

# DeepThinkVLA: Enhancing Reasoning Capability of Vision-Language-Action Models

Cheng Yin<sup>1,5,6</sup>, Yankai Lin<sup>2,3\*</sup>, Wang Xu<sup>4</sup>, Sikyuen Tam<sup>4</sup>, Xiangrui Zeng<sup>1,5\*</sup>, Zhiyuan Liu<sup>4</sup>, Zhouping Yin<sup>1,5</sup>

<sup>1</sup>Research Center for Advanced Electronics Manufacturing, School of Mechanical Science and Engineering, Huazhong University of Science and Technology

<sup>2</sup>Gaoling School of Artificial Intelligence, Renmin University of China

<sup>3</sup>Beijing Academy of Artificial Intelligence

<sup>4</sup>Department of Computer Science and Technology, Tsinghua University

<sup>5</sup>State Key Laboratory of Intelligent Manufacturing Equipment and Technology, Huazhong University of Science and Technology

<sup>6</sup>Beijing Zhongguancun Academy

yinchenghust@hust.edu.cn, yankailin@ruc.edu.cn, zeng@hust.edu.cn

## Abstract

Does Chain-of-Thought (CoT) reasoning genuinely improve Vision-Language-Action (VLA) models, or does it merely add overhead? Existing CoT-VLA systems report limited and inconsistent gains, yet no prior work has rigorously diagnosed when and why CoT helps robots act. Through systematic experiments, we identify two necessary conditions that must be jointly satisfied for CoT to be effective in VLA: (1) **Decoding Alignment**—CoT and actions must be generated with modality-appropriate mechanisms; forcing both through a single autoregressive decoder is not merely suboptimal but actively harmful, degrading performance by 4.2 percentage points; (2) **Causal Alignment**—CoT must be causally linked to task success via outcome-based optimization; without it, supervised CoT is indistinguishable from no reasoning at all under action-execution-sensitive dynamics shift, exhibiting a 32.0 pp performance drop nearly identical to the 31.6 pp drop of a reasoning-free baseline. Guided by these findings, we build DeepThinkVLA: a hybrid-attention decoder satisfies Condition 1 by pairing causal attention for language with bidirectional attention for parallel action decoding, while a two-stage SFT-then-RL pipeline satisfies Condition 2 by aligning the full reasoning-action chain with sparse task-success rewards. DeepThinkVLA achieves 97.0% success on LIBERO, 79.0% robustness on LIBERO-Plus (vs. 61.6% for  $\pi_0$ -FAST), and 59.3% success on RoboTwin 2.0, exceeding the strongest baseline by 21.7 points. Furthermore, real-robot experiments provide preliminary evidence for the physical applicability of our CoT data construction and hybrid architecture. Our codes are available at <https://github.com/OpenBMB/DeepThinkVLA>.

## 1 Introduction

Vision-Language-Action (VLA) models have driven notable progress in robotic manipulation, enabling tasks such as stacking blocks, opening drawers, and arranging household objects (Huang et al., 2023; Brohan et al., 2023; Yang et al., 2023; Cadene et al., 2024). The dominant paradigm learns a reactive, end-to-end policy that directly maps high-level goals and sensory inputs to low-level motor commands (Chi et al., 2023; Kim et al., 2024; Nvidia et al., 2025). While effective in standard settings, these System-1-style reactive policies (Cui et al., 2025) tend to overfit to training dynamics and struggle to generalize under environ-

\*Corresponding authors

mental perturbations or out-of-distribution (OOD) conditions (Ma et al., 2024; Liu et al., 2024).

A natural remedy is to endow VLAs with the capacity to “think before acting” via Chain-of-Thought (CoT) reasoning (Zawalski et al., 2024; Chen et al., 2025c). Recent work takes initial steps in this direction by fine-tuning VLAs on CoT-annotated embodied data (Zawalski et al., 2024; Lin et al., 2025; Tan et al., 2025a). The premise is appealing: explicit reasoning should decompose the monolithic perception-to-action mapping, thereby bridging the semantic gap and improving generalization. In practice, however, existing CoT-VLA systems report marginal and highly task-dependent gains (Zhao et al., 2025b; Liu et al., 2025b). This raises a fundamental, open question: Does CoT reasoning genuinely participate in robotic decision-making, or is it mere decoration—plausible text that adds latency without improving actions?

In this work, we provide a systematic answer. Through controlled experiments, we identify two necessary conditions that must be jointly satisfied for CoT to be effective in VLA models—and show that violating either renders CoT useless or actively harmful.

**(1) Condition 1: Decoding Alignment.** CoT and actions must be generated with modality-appropriate decoding mechanisms. Language reasoning is inherently sequential and suits autoregressive generation (Xiao et al., 2022); robot actions are high-dimensional vectors whose components can be determined in parallel (Liu et al., 2025c; Kim et al., 2025b; Song et al., 2025). Forcing both through a single autoregressive decoder creates a fundamental conflict: our ablation shows that naively adding CoT to an AR action decoder degrades performance from 85.5% to 81.3% while incurring  $4\times$  latency (Table 6).

**(2) Condition 2: Causal Alignment.** CoT must be causally linked to task success; otherwise, supervised fine-tuning (SFT) produces reasoning that merely imitates the style of expert annotations without influencing action selection. The critical evidence: under OOD dynamics, our SFT-only model suffers a 32.0 pp performance drop—nearly identical to the 31.6 pp drop of the reasoning-free  $\pi_0$ -FAST baseline (Table 8). This means SFT-learned CoT does not participate in adaptation at all. Only after outcome-based reinforcement learning does the drop shrink to 24.4 pp, and masking CoT at inference widens it back to 27.7 pp—indicating that RL alignment makes the CoT pathway action-relevant, turning CoT from a passive narrative into a functional planning signal.

Guided by these two conditions, we design **DeepThinkVLA**, a system in which every architectural and training choice is a direct consequence of satisfying Decoding Alignment and Causal Alignment. For Condition 1, we introduce a hybrid-attention decoder that generates CoT with causal attention and then switches to bidirectional attention for parallel action decoding, resolving the modality mismatch and reducing inference latency. For Condition 2, we employ a two-stage pipeline (Zhang et al., 2024; Kim et al., 2025a; Li et al., 2025): SFT cold-starts the model with foundational reasoning on a synthetically constructed embodied CoT dataset, followed by outcome-based RL with sparse task-success rewards that causally align the full reasoning–action chain. The contribution is not that hybrid decoding, GRPO, or the generic SFT-then-RL recipe is novel in isolation (Lu et al., 2025; Song et al., 2026), but that identifying and jointly satisfying both conditions is what makes CoT work in VLA.

Built on the public  $\pi_0$ -FAST weights (Pertsch et al., 2025), DeepThinkVLA achieves 97.0% average success on LIBERO, setting a new state of the art. On the LIBERO-Plus robustness benchmark, it attains 79.0% overall, exceeding  $\pi_0$ -FAST by 17.4 points. On the high-fidelity RoboTwin 2.0 benchmark, it reaches 59.3% success, surpassing the strongest baseline by 21.7 points with particularly strong long-horizon performance. Controlled OOD experiments indicate that the gains stem from functional reasoning rather than memorized trajectories.

In summary, our key contributions are: **(1)** We identify and empirically validate two necessary conditions—Decoding Alignment and Causal Alignment—for Chain-of-Thought reasoning to be effective in VLA models, and demonstrate that violating either condition makes CoT harmful or decorative. **(2)** We propose DeepThinkVLA, a principled system that jointly satisfies both conditions via a hybrid-attention decoder and a two-stage SFT-then-RL training pipeline. **(3)** We achieve state-of-the-art results on three benchmarks (LIBERO,

LIBERO-Plus, RoboTwin 2.0), with controlled ablations confirming that each condition is necessary for the observed gains. (4) Finally, real-robot experiments provide preliminary evidence for the physical applicability of our CoT data construction and hybrid architecture.

## 2 Related Work

**Vision–Language–Action Models.** Recent progress in VLAs has focused on architectural innovations within the reactive perception-to-action paradigm. Early work such as RT-2 (Brohan et al., 2023) popularized this direction, leading to models that adopt diverse VLM backbones (Black et al., 2025; Nvidia et al., 2025; Hung et al., 2025; Pertsch et al., 2025) and train on large-scale robotic datasets (Walke et al., 2023; Fang et al., 2023; O’Neill et al., 2024; Khazatsky et al., 2024; Wu et al., 2024). Variants include diffusion-based decoders (Black et al., 2025; Liu et al., 2025a), block-parallel decoders with bidirectional attention (Liu et al., 2025c; Kim et al., 2025b; Song et al., 2025), and hierarchical structures separating planning from execution (Belkhale et al., 2024; Cui et al., 2025; Cao et al., 2025; AgiBot-World-Contributors et al., 2025; Intelligence et al., 2025). Across these designs, however, the action-generating policy remains reactive—mapping observations and goals directly to actions without explicit deliberation (Sapkota et al., 2025). None of these works addresses the question of how to effectively integrate explicit reasoning into the action-generation loop, which is the focus of our investigation.

**Embodied Reasoning via Supervised Fine-Tuning.** To move beyond purely reactive mapping, several recent studies endow VLAs with embodied reasoning through SFT on CoT-augmented data (Zawalski et al., 2024; Lin et al., 2025; Tan et al., 2025a; Chen et al., 2025c). This line of work typically constructs CoT annotations for existing embodied datasets (O’Neill et al., 2024), often leveraging stronger cloud-based models (Team et al., 2023), and then fine-tunes open-source VLAs such as OpenVLA (Kim et al., 2024). While providing a first step toward “think before acting,” these methods face two persistent challenges: limited availability of high-quality CoT-annotated embodied data (Wang et al., 2024; Xu et al., 2024; Zhong et al., 2025), and a tendency for SFT to produce shallow memorization of reasoning traces without strong alignment to actions (Zhao et al., 2025a; Liu et al., 2025b). In the broader vision-language domain, SFT can induce pseudo reasoning paths in R1-like LVLMs while RL fosters more adaptive reasoning behavior (Chen et al., 2025a), and carefully curated SFT data can substantially close multimodal reasoning gaps (Lin et al., 2026)—indicating that SFT effectiveness is tightly coupled to data quality. Our Condition 2 targets a complementary, VLA-specific failure mode: unlike these settings, where the reasoning trace and final answer share a single textual stream, a VLA must causally link natural-language CoT to heterogeneous action outputs, so SFT alone cannot guarantee that the generated text influences action selection at any fixed data quality. Beyond SFT, recent RL-based efforts either optimize reasoning-free VLA backbones with outcome rewards (Lu et al., 2025) or train LVLMs with verifiable rewards for affordance perception and image-space trajectory prediction without closed-loop action generation (Song et al., 2026); our contribution is accordingly not the generic SFT-then-RL recipe, but the alignment of explicit reasoning with VLA action generation. Crucially, no prior work has systematically investigated under what conditions CoT reasoning transitions from decorative text to a functional decision-making signal. Our work fills this gap by identifying and validating two necessary conditions for effective CoT in VLA models.

## 3 DeepThinkVLA

Guided by these two necessary conditions, we design DeepThinkVLA so that every architectural and training choice directly serves Decoding Alignment or Causal Alignment. We first formalize the “think before acting” paradigm as a probabilistic decomposition (Section 3.1). We then present the hybrid-attention decoder, the natural architectural response to Condition 1 (Section 3.2). Finally, we detail the two-stage training pipeline—SFT followed by outcome-based RL—which is the systematic implementation of Condition 2 (Section 3.3).

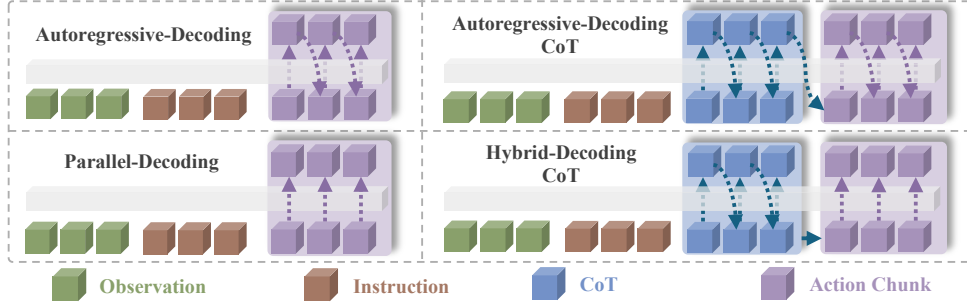


Figure 1: **Comparison of VLA architectures.** DeepThinkVLA introduces a hybrid design that pairs autoregressive CoT reasoning with parallel action decoding.

