning in LCDrive with a text-based CoT baseline.

**Latent CoT inference cost.** In LCDrive, each reasoning step  $k \in \{1, \dots, K\}$  simulates a 1.0 s future window and produces: (i) 10 discrete action tokens (representing the ego trajectory at 10 Hz), and (ii) 2 latent world model (LWM) tokens. For a model with reasoning depth  $K$  and branch factor  $B$ , the total number of latent reasoning tokens is therefore

$$N_{\text{latent}} \approx (10 + 2) \times K \times B,$$

plus a small constant overhead for the special tokens. At inference time, the inference cost of latent reasoning scales linearly with  $N_{\text{latent}}$ .

**Text CoT baseline cost.** For comparison, we tokenize the text cot reasoning produced by text-CoT baseline and compute the statistics over the validation dataset. Over this dataset we obtain an average length of 71.8 tokens, a 75-th percentile of 80 tokens, and a long tail up to 252 tokens per clip. Thus, a typical text-CoT explanation requires on the order of 70–80 additional tokens at inference time.

From the cost–performance curve in Fig. 5, we find that LCDrive already achieves *significant* improvements over the non-reasoning baseline using only a small, fixed latent budget of roughly 20–60 tokens (e.g., shallow configurations such as  $(K, B) = (1, 2), (2, 2)$ , or  $(3, 2)$ ). These settings use comparable or fewer tokens than typical text CoT, showing that compact latent reasoning is very cost-effective. As we increase the latent reasoning depth and

branch factor, the model consistently achieves better trajectory accuracy, and remains *superior* to the text-CoT baseline (as shown in Table 1 of our paper) when using similar total tokens. This suggests that latent world-model rollouts provide more actionable planning signal per token than free-form natural language reasoning.

**Potential for further latent reasoning.** Our current action tokenizer produces 10 tokens per second of motion. An promising next step is to design a more aggressive motion tokenizer (e.g., fewer tokens per second or multi-step primitives), which would *linearly* reduce the latent reasoning token count for a fixed  $(K, B)$ . Because these tokens are structured and low-entropy compared to text, they are much easier to compress than natural-language CoT, indicating significant room for future latency and cost reductions while preserving the benefits of latent reasoning.