### 3.1 Problem Formulation

Standard VLA policies learn a direct mapping from visual observations ( $V$ ) and language instructions ( $L$ ) to a sequence of actions ( $A$ ). Instead, our work adopts the principle of “think before acting”. We implement this principle by introducing a latent reasoning variable, the Chain-of-Thought ( $R$ ). This approach decomposes the problem. Rather than modeling the direct policy, we model the joint probability of reasoning and then acting:

$$P(A, R|V, L) = P(A|V, L, R) P(R|V, L). \quad (1)$$

The advantage of this decomposition is twofold. First, learning  $P(R|V, L)$  is highly efficient. Most VLAs are built upon large VLM backbones, which already contain rich semantic and reasoning knowledge. Fine-tuning such models on a relatively small set of synthetically generated embodied CoT data is often sufficient to adapt their reasoning capability to the robotics domain.

Second, learning  $P(A|V, L, R)$  becomes significantly simpler than directly modeling  $P(A|V, L)$ . The CoT serves as an explicit, step-by-step plan that disambiguates high-level instructions  $L$ , turning the ill-posed, one-to-many mapping into a constrained and well-specified mapping from a reasoning step to its corresponding motor action. This principled factorization also enables emergent self-correction behaviors, as illustrated in Figure 8.

### 3.2 Hybrid Architecture: Satisfying Decoding Alignment

Condition 1 (Decoding Alignment) requires that CoT and actions be generated with mechanisms suited to their respective modalities. To implement the factorized policy from Eq. 1 under this constraint, we design a hybrid architecture that aligns the decoding mechanism with the intrinsic properties of each modality. The core of our design is a dynamic attention mode within a single decoder: For CoT Generation ( $P(R|V, L)$ ): The decoder employs standard autoregressive causal attention. This respects the sequential nature of language, where each reasoning token is generated based on its predecessors. For Action Generation ( $P(A|V, L, R)$ ): After generating the CoT, the attention mechanism switches to bidirectional (non-causal) attention. This allows the model to process the entire action specification jointly and decode the action vector in parallel, acknowledging that different dimensions of a motor command (e.g., translation, rotation) are often determined concurrently.

Beyond resolving this core modality mismatch, the parallel decoding of actions yields a critical practical advantage: a significant reduction in inference latency. This speedup is the key enabler for our subsequent training stage. While standard autoregressive models are prohibitively slow for the massive number of rollouts required by on-policy RL, our architecture’s high-throughput action generation makes large-scale online fine-tuning computationally tractable. Further architectural details are summarized in Figure 1; a token-level specification of the hybrid decoder—action-slot representation, embedding zeroing, and the hybrid attention mask—is provided in Appendix A.3.

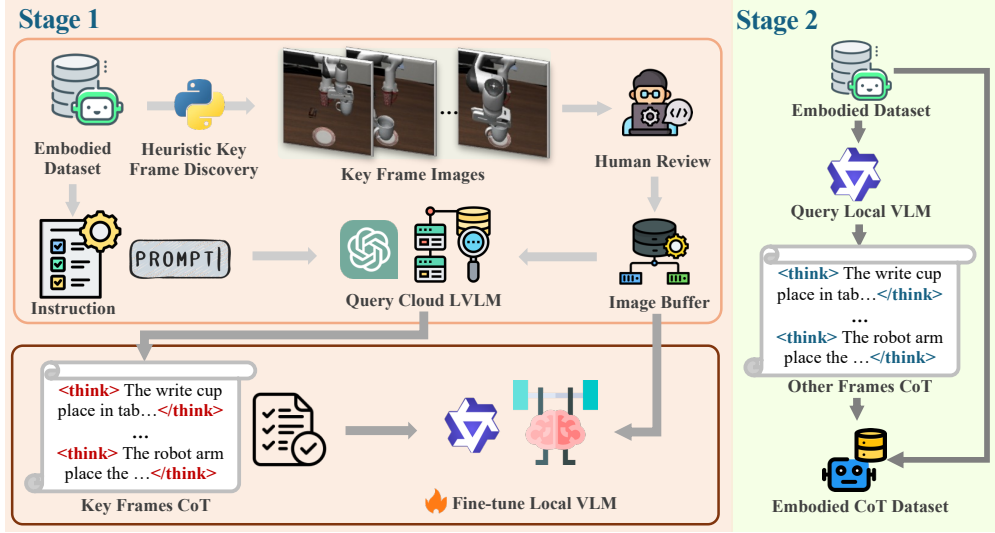


Figure 2: **Two-stage pipeline for embodied CoT dataset construction.** Stage 1 extracts keyframes via gripper state changes and queries a cloud LLM; Stage 2 fine-tunes a local VLM on the keyframe CoT to annotate intermediate frames.

### 3.3 RL Training Pipeline: Satisfying Causal Alignment

Condition 2 (Causal Alignment) demands that CoT be causally linked to task success rather than merely imitating expert annotation style. With our efficient architecture in place, we train DeepThinkVLA using a two-stage pipeline designed to first instill foundational reasoning and then establish the causal link through outcome-based optimization.

**SFT Cold-Start for Foundational Reasoning.** SFT cold-start is designed to equip the model with a foundational CoT reasoning capability. This phase requires a specific supervision format comprising complete  $(V, L, R, A)$  sequences. However, most existing large-scale embodied datasets lack explicit CoT annotations and instead provide only  $(V, L, A)$  tuples. To address this critical data gap, we developed a scalable, two-stage data augmentation pipeline that generates high-quality CoT annotations, as illustrated in Figure 2.

Our pipeline is optimized for both annotation quality and cost-efficiency. In stage 1, we identify semantically significant keyframes within each trajectory by detecting changes in the gripper state, which often indicate subtask boundaries. For these keyframes, CoT annotations are obtained by querying a powerful, general-purpose cloud-based Vision-Language Model (VLM) (see prompt in Appendix Fig. 4). In stage 2, to efficiently annotate the numerous intermediate frames, we fine-tune a smaller, locally-deployed VLM on the high-quality keyframe annotations obtained in Stage 1. This specialized model then automatically generates CoT annotations for the transitional frames. To ensure data fidelity, we apply schema checks to filter malformed outputs and enforce temporal consistency, resulting in a uniform embodied CoT dataset suitable for SFT; text-level quality audits of the resulting annotations are reported in Appendix A.7.

**Learning Reasoning and Action via RL.** While SFT provides a strong imitation learning foundation, it cannot adapt to novel scenarios or optimize beyond the sub-optimal trajectories present in the static dataset. To overcome these limitations and truly learn a policy that maximizes task success, we introduce a second training stage based on online Reinforcement Learning. We use an outcome-based reward to jointly optimize the entire reasoning-action sequence towards the singular goal of task completion (Figure 3).

We cast the online RL stage as a policy optimization problem: starting from the initial SFT policy, we aim to maximize the expected outcome-based reward through interactive learning. We adopt an on-policy policy gradient algorithm with a clipped surrogate objective akin

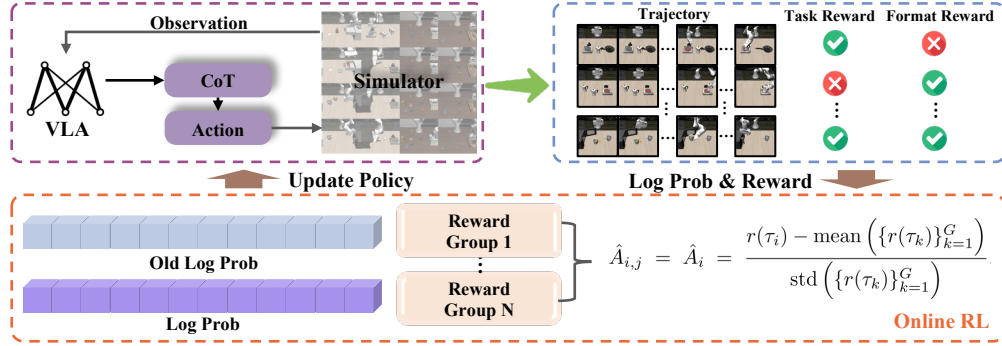


Figure 3: **RL stage with grouped credit assignment.** Trajectories are collected via simulator rollouts; rewards are group-normalized into token-level advantages and optimized with a clipped surrogate objective and KL regularization.

Table 1: **Performance on the LIBERO simulation benchmark** (success rate %, 50 trials per task). All top baselines use wrist cameras; DeepThinkVLA uses vision-language only. **Bold** = best per suite.

| Category          | Method                               | Object      | Spatial     | Goal        | Long        | Average     |
|-------------------|--------------------------------------|-------------|-------------|-------------|-------------|-------------|
| AR-Decoding       | TraceVLA (Zheng et al., 2024)        | 85.2        | 84.6        | 75.1        | 54.1        | 74.8        |
|                   | Octo (Team et al., 2024)             | 85.7        | 78.9        | 84.6        | 51.1        | 75.1        |
|                   | OpenVLA (Kim et al., 2024)           | 88.4        | 84.7        | 79.2        | 53.7        | 76.5        |
|                   | SpatialVLA (Qu et al., 2025)         | 89.9        | 88.2        | 78.6        | 55.5        | 78.1        |
|                   | GRAPE (Zhang et al., 2024)           | 91.2        | 87.6        | 82.2        | 55.8        | 79.2        |
|                   | NORA (Hung et al., 2025)             | 95.4        | 92.2        | 89.4        | 74.6        | 87.9        |
|                   | VLA-RL (Lu et al., 2025)             | 91.8        | 90.2        | 82.2        | 59.8        | 81.0        |
|                   | $\pi_0$ -FAST (Pertsch et al., 2025) | 96.8        | 96.4        | 88.6        | 60.2        | 85.5        |
| Diffusion         | UniVLA (Bu et al., 2025)             | 96.8        | 96.5        | 95.6        | 92.0        | 95.2        |
|                   | Diffusion Policy (Chi et al., 2023)  | 92.5        | 78.3        | 68.3        | 50.5        | 72.4        |
| Parallel-Decoding | $\pi_0$ (Black et al., 2025)         | 98.8        | <b>96.8</b> | 95.8        | 85.2        | 94.2        |
|                   | CoT-VLA-7B (Zhao et al., 2025b)      | 91.6        | 87.5        | 87.6        | 69.0        | 81.1        |
| Hybrid-Decoding   | OpenVLA-OFT (Kim et al., 2025b)      | 92.7        | 91.3        | 90.5        | 86.5        | 90.3        |
|                   | DeepThinkVLA (Qwen3-VL)              | 98.6        | 92.6        | 96.2        | 92.0        | 94.9        |
| Hybrid-Decoding   | DeepThinkVLA ( $\pi_0$ -FAST)        | <b>99.0</b> | 96.6        | <b>96.4</b> | <b>96.2</b> | <b>97.0</b> |

to PPO. The VLA collects trajectories from the environment, computes advantages, and updates its policy accordingly.

Formally, during the RL rollout, the state at each step  $t$  is defined as  $s_t = [o_t^{\text{vis}}, \ell_{\text{task}}]$ , where  $o_t^{\text{vis}}$  denotes the visual observation and  $\ell_{\text{task}}$  is the task instruction. Given the state input, the VLA outputs  $\mathcal{A}_t = [a_t^{\text{cot}}, a_t^{\text{robot}}]$ , where the reasoning tokens  $a_t^{\text{cot}}$  are generated autoregressively, and the action tokens  $a_t^{\text{robot}} \in \mathbb{R}^{h \times d}$  are decoded in parallel. Here,  $h$  denotes the action chunk size and  $d$  corresponds to the robot’s control dimension (e.g.,  $d = 7$  for a 6-DoF manipulator plus gripper control).

Let  $\pi_\theta$  denote the current policy. A trajectory  $\tau$  sampled from the old policy  $\pi_{\theta_{\text{old}}}$  is defined as  $\tau = [(s_0, \mathcal{A}_0), (s_1, \mathcal{A}_1), \dots, (s_T, \mathcal{A}_T)]$ . The reward function  $\mathcal{R}(\tau)$  is sparse, awarded only at the end of the trajectory based on a verifiable task completion signal  $\mathcal{I}_{\text{success}}$ . No intermediate reward is given for the semantics of the reasoning trace. A small format-regularization reward  $\mathcal{I}_{\text{format}}$  is added to prevent stylistic drift. Hence, the reward function is defined as

$$\mathcal{R}(\tau) = \alpha_s \cdot \mathcal{I}_{\text{success}} + \alpha_f \cdot \mathcal{I}_{\text{format}},$$

$$\mathcal{I}_{\text{success}} = \begin{cases} 1, & \text{if task success,} \\ 0, & \text{otherwise,} \end{cases} \quad \mathcal{I}_{\text{format}} = \begin{cases} 1, & \text{if CoT format correct,} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

where  $\alpha_s$  and  $\alpha_f$  are weighting coefficients. Then the token-level clipped surrogate objective is:

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta_{\text{old}}}} \left[ \sum_{j=1}^N \min \left( \omega_j(\theta) \hat{A}_j, \text{clip}(\omega_j(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_j \right) \right], \quad (3)$$

where  $N = |\mathcal{A}_t| \times T$  denotes the total number of tokens in a trajectory  $\tau$ .  $\omega_j(\theta) = \frac{\pi_{\theta}(a_j|s_t, a_{<j})}{\pi_{\theta_{\text{old}}}(a_j|s_t, a_{<j})}$  and  $\hat{A}_j$  are the importance ratio and the advantage for token  $a_j \in [\mathcal{A}_0, \dots, \mathcal{A}_T]$  within trajectory  $\tau$ , respectively.

To propagate the sparse, outcome-based reward  $\mathcal{R}(\tau)$  to each token prediction, we adopt the credit assignment strategy from GRPO (Shao et al., 2024). A group of  $G$  trajectories is collected for each task prompt and their rewards are standardized to compute a shared advantage value for all tokens within a given trajectory. For token  $j$  in trajectory  $i$ , the advantage is:

$$\hat{A}_{i,j} = \frac{\mathcal{R}(\tau_i) - \text{mean}(\{\mathcal{R}(\tau_k)\}_{k=1}^G)}{\text{std}(\{\mathcal{R}(\tau_k)\}_{k=1}^G)}. \quad (4)$$

This relative credit assignment encourages the model to prefer reasoning and action sequences that lead to better-than-average outcomes, effectively selecting for more functional thought processes.

Combining the clipped surrogate objective with the GRPO-style advantage and a KL-divergence penalty to the original SFT policy  $\pi_{\text{ref}}$ , which prevents catastrophic forgetting, our final objective is:

$$\begin{aligned} \mathcal{J}_{\text{final}}(\theta) = \mathbb{E}_{\substack{s \sim \text{env}, \\ \{\tau_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}}} & \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{N} \sum_{j=1}^N \min \left( \omega_{i,j}(\theta) \hat{A}_{i,j}, \text{clip}(\omega_{i,j}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_{i,j} \right) \right. \\ & \left. - \beta \text{KL}(\pi_{\theta}(\cdot | s) \parallel \pi_{\text{ref}}(\cdot | s)) \right], \end{aligned} \quad (5)$$

where  $\omega_{i,j}(\theta)$  is the importance ratio for token  $a_j \in [\mathcal{A}_0, \dots, \mathcal{A}_T]$  in trajectory  $i$ . By maximizing  $\mathcal{J}_{\text{final}}$ , the VLA simultaneously refines its reasoning and action abilities, with both aligned toward the singular goal of maximizing the final task success rate.

## 4 Experiments

### 4.1 Experimental Setup

**Implementation Details.** DeepThinkVLA is initialized from the public  $\pi_0$ -FAST weights (Pertsch et al., 2025). We refactor the baseline policy with our hybrid-attention decoder (Sec. 3.2), yielding a 2.9B parameter model. Training proceeds in two stages. First, an SFT cold-start is performed on our embodied CoT dataset. A hybrid attention mask supervises CoT tokens with causal attention and action tokens with bidirectional attention in a single forward pass, optimized by token-level cross-entropy loss. Second, we apply online RL to align generated CoT with action execution, using outcome-based rewards and GRPO-style grouped credit assignment. Additional training details, including dataset statistics, hyperparameters, and inference settings, are provided in Appendix A.2.

**Benchmarks.** We evaluate DeepThinkVLA on three simulation benchmarks and a set of real-robot tasks to ensure a comprehensive assessment of foundational skills, long-horizon robustness, and physical transferability. LIBERO (Liu et al., 2023): A standard language-conditioned manipulation benchmark containing four suites (Object, Spatial, Goal, and Long). Evaluation is conducted under 50 randomized initial conditions for each task.

LIBERO-Plus (Fei et al., 2025): A robustness benchmark that introduces controlled perturbations across seven dimensions (e.g., camera viewpoints, robot initial states, and language rewrites) and reports per-dimension success rates along with an overall score. RoboTwin

Table 2: **Performance on RoboTwin 2.0** (success rate %). DeepThinkVLA achieves **59.3%** overall, exceeding  $\pi_0$ -FAST by **+21.7** points, with the largest gains on long-horizon tasks.

| Short Horizon Tasks (100–130 Steps)                                                                                  |                 |                   |                   |                     |                 |
|----------------------------------------------------------------------------------------------------------------------|-----------------|-------------------|-------------------|---------------------|-----------------|
| Model                                                                                                                | Lift Pot        | Beat Hammer Block | Pick Dual Bottles | Place Phone Stand   | Avg             |
| $\pi_0$ (Black et al., 2025)                                                                                         | 51.0            | 59.0              | 50.0              | 22.0                | 45.5            |
| RDT (Liu et al., 2025a)                                                                                              | 45.0            | 22.0              | 18.0              | 13.0                | 24.5            |
| OpenVLA-OFT (Kim et al., 2025b)                                                                                      | 10.1            | 28.1              | 29.7              | 17.1                | 21.3            |
| $\pi_0$ -FAST (Pertsch et al., 2025)                                                                                 | 30.0            | 38.0              | 25.0              | 16.0                | 27.3            |
| <b>DeepThinkVLA</b>                                                                                                  | <b>62.0</b>     | <b>73.0</b>       | <b>61.0</b>       | <b>24.0</b>         | <b>55.0</b>     |
| $\Delta$                                                                                                             | $\uparrow 32.0$ | $\uparrow 35.0$   | $\uparrow 36.0$   | $\uparrow 8.0$      | $\uparrow 27.8$ |
| Medium Horizon Tasks (150–230 Steps)                                                                                 |                 |                   |                   |                     |                 |
| Model                                                                                                                | Move Can Pot    | Place A2B Left    | Place Empty Cup   | Handover Mic        | Avg             |
| $\pi_0$ (Black et al., 2025)                                                                                         | 41.0            | 38.0              | 60.0              | 96.0                | 58.8            |
| RDT (Liu et al., 2025a)                                                                                              | 33.0            | 21.0              | 42.0              | 95.0                | 47.8            |
| OpenVLA-OFT (Kim et al., 2025b)                                                                                      | 28.1            | 37.5              | 77.3              | 45.3                | 47.1            |
| $\pi_0$ -FAST (Pertsch et al., 2025)                                                                                 | 34.0            | 36.0              | 54.0              | 83.0                | 51.8            |
| <b>DeepThinkVLA</b>                                                                                                  | <b>52.0</b>     | <b>38.0</b>       | <b>83.0</b>       | <b>88.0</b>         | <b>65.3</b>     |
| $\Delta$                                                                                                             | $\uparrow 18.0$ | $\uparrow 2.0$    | $\uparrow 29.0$   | $\uparrow 5.0$      | $\uparrow 13.5$ |
| Long (280–320 Steps) & Extra Long Horizon Tasks (450–650 Steps)                                                      |                 |                   |                   |                     |                 |
| Model                                                                                                                | Handover Block  | Stack Bowls Two   | Blocks Rank Rgb   | Put Bottles Dustbin | Avg             |
| $\pi_0$ (Black et al., 2025)                                                                                         | 39.0            | 53.0              | 45.0              | 36.0                | 43.3            |
| RDT (Liu et al., 2025a)                                                                                              | 26.0            | 42.0              | 17.0              | 26.0                | 27.8            |
| OpenVLA-OFT (Kim et al., 2025b)                                                                                      | 33.1            | 40.6              | 70.2              | 42.2                | 46.5            |
| $\pi_0$ -FAST (Pertsch et al., 2025)                                                                                 | 32.0            | 48.0              | 28.0              | 27.0                | 33.8            |
| <b>DeepThinkVLA</b>                                                                                                  | <b>43.0</b>     | <b>62.0</b>       | <b>77.0</b>       | <b>49.0</b>         | <b>57.8</b>     |
| $\Delta$                                                                                                             | $\uparrow 11.0$ | $\uparrow 14.0$   | $\uparrow 49.0$   | $\uparrow 22.0$     | $\uparrow 24.0$ |
| Overall Avg RDT: 33.3 $\pi_0$ : 49.2 OpenVLA-OFT: 38.3 $\pi_0$ -FAST: 37.6 <b>DeepThinkVLA: 59.3</b> $\uparrow 21.7$ |                 |                   |                   |                     |                 |

2.0 (Chen et al., 2025b): A high-fidelity digital twin benchmark featuring complex contact-rich manipulation and significantly longer task horizons. This is introduced to rigorously test the model’s capability to maintain reasoning context over extended execution periods. Real-Robot Tasks: To assess the preliminary physical applicability of our CoT data construction and hybrid architecture, we additionally evaluate on three physical manipulation tasks (Stack Bowls Two, Handover Block, Blocks Rank RGB) using human-teleoperated demonstrations and our CoT annotation pipeline (details in Section 4.3). Performance is measured by the average success rate. Additionally, to provide concrete examples of the model’s reasoning process and error recovery behaviors (Case Studies), we include a detailed qualitative analysis in Appendix A.8.

**Baselines.** We evaluate DeepThinkVLA against a wide spectrum of recent VLA systems, covering autoregressive SFT models, diffusion-based approaches, parallel-decoding methods and commercial baselines such as  $\pi_0$  and  $\pi_0$ -FAST. A detailed description of all baselines is provided in Appendix A.4.

## 4.2 Main Results

**Results on LIBERO.** The main results on the LIBERO benchmark are shown in Table 1. DeepThinkVLA establishes a new state-of-the-art by achieving the highest average success rate (SR) of 97.0%. Our model shows exceptional proficiency across all categories, particularly in Object (99.0%) and long-horizon tasks (96.2%), significantly outperforming leading diffusion models like  $\pi_0$  (Object: 98.8%, Long: 85.2%) and autoregressive baselines like UniVLA (Object: 96.8%, Long: 92.0%).

Notably, all competitive baselines in Table 1 utilize wrist camera inputs, while DeepThinkVLA achieves this performance using a pure vision-language setup without proprioceptive state.

**Results on RoboTwin 2.0.** On the high-fidelity RoboTwin 2.0 benchmark (Table 2), DeepThinkVLA achieves 59.3% average success, outperforming  $\pi_0$ -FAST (37.6%) by +21.7 points. The gains are most pronounced on long and extra-long horizon tasks (57.8% vs. 33.8%), confirming that explicit reasoning helps maintain context over extended execution.

Table 3: **Zero-shot robustness on LIBERO-Plus** (seven perturbation dimensions, no adaptation). OFT\_w = wrist-camera variant; OFT\_m = mixed-data variant. **Bold** = best per dimension.

| Model                                         | Perturbation Dimensions |             |             |             |             |             |             | Total       |
|-----------------------------------------------|-------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                                               | Camera                  | Robot-Init  | Language    | Light       | Background  | Noise       | Layout      |             |
| $\pi_0$ (Black et al., 2025)                  | 13.8                    | 6.0         | 58.8        | 85.0        | 81.4        | 79.0        | 68.9        | 53.6        |
| UniVLA (Bu et al., 2025)                      | 1.8                     | <b>46.2</b> | 69.6        | 69.0        | 81.0        | 21.2        | 31.9        | 42.9        |
| WorldVLA (Cen et al., 2025)                   | 0.1                     | 27.9        | 41.6        | 43.7        | 17.1        | 10.9        | 38.0        | 25.0        |
| OpenVLA (Kim et al., 2024)                    | 0.8                     | 3.5         | 23.0        | 8.1         | 34.8        | 15.2        | 28.5        | 15.6        |
| OpenVLA-OFT (Kim et al., 2025b)               | 56.4                    | 31.9        | 79.5        | 88.7        | 93.3        | 75.8        | 74.2        | 69.6        |
| OpenVLA-OFT_w                                 | 10.4                    | 38.7        | 70.5        | 76.8        | <b>93.6</b> | 49.9        | 69.9        | 55.8        |
| OpenVLA-OFT_m                                 | 55.6                    | 21.7        | 81.0        | 92.7        | 91.0        | 78.6        | 68.7        | 67.9        |
| NORA-long (Hung et al., 2025)                 | 2.2                     | 37.0        | 65.1        | 45.7        | 58.6        | 12.8        | 62.1        | 39.0        |
| RIPT-VLA (Tan et al., 2025b)                  | 55.2                    | 31.2        | 77.6        | 88.4        | 91.6        | 73.5        | 74.2        | 68.4        |
| $\pi_0$ -FAST (Pertsch et al., 2025)          | 65.1                    | 21.6        | 61.0        | 73.2        | 73.2        | 74.4        | 68.8        | 61.6        |
| DeepThinkVLA (Qwen3-VL)                       | 63.7                    | 38.8        | 81.5        | <b>94.7</b> | 92.4        | 89.8        | 77.8        | 77.0        |
| <b>DeepThinkVLA (<math>\pi_0</math>-FAST)</b> | <b>88.5</b>             | <b>40.5</b> | <b>84.5</b> | 90.0        | 75.3        | <b>94.4</b> | <b>79.9</b> | <b>79.0</b> |

**Results on LIBERO-Plus.** As shown in Table 3, DeepThinkVLA achieves 79.0% overall on the seven-dimension robustness benchmark, outperforming  $\pi_0$ -FAST by +17.4 points. The gains are broad-based: Camera +23.4, Language +23.5, Noise +20.0, confirming generalization under substantial distribution shifts.

### 4.3 Real-Robot Experiments

As a preliminary physical-applicability check, we deploy DeepThinkVLA on an AGILEX ALOHA bimanual platform for three manipulation tasks (Stack Bowls Two, Handover Block, Blocks Rank RGB): teleoperated demonstrations are annotated with the same two-stage CoT pipeline (Section 3.3), and the model is fine-tuned via SFT only—performing online RL on hardware would require many unsafe trial-and-error rollouts, so claims about RL-based causal alignment remain grounded in simulation. As shown in Table 4, DeepThinkVLA attains 45% average success over 20 trials per task. This figure is best compared against the harder, contact-rich RoboTwin 2.0 simulation results on the corresponding task categories (62.0% / 43.0% / 77.0%; Table 2) rather than near-saturated LIBERO: the physical results are lower but remain within the same order of difficulty, showing that the pipeline can be instantiated on hardware and achieve nontrivial closed-loop execution under real camera, calibration, contact, and actuation noise. This evidence supports the preliminary physical applicability of the CoT data construction and hybrid action decoder; it does not validate the full SFT-then-RL pipeline in the physical domain. Task definitions and further discussion are provided in Appendix A.2.

Table 4: **Real-robot results** (% , 20 trials).

| Task        | Stack Bowls | Handover Block | Blocks Rank RGB | Avg |
|-------------|-------------|----------------|-----------------|-----|
| Performance | 55%         | 45%            | 35%             | 45% |

### 4.4 Ablation Studies

**Backbone Generality.** To test whether our two conditions are architecture-specific or general principles, we apply the same hybrid-decoding design and CoT annotation pipeline to Qwen3-VL (Bai et al., 2025)—a VLM without any embodied pretraining. As shown in Tables 1 and 3, DeepThinkVLA (Qwen3-VL) achieves 94.9% on LIBERO and 77.0% on LIBERO-Plus zero-shot, both substantially above all baselines except the  $\pi_0$ -FAST-based variant. This confirms that the two conditions—not the choice of pretrained backbone—are the primary drivers of effective embodied reasoning. A model without any robotic pretraining can still acquire strong manipulation and robustness through proper decoding alignment and CoT construction. The gains are likewise not driven by sensor modalities: even without a wrist camera, DeepThinkVLA reaches 86.0% on LIBERO, already surpassing the  $\pi_0$ -FAST baseline (85.5%; Appendix A.5).

Table 5: **Same-architecture w/o-CoT control on LIBERO** (success rate %). The action decoder and RL procedure are identical across rows; only the CoT pathway differs. RL alone yields +1.1 pp over the w/o-CoT SFT model, while retaining CoT under the identical RL procedure adds a further +3.2 pp, with the largest gains on Goal (+4.6 pp) and Long (+6.0 pp).

| Method                              | Object      | Spatial     | Goal        | Long        | Avg.        |
|-------------------------------------|-------------|-------------|-------------|-------------|-------------|
| DeepThinkVLA (w/o CoT, SFT-Only)    | 97.2        | 96.4        | 89.6        | 87.8        | 92.7        |
| DeepThinkVLA (w/o CoT, RL-Aligned)  | 97.2        | 96.2        | 91.8        | 90.2        | 93.8        |
| DeepThinkVLA (Full CoT, RL-Aligned) | <b>99.0</b> | <b>96.6</b> | <b>96.4</b> | <b>96.2</b> | <b>97.0</b> |

**Decoding Alignment.** To directly validate Condition 1, we apply identical embodied CoT supervision to the fully autoregressive  $\pi_0$ -FAST model: naively decoding CoT and actions through a single AR stream drops the LIBERO average from 85.5% to 81.3% while incurring  $4\times$  inference latency, whereas the hybrid decoder reaches 96.8% with parallel action decoding (Table 6 in Appendix A.6)—a +15.5 pp swing attributable solely to matching each modality with its appropriate decoding mechanism. The parallel action pass also keeps inference at  $0.175\times$ – $1.4\times$  of the baseline (vs.  $4.0\times$  for AR-CoT), which is what makes the massive on-policy rollouts of the RL stage computationally tractable.

**Causal Alignment via RL.** Applying RL improves LIBERO-Long from 94.2% to 96.2% (Figure 7 in Appendix A.5), with a more pronounced +6.8% gain on the harder RoboTwin 2.0 (Appendix A.9). The controlled OOD experiment (Table 8 in Appendix A.5) provides the strongest evidence: SFT-only suffers a 32.0 pp drop under Joint-Limit dynamics—nearly identical to the reasoning-free  $\pi_0$ -FAST (31.6 pp)—while RL-aligned DeepThinkVLA limits the drop to 24.4 pp. Masking CoT widens it to 27.7 pp, indicating that RL alignment makes the CoT pathway action-relevant rather than decorative. A same-architecture control that removes CoT throughout SFT, RL, and inference (Table 5) further separates the contribution of the CoT pathway from that of the decoder and RL stage: RL without CoT reaches 93.8% on LIBERO, while retaining CoT under the identical RL procedure reaches 97.0% (+3.2 pp, with +4.6 pp on Goal and +6.0 pp on Long). We therefore state Condition 2 precisely: Causal Alignment is required for CoT to remain functional in *action-execution-sensitive* OOD settings; under perception/instruction-side shifts, RL alignment preserves the broad zero-shot robustness of the SFT model (79.1% vs. 79.0% overall on LIBERO-Plus; Table 7 in Appendix A.5). Further architectural ablations (Hybrid vs. AR) and CoT semantic analyses are provided in Appendix A.6, and text-level quality audits of demonstration and deployment-time CoTs in Appendix A.7.

## 5 Conclusions

We identified two necessary conditions for CoT reasoning to be effective in VLA models—**Decoding Alignment** and **Causal Alignment**—and showed that violating either renders CoT harmful or decorative: misaligned decoding actively degrades performance, while Causal Alignment is required for CoT to remain functional in action-execution-sensitive OOD settings. DeepThinkVLA jointly satisfies both conditions via a hybrid-attention decoder and SFT-then-RL training, achieving state-of-the-art results on LIBERO, LIBERO-Plus, and RoboTwin 2.0. Backbone ablations on Qwen3-VL confirm that the conditions, not the pretrained weights, drive the gains. Real-robot experiments provide preliminary evidence for the physical applicability of the CoT data construction and hybrid architecture. The main limitations are that the physical evidence remains SFT-only and small-scale, and that our causal analysis is functional rather than mechanistic; safely extending online RL to hardware and probing the internal mechanism are important next steps. We believe this two-condition framework offers a useful lens for future embodied reasoning research; as one such direction, Appendix A.10 presents a preliminary exploration of augmenting language CoT with explicit spatial reasoning.

## Acknowledgement

We sincerely thank all the anonymous reviewers and (S)ACs for their constructive comments and helpful suggestions. This work was supported by Beijing Major Science and Technology Project under Contract no.Z251100008125046, the National Natural Science Foundation of China under Grant 52188102 and the Hubei Province Key Research and Development Project (No. 2025BEB008)

## Ethics Statement

This work develops methods to improve robotic manipulation through explicit reasoning in controlled environments. All real-robot experiments are safely conducted on an AGILEX ALOHA platform under direct human supervision using benign objects. Our training relies strictly on public benchmarks (LIBERO, RoboTwin 2.0) and consensual laboratory demonstrations, involving no sensitive or personal data. While the reinforcement learning stage incurs computational costs across an 8-GPU cluster, we mitigate this environmental impact through an efficient hybrid-decoding architecture and parallelized rendering. Ultimately, by rigorously validating the functional role of reasoning in Vision–Language–Action models, we aim to contribute to the development of safer, more transparent, and interpretable autonomous systems.

## References

- AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, X. Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *IEEE/RJS International Conference on Intelligent RObots and Systems*, pp. 3549–3556, 2025. doi: 10.1109/IROS60139.2025.11247088.
- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Ke-qin Chen, Xiong-Hui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Rongyao Fang, Chang Gao, et al. Qwen3-vl technical report. *arXiv.org*, 2025. doi: 10.48550/arXiv.2511.21631.
- Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quan Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, 2024. doi: 10.15607/rss.2024.xx.049.
- Kevin Black, Noah Brown, Danny Driess, A. Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. *pi\_0*: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems XXI*. Robotics: Science and Systems Foundation, 2025. doi: 10.15607/rss.2025.xxi.010.
- Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, K. Choromanski, Tianli Ding, Danny Driess, Kumar Avinava Dubey, Chelsea Finn, Peter R. Florence, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023. doi: 10.48550/arXiv.2307.15818.
- Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *Robotics: Science and Systems XXI*, 2025. doi: 10.15607/rss.2025.xxi.014.
- Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, et al. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. *arXiv preprint*, 2024.
- Mingyu Cao, Huajie Tan, Yuheng Ji, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, Yi Han, Yingbo Tang, et al. Robobrain 2.0 technical report. *arXiv.org*, 2025. doi: 10.48550/arXiv.2507.02029.

- Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv.org*, 2025. doi: 10.48550/arXiv.2506.21539.
- Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *Trans. Mach. Learn. Res.*, 2025a. doi: 10.48550/arXiv.2504.11468.
- Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianlian Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv.org*, 2025b. doi: 10.48550/arXiv.2506.18088.
- William Chen, Suneel Belkhale, Suvir Mirchandani, Oier Mees, Danny Driess, Karl Pertsch, and Sergey Levine. Training strategies for efficient embodied reasoning. *arXiv.org*, 2025c. doi: 10.48550/arXiv.2505.08243.
- Cheng Chi, S. Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, B. Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Robotics: Science and Systems Conference*, pp. 02783649241273668, 2023. doi: 10.1177/02783649241273668.
- Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, et al. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. *arXiv.org*, 2025. doi: 10.48550/arXiv.2505.03912.
- Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. In *IEEE International Conference on Robotics and Automation*, pp. 653–660. IEEE, 2023. doi: 10.1109/ICRA57147.2024.10611615.
- Senyu Fei, Siyin Wang, Junhao Shi, Z. G. Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv.org*, 2025. doi: 10.48550/arXiv.2510.13626.
- Wenlong Huang, F. Xia, Ted Xiao, Harris Chan, Jacky Liang, Peter R. Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782. PMLR, 2023. doi: 10.48550/arXiv.2207.05608.
- Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. Nora: A small open-sourced generalist vision language action model for embodied tasks. *arXiv.org*, 2025. doi: 10.48550/arXiv.2504.19854.
- Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, A. Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al.  $\pi 0$ . 5: a vision-language-action model with open-world generalization. *arXiv.org*, 1(2):3, 2025. doi: 10.48550/arXiv.2504.16054.
- Alexander Khazatsky, Karl Pertsch, S. Nair, Ashwin Balakrishna, S. Dasari, Siddharth Karamcheti, Soroush Nasiriany, M. K. Srirama, L. Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems XX*. Robotics: Science and Systems Foundation, 2024. doi: 10.15607/rss.2024.xx.120.
- Dongyoung Kim, Sumin Park, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. Robot-r1: Reinforcement learning for enhanced embodied reasoning in robotics. *Neural Information Processing Systems*, 2025a. doi: 10.48550/arXiv.2506.00070.
- Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, A. Balakrishna, Suraj Nair, Rafael Rafailov, E. Foster, Grace Lam, Pannag R. Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning*, 2024. doi: 10.48550/arXiv.2406.09246.

- Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems XXI*. Robotics: Science and Systems Foundation, 2025b. doi: 10.15607/rss.2025.xxi.017.
- Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv.org*, 2025. doi: 10.48550/arXiv.2509.09674.
- Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onet-wovla: A unified vision-language-action model with adaptive reasoning. *arXiv.org*, 2025. doi: 10.48550/arXiv.2505.11917.
- Honglin Lin, Zheng Liu, Yun Zhu, Chonghan Qin, Juekai Lin, Xiaoran Shang, Conghui He, Wentao Zhang, and Lijun Wu. Mmfinereason: Closing the multimodal reasoning gap via open data-centric methods. *arXiv.org*, 2026. doi: 10.48550/arXiv.2601.21821.
- Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems 36*, pp. 44776–44791, 2023. doi: 10.52202/075280-1939.
- Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, 2025a. doi: 10.48550/arXiv.2410.07864.
- Yang Liu, Weixing Chen, Yongjie Bai, Jing-Hua Luo, Xinshuai Song, Kaixuan Jiang, Zhida Li, Ganlong Zhao, Junyi Lin, Guanbin Li, et al. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics*, 30(6): 7253–7274, 2024. doi: 10.1109/TMECH.2025.3574943.
- Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanguang Zhang, Yaochen Hu, Lingfeng Zhang, Yingxue Zhang, Shuang Wu, Tongtong Cao, Guowei Huang, et al. Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning. *arXiv.org*, 2025b. doi: 10.48550/arXiv.2501.10074.
- Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via closed-loop resampling. *arXiv.org*, 2025c. doi: 10.48550/arXiv.2408.17355.
- Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv.org*, 2025. doi: 10.48550/arXiv.2505.18719.
- Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *IEEE Transactions on Neural Networks and Learning Systems*, 37(7):3031–3051, 2024. doi: 10.1109/TNNLS.2025.3650584.
- Nvidia, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, LinxiJimFan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv.org*, 2025. doi: 10.48550/arXiv.2503.14734.
- Abigail O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, A. Padalkar, Abraham Lee, A. Pooley, Agrim Gupta, A. Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration. In *IEEE International Conference on Robotics and Automation*, pp. 6892–6903. IEEE, IEEE, 2024. doi: 10.1109/icra57147.2024.10611477.
- Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. In *Robotics: Science and Systems XXI*. Robotics: Science and Systems Foundation, 2025. doi: 10.15607/rss.2025.xxi.012.

- Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yani Ding, Zhigang Wang, Jiayuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *Robotics: Science and Systems XXI*, 2025. doi: 10.15607/rss.2025.xxi.011.
- Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. Vision-language-action models: Concepts, progress, applications and challenges. *arXiv.org*, 2025. doi: 10.48550/arXiv.2505.04769.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, R. Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv.org*, 2024. doi: 10.48550/arXiv.2402.03300.
- Wenxuan Song, Jiayi Chen, Pengxiang Ding, Han Zhao, Wei Zhao, Zhide Zhong, Zongyuan Ge, Jun Ma, and Haoang Li. Accelerating vision-language-action model integrated with action chunking via parallel decoding. *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 13162–13169, 2025. doi: 10.1109/IROS60139.2025.11247519.
- Zirui Song, Guangxian Ouyang, Mingzhe Li, Yuheng Ji, Chenxi Wang, Zixiang Xu, Zeyu Zhang, Xiaoqing Zhang, Qian Jiang, Zhenhao Chen, et al. Manipvm-r1: Reinforcement learning for reasoning in embodied manipulation with large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18558–18566. Association for the Advancement of Artificial Intelligence (AAAI), 2026. doi: 10.1609/aaai.v40i22.38922.
- Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-rft: Reinforcement fine-tuning for visual reasoning. *arXiv.org*, 2025a. doi: 10.48550/arXiv.2503.20752.
- Huajie Tan, Enshen Zhou, Zhiyu Li, Yijie Xu, Yuheng Ji, Xiansheng Chen, Cheng Chi, Pengwei Wang, Huizhu Jia, Yulong Ao, et al. Robobrain 2.5: Depth in sight, time in mind. *arXiv.org*, 2026. doi: 10.48550/arXiv.2601.14352.
- Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv.org*, 2025b. doi: 10.48550/arXiv.2505.17016.
- Gemini Robotics Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv (Cornell University)*, 2023. doi: 10.48550/arxiv.2312.11805.
- O. Team, Dibya Ghosh, H. Walke, Karl Pertsch, Kevin Black, Oier Mees, S. Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *Robotics: Science and Systems XX*, 2024. doi: 10.15607/rss.2024.xx.090.
- H. Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Maximilian Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Q. Vuong, Andre Wang He, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736. PMLR, 2023. doi: 10.48550/arXiv.2308.12952.
- Zhiqiang Wang, Hao Zheng, Yunshuang Nie, Wenjun Xu, Qingwei Wang, Hua Ye, Zhe Li, Kaidong Zhang, Xuewen Cheng, Wanxi Dong, et al. All robots in one: A new standard and unified dataset for versatile, general-purpose embodied agents. *arXiv.org*, 2024. doi: 10.48550/arXiv.2408.10899.
- Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics. Robotics: Science and Systems Foundation*, 2024. doi: 10.15607/RSS.2025.XXI.152.
- Yisheng Xiao, Lijun Wu, Junliang Guo, Juntao Li, Min Zhang, Tao Qin, and Tie-Yan Liu. A survey on non-autoregressive generation for neural machine translation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11407–11427, 2022. doi: 10.1109/TPAMI.2023.3277122.

- Zhiyuan Xu, Kun Wu, Junjie Wen, Jinming Li, Ning Liu, Zhengping Che, and Jian Tang. A survey on robotics with foundation models: toward embodied ai. *arXiv.org*, 2024. doi: 10.48550/arXiv.2402.02385.
- Jingyun Yang, Max Sobol Mark, Brandon Vu, Archit Sharma, Jeannette Bohg, and Chelsea Finn. Robot fine-tuning made easy: Pre-training rewards and policies for autonomous real-world reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pp. 4804–4811. IEEE, IEEE, 2023. doi: 10.1109/ICRA57147.2024.10610421.
- Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning*, 2024. doi: 10.48550/arXiv.2407.08693.
- Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, J. Jang, Yi Li, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. Grape: Generalizing robot policy via preference alignment. *arXiv.org*, 2024. doi: 10.48550/arXiv.2411.19309.
- Baining Zhao, Ziyu Wang, Jianjie Fang, Chen Gao, Fanhang Man, Jinqiang Cui, Xin Wang, Xinlei Chen, Yong Li, and Wenwu Zhu. Embodied-r: Collaborative framework for activating embodied spatial reasoning in foundation models via reinforcement learning. *ACM Multimedia*, pp. 11071–11080, 2025a. doi: 10.1145/3746027.3755703.
- Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Computer Vision and Pattern Recognition*, pp. 1702–1713. IEEE, 2025b. doi: 10.1109/CVPR52734.2025.00166.
- Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daum’e, A. Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2412.10345.
- Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhan Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv.org*, 2025. doi: 10.48550/arXiv.2507.01925.

## A Appendix

### A.1 CoT Data Construction Prompt

```

Role Assignment
You are an advanced robotic intelligence agent.
Inputs
• Global Task Instruction – overall goal.
• Keyframes – an ordered list of N external-camera images capturing critical moments.
– Frame 1 is before any subtask; Frame i (i > 1) is after subtask i-1 and before subtask i.
Required Output
For each keyframe (in order), output exactly:
Produce exactly N consecutive (<think> ... </think><subtask> ... </subtask>)
pairs—one pair per image, same order, plain text, no extra lines. (≤ 50
words, single paragraph, no line breaks)
<think> rules
1. Frame 1: declare initial frame.
2. Frame i (i > 1): internally compare the previous image with the current image
to infer the effect of the last subtask, but do not state Success or
Failure explicitly.
3. Tag every instruction-relevant object with its location in the current image,
e.g., bowl (right-front).
4. Describe spatial layout, affordances, obstacles, and reasoning that motivates
an immediate next subgoal advancing the Global Task.
5. Do not mention other frames, prior success, failure, or progress metrics. Just
analyze the current frame.
6. Do not mention frame indices like Frame 1; use initial frame.
<subtask> rules
• State the immediate next subgoal in clear natural language, without numerals
or explicit quantities.
• Each subtask must correspond to the change that should occur between the
current image and the next one (or to completion at the final image).
• If the Global Task is fulfilled in the current image, output exactly <subtask>
finish</subtask> and stop; omit remaining pairs.
Global Constraints
• Output pairs equal exactly the number of keyframes provided.
• No extra text outside the mandated tags.
• No bullet points or lists.
• No numbers inside any <subtask> tag.
• Reasoning and subtasks must align with the images, temporal flow, and the
global instruction.

```

Figure 4: **Prompt for keyframe CoT annotation via a cloud LVLM.** The prompt elicits per-frame spatial scene descriptions and temporal subtask decompositions.

### A.2 Implementation Details

DeepThinkVLA is initialized from the public  $\pi_0$ -FAST weights (Pertsch et al., 2025). We refactor the baseline policy with our hybrid-attention decoder (Sec. 3.2), yielding a 2.9B parameter model.

**Dataset Construction.** Before SFT, we construct embodied CoT datasets using the two-stage pipeline described in Sec. 3.3, based on public robot manipulation demonstrations. Specifically, starting from the LIBERO dataset, this process yields 273,465 CoT-annotated frames, which serve as the primary supervision source for the SFT cold-start stage. In addition, to support training and evaluation under high-fidelity dynamics and longer

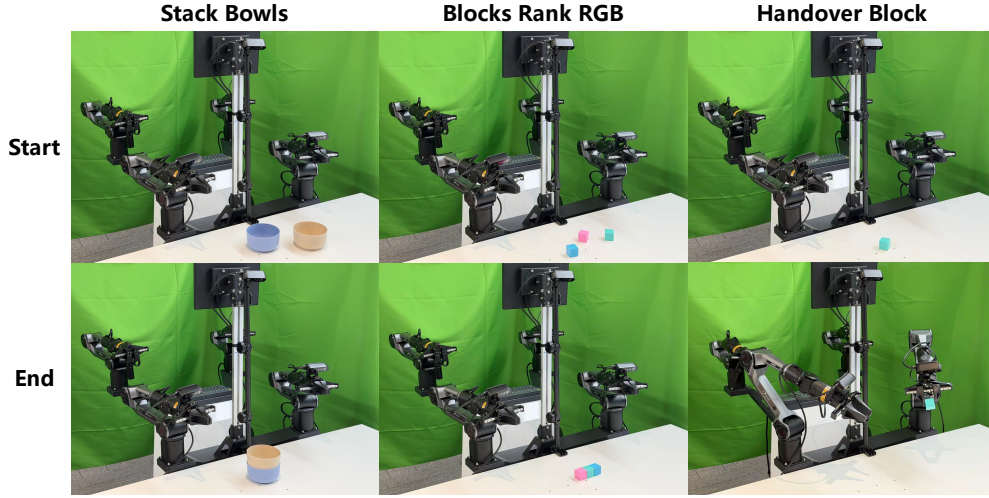


Figure 5: **Real-robot setup.** Start (top row) and end (bottom row) configurations of the three physical manipulation tasks—Stack Bowls, Blocks Rank RGB, and Handover Block—on the AGILEX ALOHA bimanual platform with external cameras.

task horizons, we apply the same pipeline to RoboTwin 2.0 and construct a large-scale embodied CoT dataset comprising 2,847,856 annotated frames. Together, these datasets provide grounded reasoning supervision across both standard simulation and high-fidelity digital twin environments.

**Supervised Fine-Tuning (SFT).** For the SFT stage, we train with a batch size of 128 and a learning rate of  $2.5 \times 10^{-5}$  for 150k steps. A hybrid attention mask is employed so that CoT tokens are supervised autoregressively while action tokens are supervised bidirectionally within the same forward pass. Optimization is performed using a standard token-level cross-entropy loss.

**Reinforcement Learning (RL).** For the RL stage, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024). The action chunk size is set to 10. Each trajectory receives a sparse task-success reward plus a small format-regularization reward to maintain CoT quality. Policy updates use a mini-batch size of 128, a low clip ratio  $\epsilon = 0.2$ , a high clip ratio  $\epsilon = 0.28$ , and a KL penalty to the SFT reference model to avoid catastrophic forgetting.

**Infrastructure and Inference.** Training is conducted on  $8 \times$  NVIDIA A800 GPUs. At inference time, we use greedy decoding for both reasoning and action tokens. For CoT ablation experiments, we evaluate three inference modes: Full CoT, Mask CoT, and Random CoT.

**Real-Robot Setup and Calibrated Comparison.** For the real-robot experiments (Section 4.3), we use an AGILEX ALOHA bimanual robot platform equipped with external cameras; Figure 5 shows the physical setup together with the start and end configurations of each task. A human teleoperator collects demonstration trajectories for each of the three tasks, which span a range of manipulation skills and difficulty levels: Stack Bowls Two requires stacking two bowls into a nested configuration with precise alignment; Handover Block transfers a block between grippers, testing bimanual coordination; Blocks Rank RGB arranges colored blocks (red, green, blue) in a specified order, requiring color recognition and sequential reasoning.

We apply the identical two-stage CoT annotation pipeline (Section 3.3) to the collected demonstrations: keyframes identified by gripper state changes are annotated using a cloud-based VLM, and a locally fine-tuned model propagates CoT to intermediate frames. The model is initialized from the same  $\pi_0$ -FAST weights and fine-tuned via SFT on the resulting

real-robot CoT dataset. Evaluation is conducted over 20 trials per task with randomized initial object configurations.

The physical study deliberately uses SFT only. Performing online RL on the physical robot would require many trial-and-error rollouts—including failed grasps and contact-rich mistakes—which is unsafe without dedicated online-RL safety infrastructure; claims about RL-based causal alignment are therefore grounded in the controlled simulation and RoboTwin 2.0 experiments, where on-policy rollouts can be collected safely and at scale. Moreover, the 45% average success should not be compared directly with the near-saturated LIBERO results; a more appropriate reference is the harder, contact-rich RoboTwin 2.0 benchmark (Table 2), where the simulation results on the three corresponding task categories are 62.0% (Stack Bowls Two), 43.0% (Handover Block), and 77.0% (Blocks Rank RGB). The physical results are lower—most notably on Blocks Rank RGB—but remain within the same order of difficulty rather than collapsing from LIBERO-level performance, showing that the CoT annotation pipeline and hybrid decoding architecture can be instantiated on hardware and achieve nontrivial closed-loop execution under real camera, calibration, contact, and actuation noise. We emphasize the scope of this evidence: it supports the preliminary physical applicability of the CoT data construction and hybrid action decoder; it does not validate the full SFT-then-RL pipeline in the physical domain.

### A.3 Hybrid Decoder: Token-Level Specification

This section specifies the hybrid decoder of Section 3.2 at the token-flow level, covering how action chunk slots are represented, how ground-truth leakage is prevented, and how the attention mask is implemented.

**Sequence Layout and Action Tokenization.** Each policy call follows the sequence layout: observation/instruction prompt, `<think>` CoT tokens `</think>`, `<action>`, an action chunk, `</action>`, and EOS. The `<think>...</think>` delimiters mark the natural-language reasoning segment. The `<action>` and `</action>` tokens are only delimiters for the action segment: the content inside is not natural language but a fixed block of scalar action-value positions. Each scalar action value is normalized to  $[-1, 1]$  and discretized by the ActionTokenizer into one of 2,048 action-token IDs reused from the tail of the base tokenizer vocabulary; these are reused vocabulary IDs, not newly added per-bin string tokens. With `n_bins=2048` and `fast_skip_tokens=128`, the action-token ID range is  $[\text{vocab\_size} - 128 - 2048, \text{vocab\_size} - 128 - 1]$ ; in our released base tokenizer with `vocab_size=257152`, this is  $[254976, 257023]$ . The final 128 base-vocabulary IDs are skipped, and `<action>/</action>` are separate delimiter special tokens. The mapping is reversed: discrete bin index  $k$  is encoded as token ID  $\text{vocab\_size} - 1 - 128 - k$ . The action block has  $H \times d_a$  positions: the position index specifies the action timestep and action dimension, while the predicted token ID specifies the discretized scalar value for that coordinate. For LIBERO,  $H = 10$  and  $d_a = 7$ , so each policy call contains 70 scalar action positions; RoboTwin uses the same construction with 14-D actions.

**Action Slots: Neither Mask Tokens Nor Learned Queries.** The action positions are not learned query tokens and are not semantic mask tokens. During SFT, the labels contain the ground-truth action-value tokens, but before the transformer forward pass, the input embeddings at action-token positions are zeroed by the action-position mask. During inference, we append dummy placeholder IDs only to allocate the same  $H \times d_a$  positions, and these placeholder embeddings are likewise zeroed. The slots therefore preserve positional structure but do not reveal ground-truth action embeddings.

**Hybrid Attention Mask and Parallel Decoding.** Prompt and visual tokens are visible as context; the `<think>...</think>` reasoning segment is generated with the standard causal language-model mask. From the `<action>` region onward, the action-slot rows copy a fully visible valid-sequence row, so every action slot can attend to the prompt, the generated CoT, and all other action slots. At inference time, the model first autoregressively generates the reasoning segment up to `</think>` and `<action>`; we then append the zero-input action slots, `</action>`, and EOS, run one transformer forward pass with the hybrid mask, restrict

```

# Notation: H = action-chunk length, d_a = action dimension,
# V = vocab_size (257152), S = fast_skip_tokens (128), B = n_bins (2048)
# Reserved action-token slice (reused tail of the base vocabulary):
# begin = V - S - B # 254976
# end = V - S - 1 # 257023
# encode(bin k) -> id = V - 1 - S - k ; decode(id) -> bin = V - 1 - S - id

# 1) Placeholder construction: allocate H x d_a action positions
seq = [prompt, <think>, cot, </think>, <action>, act_slots(H x d_a), </action>, EOS]
# SFT: act_slots hold ground-truth action-value ids (labels only)
# inference: act_slots hold dummy placeholder ids

# 2) Embedding zeroing: slots keep positional structure, leak no action values
emb = embed(seq)
emb[action_positions] = 0

# 3) Hybrid attention mask: action-row replacement
mask = causal_mask(len(seq)) # prompt + CoT remain causal
for row in action_positions:
    mask[row, :valid_len] = 1 # action rows: fully visible

# 4) Training: one forward pass, token-level cross-entropy
logits = transformer(emb, mask) # CoT causal, action slots bidirectional

# 5) Inference
cot = autoregressive_decode(prompt, until=</think><action>) # causal attention
seq = append(cot, zeroed_act_slots, </action>, EOS)
logits = transformer(embed_and_zero(seq), hybrid_mask) # one forward pass
action_logits = logits[action_positions, begin:end+1] # restrict LM head
actions = detokenize(argmax(action_logits)) # H x d_a values in parallel

```

Figure 6: **Pseudocode for the hybrid decoder.** Placeholder construction allocates  $H \times d_a$  zero-input action slots; embedding zeroing prevents ground-truth leakage during SFT; action-row mask replacement grants action slots bidirectional visibility over the prompt, generated CoT, and all other slots, enabling one-pass parallel action decoding.

the LM head to the reserved action-token range, and decode all  $H \times d_a$  action values in parallel. Figure 6 summarizes placeholder construction, embedding zeroing, and action-row mask replacement.

#### A.4 Baselines Details

We compare DeepThinkVLA with a comprehensive set of Vision–Language–Action (VLA) baselines used across our three benchmarks: LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and RoboTwin 2.0 (Chen et al., 2025b). The baselines span autoregressive policies, diffusion/flow-based policies, and parallel/block-decoding variants. Unless otherwise specified, results are reported following the official evaluation protocols of each benchmark.

**Autoregressive VLA baselines (LIBERO).** We include representative autoregressive (AR) VLAs that map vision and language to action tokens sequentially. TraceVLA (Zheng et al., 2024) augments AR decoding with spatial-temporal trace cues from past trajectories. Octo (Team et al., 2024) is a generalist AR policy trained across diverse manipulation data. OpenVLA (Kim et al., 2024) is a widely adopted open-source AR baseline for language-conditioned manipulation. SpatialVLA (Qu et al., 2025) incorporates explicit spatial modeling to improve AR policies. NORA (Hung et al., 2025) is an efficient AR VLA with strong performance on standard manipulation suites. UniVLA (Bu et al., 2025) trains a unified AR policy across multiple task suites.  $\pi_0$ -FAST (Pertsch et al., 2025) is a strong AR baseline based on the FAST action tokenizer. We also include AR policies further optimized beyond static imitation: GRAPE (Zhang et al., 2024) introduces RL-style optimization on top of a base policy. VLA-RL (Lu et al., 2025) applies outcome-driven RL to improve task success over an AR backbone.

Table 6: **Ablation on CoT and hybrid-decoding** (success rate % and relative inference time). Naively adding CoT to an AR decoder hurts performance; hybrid decoding restores and amplifies the benefit.

| Category   | Method                    | Object (%)  | Spatial (%) | Goal (%)    | Long (%)    | Average (%) | Rel. Inference Time |
|------------|---------------------------|-------------|-------------|-------------|-------------|-------------|---------------------|
| Baseline   | $\pi_0$ -FAST             | 96.8        | 96.4        | 88.6        | 60.2        | 85.5        | 1.0×                |
| AR-CoT     | $\pi_0$ -FAST (Full CoT)  | 95.8        | 93.8        | 74.6        | 61.0        | 81.3        | 4.0×                |
| Hybrid CoT | DeepThinkVLA (Mask CoT)   | 99.0        | 97.2        | 96.0        | 93.6        | 96.5        | 0.175×              |
|            | DeepThinkVLA (Random CoT) | 97.8        | 94.4        | 60.2        | 87.8        | 85.1        | 0.175×              |
|            | DeepThinkVLA (Full CoT)   | <b>99.0</b> | <b>97.2</b> | <b>96.8</b> | <b>94.2</b> | <b>96.8</b> | 1.4×                |

**Diffusion / flow-based policies (LIBERO and RoboTwin 2.0).** Diffusion-based policies generate continuous actions through iterative refinement. Diffusion Policy (Chi et al., 2023) is a standard diffusion baseline widely used in robot learning.  $\pi_0$  (Black et al., 2025) is a strong flow/diffusion-style VLA that serves as a competitive non-autoregressive baseline and is evaluated in our LIBERO, LIBERO-Plus, and RoboTwin 2.0 comparisons.

**Parallel / block-decoding policies (LIBERO).** Parallel or block-decoding methods use bidirectional attention to predict multiple action tokens jointly, reducing latency and mitigating autoregressive interference. CoT-VLA-7B (Zhao et al., 2025b) supports Chain-of-Thought supervision with block-parallel action decoding. OpenVLA-OFT (Kim et al., 2025b) is a bidirectional/parallel-decoding variant built on OpenVLA, designed to improve inference efficiency and stability.

**Baselines specific to LIBERO-Plus (robustness benchmark).** LIBERO-Plus (Fei et al., 2025) evaluates robustness under seven perturbation families (camera, robot initialization, language rewrites, lighting, background, sensor noise, and object layout). We report the official robustness results for the following baselines included in Table 3:  $\pi_0$  (Black et al., 2025),  $\pi_0$ -FAST (Pertsch et al., 2025), UniVLA (Bu et al., 2025), WorldVLA (Cen et al., 2025), OpenVLA (Kim et al., 2024), OpenVLA-OFT (Kim et al., 2025b), NORA-long (Hung et al., 2025), and RIPT-VLA (Tan et al., 2025b). For clarity, we follow the benchmark conventions and additionally include two commonly reported OpenVLA-OFT variants: OpenVLA-OFT\_w (augmented with wrist-camera observations) and OpenVLA-OFT\_m (trained with additional mixed data), as shown in Table 3.

**Baselines specific to RoboTwin 2.0 (high-fidelity digital twin).** RoboTwin 2.0 (Chen et al., 2025b) features longer horizons and contact-rich dynamics. We compare against widely used baselines reported in Table 2:  $\pi_0$  (Black et al., 2025),  $\pi_0$ -FAST (Pertsch et al., 2025), RDT (Liu et al., 2025a), and OpenVLA-OFT (Kim et al., 2025b). These cover both diffusion/flow-based policies and parallel-decoding approaches under the same benchmark protocol.

## A.5 Detailed Robustness Analysis

**Wrist Camera Ablation.** Incorporating a wrist camera provides additional improvements by capturing near-field contact information. However, even without the wrist camera, DeepThinkVLA achieves 86.0% on LIBERO, already outperforming  $\pi_0$ -FAST (85.5%). This confirms that the gains primarily stem from the hybrid architecture and reasoning capability rather than specific sensor modalities.

**Deconstructing LIBERO-Plus Robustness.** The LIBERO-Plus gains (Table 3) are broad-based rather than driven by a single factor. DeepThinkVLA improves camera robustness from 65.1% to 88.5%, robot-initial robustness from 21.6% to 40.5%, language robustness from 61.0% to 84.5%, and noise robustness from 74.4% to 94.4%. These results indicate that the hybrid decoding and embodied CoT reasoning help maintain functional competence under substantial perception/instruction-side distribution shifts.

Table 7: **LIBERO-Plus perception/instruction-side robustness: SFT-only vs. RL-aligned** (success rate %, zero-shot, no adaptation). RL alignment preserves broad robustness under perception/instruction-side shifts; the causal evidence for Condition 2 remains the action/dynamics-side Joint-Limit study in Table 8.

| Model                                     | Camera | Robot-Init | Language | Light | Background | Noise | Layout | Total |
|-------------------------------------------|--------|------------|----------|-------|------------|-------|--------|-------|
| DeepThinkVLA ( $\pi_0$ -FAST, SFT-Only)   | 88.5   | 40.5       | 84.5     | 90.0  | 75.3       | 94.4  | 79.9   | 79.0  |
| DeepThinkVLA ( $\pi_0$ -FAST, RL-Aligned) | 88.7   | 41.3       | 83.9     | 90.4  | 76.2       | 93.9  | 79.4   | 79.1  |
| $\Delta$ (RL – SFT)                       | +0.2   | +0.8       | −0.6     | +0.4  | +0.9       | −0.5  | −0.5   | +0.1  |

Table 8: **Controlled OOD analysis via Joint-Limit dynamics.** SFT-only drops as much as the reasoning-free baseline ( $\sim 32$  pp); RL alignment reduces the drop to 24.4 pp, and masking CoT widens it again, indicating that RL-aligned reasoning actively aids OOD adaptation.

| Method / Stage                      | Inference Mode | Dynamics  | Success Rate (%) |      |      |      |      | Drop (pp) |
|-------------------------------------|----------------|-----------|------------------|------|------|------|------|-----------|
|                                     |                |           | Obj              | Spa  | Goal | Long | Avg  |           |
| $\pi_0$ -FAST Pertsch et al. (2025) | -              | Standard  | 96.8             | 96.4 | 88.6 | 60.2 | 85.5 | -         |
|                                     |                | OOD Limit | 77.0             | 64.0 | 31.2 | 43.6 | 53.9 | 31.6      |
| DeepThinkVLA(SFT-Only)              | Full CoT       | Standard  | 99.0             | 97.2 | 96.8 | 94.2 | 96.8 | -         |
|                                     |                | OOD Limit | 87.0             | 61.4 | 54.2 | 56.4 | 64.8 | 32.0      |
|                                     | Mask CoT       | Standard  | 99.0             | 97.2 | 95.8 | 93.4 | 96.4 | -         |
|                                     |                | OOD Limit | 85.2             | 58.0 | 50.6 | 48.2 | 60.5 | 35.9      |
| DeepThinkVLA(RL-Aligned)            | Full CoT       | Standard  | 99.0             | 96.6 | 96.4 | 96.2 | 97.0 | -         |
|                                     |                | OOD Limit | 91.6             | 64.2 | 66.2 | 68.4 | 72.6 | 24.4      |
|                                     | Mask CoT       | Standard  | 99.0             | 97.2 | 96.0 | 93.6 | 96.5 | -         |
|                                     |                | OOD Limit | 88.2             | 60.2 | 61.0 | 65.6 | 68.8 | 27.7      |

**Perception-Side vs. Action-Side OOD: Effect of RL Alignment.** The seven LIBERO-Plus dimensions primarily perturb the *perception/instruction side* (camera view, robot initialization, language rewrites, lighting, background, sensor noise, and object layout): they mainly stress the visual-language encoder’s invariance and grounding robustness, since a correctly grounded subgoal remains essentially unchanged under, e.g., a camera or lighting change. The Joint-Limit intervention instead perturbs the *action/dynamics side*: the same subgoal plan must be realized by different action trajectories and recovery behavior, so the quality of the CoT-to-action link directly matters. This distinction is why Joint-Limit serves as the designed causal test for Condition 2, and it delimits the scope of our claim: Causal Alignment is required for CoT to remain functional in action-execution-sensitive OOD settings, not for every OOD factor.

To verify that RL alignment does not sacrifice perception-side robustness, we evaluate the SFT-only and RL-aligned models on every LIBERO-Plus perturbation dimension with no adaptation (Table 7); the DeepThinkVLA ( $\pi_0$ -FAST) row of Table 3 corresponds to the SFT-only model. RL alignment preserves the broad zero-shot robustness (79.1% vs. 79.0% overall), with per-dimension changes confined to a narrow band. We interpret this as a positive robustness check rather than causal evidence: aligning the CoT-to-action pathway neither requires nor damages the perception-side robustness measured by LIBERO-Plus.

The OOD Joint-Limit experiment (Table 8) provides the strongest evidence for Condition 2. Despite high standard performance, the SFT-only model mirrors the fragility of the reasoning-free baseline (32.0 pp vs. 31.6 pp drop)—SFT-learned CoT is “fake reasoning” that mimics annotation style without causal influence on actions. Only after RL does the drop shrink to 24.4 pp, and masking CoT widens it to 27.7 pp, indicating that RL alignment makes the CoT pathway action-relevant and turns CoT into a functional planning signal under dynamics shift.

## A.6 Ablations on Architecture and the Role of CoT

**Validating Condition 1: Decoding Alignment.** To directly test whether Decoding Alignment is necessary for CoT to be effective, we compare a naive autoregressive CoT baseline with our hybrid-decoding architecture. Specifically, we apply identical embodied CoT su-

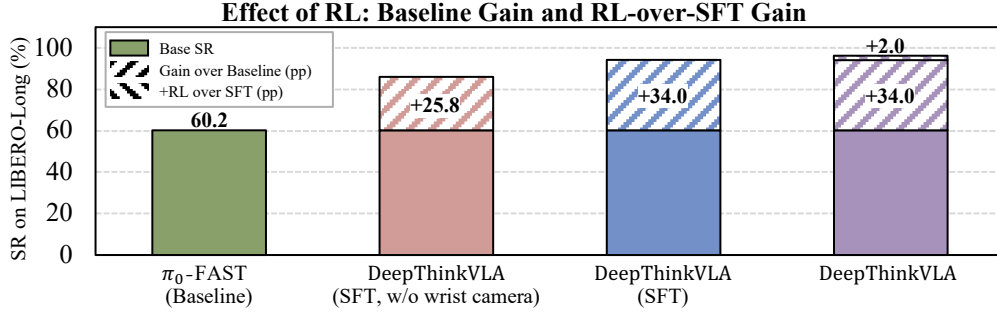


Figure 7: **Effect of RL on LIBERO-Long.** Solid bars share the  $\pi_0$ -FAST baseline (60.2%); shaded segments show gains from architecture and RL. The teal segment marks the +2 pp RL improvement over SFT-only.

pervision to the autoregressive  $\pi_0$ -FAST model and evaluate against DeepThinkVLA under the same training conditions.

As shown in Table 6, directly introducing CoT into a fully autoregressive decoder degrades performance (Average: 81.3% vs. 85.5% without CoT) and incurs a  $4\times$  increase in inference latency. This is the direct evidence for Condition 1: forcing a single autoregressive stream to jointly model sequential reasoning and high-dimensional actions creates destructive interference between modalities, making CoT actively harmful. In contrast, the hybrid-decoding architecture consistently achieves substantially higher success rates (Average: 96.8%) with only modest latency overhead, yielding a +15.5 percentage point improvement over the naive AR-CoT baseline. These results confirm that Decoding Alignment—matching the generation mechanism to each modality’s intrinsic structure—is a necessary condition for CoT to benefit VLA models.

**Semantic Coherence of CoT.** To examine whether CoT functions as an explicit reasoning signal rather than a mere auxiliary token stream, we further analyze its semantic role at inference time. We consider two controlled variants: Mask CoT, where reasoning tokens are replaced by placeholders, and Random CoT, where CoT tokens are substituted with randomly generated content.

As reported in Table 6, Mask CoT yields only a minor performance change under standard evaluation (Average: 96.5%), reflecting near-saturation in-distribution. In contrast, Random CoT results in a substantial degradation (Average: 85.1%), demonstrating that once reasoning tokens are consumed by the policy, their semantic coherence is critical for successful execution. The apparently conflicting pattern—Mask CoT  $\approx$  Full CoT yet Random CoT  $\ll$  Full CoT—is best explained by a *dual-channel* behavior: the policy conditions on a coherent CoT when it is present, but falls back to a strong visual-action prior when the CoT signal is absent. Consistent with this view, under out-of-distribution dynamics (Table 8) the visual-action prior becomes less sufficient—the same subgoal must be realized by a different execution trajectory—and the gap between Full CoT (72.6%) and Mask CoT (68.8%) widens accordingly. We therefore state the role of CoT precisely: aligned CoT is not uniformly important on every in-distribution step; it functions as a planning signal that is most measurably useful when the visual-action prior is insufficient, such as under dynamics shift, error recovery, and long-horizon execution.

**Same-Architecture Control: Removing CoT from Training.** The Mask/Random CoT interventions above operate at inference time with fixed weights. As a complementary training-time control, we retrain DeepThinkVLA *without* the CoT pathway while keeping the action decoder and the RL alignment procedure fixed: during SFT, CoT supervision is removed; during RL rollouts and inference, CoT generation is skipped. This control separates the contribution of the explicit CoT pathway from that of the hybrid action

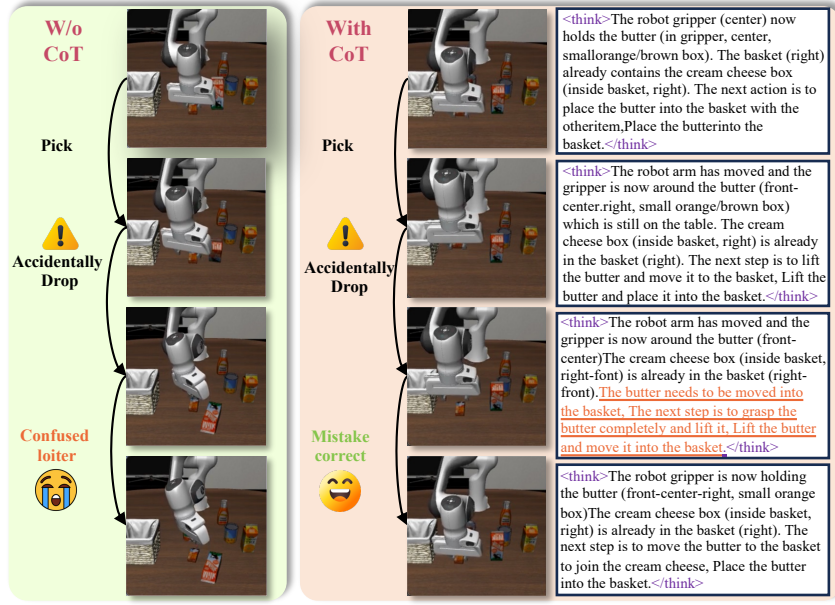


Figure 8: “Think before acting” enables error recovery. **Left:** the baseline enters a repetitive failure loop after a missed grasp. **Right:** DeepThinkVLA re-articulates the subgoal via CoT and recovers.

decoder and the RL training stage, and directly tests whether the final performance can be explained by the generic SFT-then-RL recipe alone.

As shown in Table 5 (in the main text), the w/o-CoT model reaches 92.7% after SFT and 93.8% after RL, confirming that the hybrid decoder and the RL pipeline are effective on their own. However, the full model with CoT under the identical RL procedure reaches 97.0%: a further +3.2 pp that cannot be attributed to the action decoder or to the generic SFT-then-RL recipe, with especially large gains on Goal (+4.6 pp) and Long (+6.0 pp). This control also fixes the data pipeline, task distribution, and CoT annotation quality; it therefore isolates what the explicit CoT pathway contributes at this data-quality level, while higher-quality CoT annotation remains an orthogonal axis that may further raise performance (Chen et al., 2025a; Lin et al., 2026). Together with the fixed-weight Mask/Random CoT interventions and the OOD analysis in Appendix A.5, these results provide converging functional evidence—from training-time removal, fixed-weight masking, random semantic corruption, and dynamics-shift robustness—that the CoT pathway is functionally useful for action generation. We note this is functional evidence, not a complete mechanistic proof of human-like reasoning.

### A.7 CoT Quality Audits

Because dataset scale alone does not guarantee supervision quality, we report two complementary text-level audits: a *demonstration-data audit* that checks the CoT annotations used as SFT supervision, and a *deployment audit* that checks whether the trained policy still generates structured CoTs at test time. Both audits are necessary-not-sufficient checks: they measure text form, not the functional use of CoT, which is established separately in Appendix A.6.

**Audit Protocol.** Each sample is first parsed for exactly one nonempty `<think>...</think>` block; malformed blocks are counted as malformed and receive 0 on the five pass checks. For valid blocks, the criteria are deterministic and intentionally broad: *object grounding* requires at least one task/allowed object mention; *spatial relation* checks spatial terms; *temporal subgoal* checks task/action verbs; *no unexpected object* rejects unexpected object phrases; and *not low-level transcript* requires object mentions, at least 12 alphabetic words, and a low-level action/coordinate-term ratio below 0.08. The deployment audit adds two deliberately

Table 9: **Demonstration-data CoT quality audit** (pass counts and rates over 1,000 sampled annotations). The supervision is object-grounded, spatially specific, and subgoal-oriented rather than a low-level action transcript; unexpected object mentions are conservatively counted as failures.

| Quality criterion                         | LIBERO demos (N=500) | RoboTwin demos (N=500) | Overall (N=1000)     |
|-------------------------------------------|----------------------|------------------------|----------------------|
| Object grounding                          | 500 / 500 (100.0%)   | 500 / 500 (100.0%)     | 1000 / 1000 (100.0%) |
| Spatial relation description              | 500 / 500 (100.0%)   | 500 / 500 (100.0%)     | 1000 / 1000 (100.0%) |
| Temporal subgoal description              | 499 / 500 (99.8%)    | 500 / 500 (100.0%)     | 999 / 1000 (99.9%)   |
| No hallucinated/unexpected object mention | 468 / 500 (93.6%)    | 500 / 500 (100.0%)     | 968 / 1000 (96.8%)   |
| Not low-level action transcript           | 500 / 500 (100.0%)   | 500 / 500 (100.0%)     | 1000 / 1000 (100.0%) |
| Malformed tag rate                        | 0 / 500 (0.0%)       | 0 / 500 (0.0%)         | 0 / 1000 (0.0%)      |

Table 10: **Deployment generated-CoT text audit** over 1,000 CoTs generated by the RL-aligned checkpoint. Upper rows: broad format/non-degeneration checks. Lower rows: deliberately stricter planning-marker checks on the same generated CoTs.

| Format / non-degeneration check                                         | Standard inference (N=500) | Joint-Limit inference (N=500) | Overall (N=1000)   |
|-------------------------------------------------------------------------|----------------------------|-------------------------------|--------------------|
| Contains task/scene object reference                                    | 500 / 500 (100.0%)         | 499 / 500 (99.8%)             | 999 / 1000 (99.9%) |
| Contains spatial relation term                                          | 500 / 500 (100.0%)         | 499 / 500 (99.8%)             | 999 / 1000 (99.9%) |
| Contains manipulation/subgoal verb                                      | 500 / 500 (100.0%)         | 499 / 500 (99.8%)             | 999 / 1000 (99.9%) |
| No unexpected object mention                                            | 498 / 500 (99.6%)          | 496 / 500 (99.2%)             | 994 / 1000 (99.4%) |
| Not low-level action transcript                                         | 500 / 500 (100.0%)         | 499 / 500 (99.8%)             | 999 / 1000 (99.9%) |
| Malformed tag rate                                                      | 0 / 500 (0.0%)             | 1 / 500 (0.2%)                | 1 / 1000 (0.1%)    |
| <i>Stricter planning-marker checks on the same 1,000 generated CoTs</i> |                            |                               |                    |
| Contains $\geq 2$ manipulation-stage groups                             | 200 / 500 (40.0%)          | 167 / 500 (33.4%)             | 367 / 1000 (36.7%) |
| Contains explicit goal-to-next-step phrase                              | 188 / 500 (37.6%)          | 194 / 500 (38.8%)             | 382 / 1000 (38.2%) |

stricter planning-marker checks on the same generated CoTs: the first passes when the text contains terms from at least two manipulation-stage groups (approach, grasp, transport, place, operate); the second passes when the text contains an explicit phrase linking the task goal or scene state to the next subgoal.

**Demonstration-Data Audit.** We sample 1,000 CoT annotations, balanced across 500 LIBERO demonstration CoTs (from a 40-task stratified audit pool) and 500 RoboTwin demonstration CoTs (from 24 random episodes). As shown in Table 9, the CoT supervision is generally object-grounded, spatially specific, and subgoal-oriented rather than an action transcript: 100.0% pass object grounding, 99.9% pass temporal subgoal description, 100.0% pass the not-low-level-transcript check, and no malformed tags occur. We conservatively count any unexpected object mention as a failure, which gives 96.8% overall on that criterion.

**Deployment Audit.** We further sample 1,000 generated CoTs from the official RL-aligned checkpoint during deployment inference: 500 from standard inference and 500 from Joint-Limit inference, covering 10 tasks per regime with 50 generated CoTs per task. As shown in Table 10, generated CoTs remain object-grounded, spatially specific, and subgoal-oriented in deployment, including under Joint-Limit inference: the broad non-degeneration checks pass at 99.4–99.9% overall, and malformed tags occur in 0.1% of samples. The stricter rows are intentionally discriminative and non-saturated: 36.7% of generated CoTs contain terms from at least two manipulation-stage groups, and 38.2% contain an explicit goal-to-next-step phrase.

We interpret these audits conservatively. The text evidence supports object-grounded next-subgoal planning with a nontrivial subset of explicit temporal/planning markers; it is not, by itself, proof of human-like reasoning or deep counterfactual abstraction. The current CoTs are mainly object-grounded next-subgoal statements: Figure 8 shows implicit condition-on-current-state re-planning after an execution error, but explicit branching or hypothetical simulation (e.g., “if the grasp fails, then re-grasp”) remains future work.

Table 11: **RL vs. SFT on RoboTwin 2.0.** RL yields an overall **+6.8%** gain (52.5%  $\rightarrow$  59.3%), confirming that outcome-based alignment is essential in contact-rich settings.

| Short Horizon Tasks (100–130 Steps)                             |                |                   |                   |                     |      |
|-----------------------------------------------------------------|----------------|-------------------|-------------------|---------------------|------|
| Model                                                           | Lift Pot       | Beat Hammer Block | Pick Dual Bottles | Place Phone Stand   | Avg  |
| DeepThinkVLA(SFT)                                               | 61.0           | 64.0              | 46.0              | 19.0                | 47.5 |
| DeepThinkVLA(RL)                                                | 62.0           | 73.0              | 61.0              | 24.0                | 55.0 |
| $\Delta$                                                        | +1.0           | +9.0              | +15.0             | +5.0                | +7.5 |
| Medium Horizon Tasks (150–230 Steps)                            |                |                   |                   |                     |      |
| Model                                                           | Move Can Pot   | Place A2B Left    | Place Empty Cup   | Handover Mic        | Avg  |
| DeepThinkVLA(SFT)                                               | 33.0           | 30.0              | 78.0              | 82.0                | 55.8 |
| DeepThinkVLA(RL)                                                | 52.0           | 38.0              | 83.0              | 88.0                | 65.3 |
| $\Delta$                                                        | +19.0          | +8.0              | +5.0              | +6.0                | +9.5 |
| Long (280–320 Steps) & Extra Long Horizon Tasks (450–650 Steps) |                |                   |                   |                     |      |
| Model                                                           | Handover Block | Stack Bowls Two   | Blocks Rank Rgb   | Put Bottles Dustbin | Avg  |
| DeepThinkVLA(SFT)                                               | 40.0           | 59.0              | 76.0              | 42.0                | 54.3 |
| DeepThinkVLA(RL)                                                | 43.0           | 62.0              | 77.0              | 49.0                | 57.8 |
| $\Delta$                                                        | +3.0           | +3.0              | +1.0              | +7.0                | +3.5 |
| Overall Gap ( $\Delta$ = RL - SFT): <b>+6.8</b>                 |                |                   |                   |                     |      |

## A.8 Qualitative Case Study

To qualitatively assess how explicit reasoning influences execution behavior, we compare rollout trajectories of DeepThinkVLA and the  $\pi_0$ -FAST baseline on representative LIBERO tasks (Figure 8). The baseline policy initially approaches the target object but fails to establish a stable grasp. Following this error, it repeatedly executes similar motions without corrective adjustment, eventually leading to task failure. This behavior reflects a key limitation of direct perception-to-action policies: once an error occurs, the policy lacks an internal mechanism to reassess progress or revise its plan.

In contrast, DeepThinkVLA generates a concise `<think>...</think>` trace prior to each action chunk. When an accidental drop occurs, the model explicitly re-articulates the subgoal (e.g., “the butter needs to be moved into the basket”), enabling corrective re-grasping and successful task completion. This qualitative difference illustrates that CoT in DeepThinkVLA is not merely descriptive, but functions as an actionable planning scaffold that supports recovery and re-planning. As a result, DeepThinkVLA exhibits more stable and resilient behavior under execution errors and perturbations.

## A.9 Detailed Ablation: SFT vs. RL on RoboTwin 2.0

To further substantiate the role of outcome-based reinforcement learning in complex, long-horizon settings, we provide a detailed comparison between SFT-only and RL-aligned models on the RoboTwin 2.0 benchmark. Table 11 reports per-task success rates across short, medium, and long (including extra-long) horizons.

The results show that RL yields consistent performance improvements across all horizon categories. Notably, the gains are most pronounced in medium (+9.5%) and short (+7.5%) horizon tasks, while still providing measurable benefits in long-horizon scenarios. These findings confirm that outcome-based RL plays a crucial role in refining reasoning–action alignment beyond what can be achieved through supervised imitation alone, particularly in high-fidelity, contact-rich environments.

## A.10 Future Direction: Spatial Reasoning as CoT Augmentation

The CoT in DeepThinkVLA takes the form of natural-language next-subgoal statements. A natural question for future work is whether *explicitly spatial* reasoning—e.g., predicting the end-effector trajectory that realizes the next subgoal—can serve as a complementary, more grounded intermediate representation along the CoT-to-action pathway.

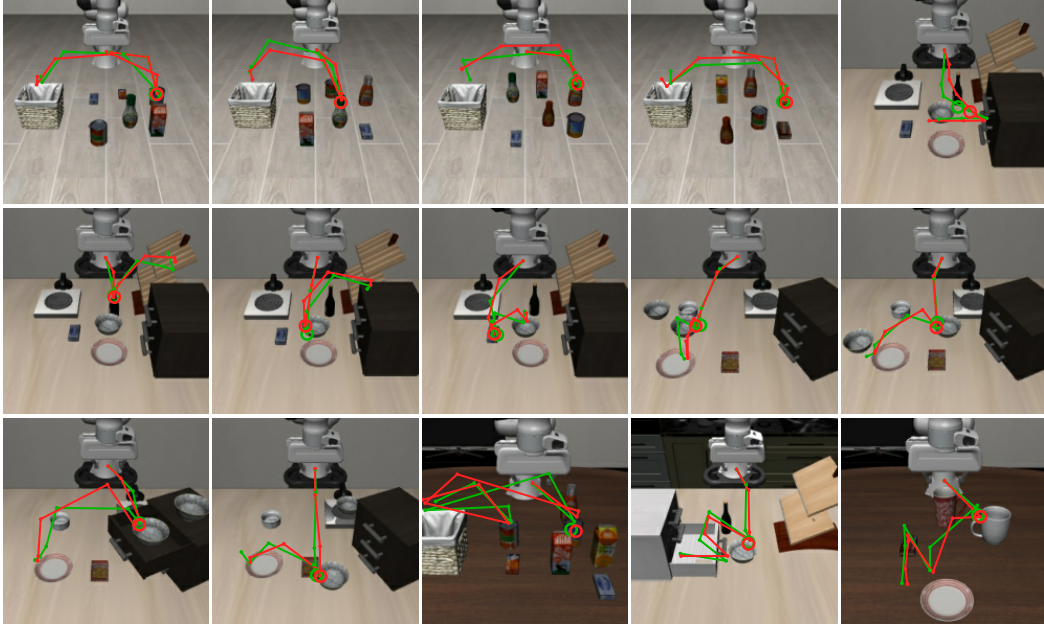


Figure 9: **Preliminary study: spatial-plan prediction with a LoRA-tuned RoboBrain 2.5 on LIBERO scenes.** Each panel overlays the ground truth (green) and the prediction of the LoRA-tuned RoboBrain 2.5 (red): lines trace the gripper waypoint path (end-effector positions sampled at action-chunk boundaries, projected into the image), and circles mark the grasp point (projected end-effector position at the first gripper closure). Closer red-to-green agreement indicates a more accurate spatial plan. The predictions are object-directed and broadly consistent with the ground truth, suggesting that spatial reasoning is a promising complement to language CoT.

As a preliminary exploration of this direction, we conduct an experiment based on RoboBrain 2.5 (Tan et al., 2026), an embodied vision-language foundation model with strong spatial reasoning and trajectory-generation capabilities. We fine-tune RoboBrain 2.5 with LoRA on spatial plans derived from our LIBERO demonstrations. For each episode, the ground-truth plan consists of (i) the *gripper waypoint path*: the end-effector positions sampled at action-chunk boundaries (every 10 steps, yielding 8 waypoints), projected into the image and connected as a polyline; and (ii) the *grasp point*: the projected end-effector position at the first gripper closure. Given the task instruction and the scene observation, the LoRA-tuned model predicts the same representation—a waypoint path and a grasp point in image space. Figure 9 presents qualitative results on 15 LIBERO scenes, overlaying the ground-truth plans (green) with the model predictions (red); the more closely the red path and circle track their green counterparts, the more accurate the predicted spatial plan.

The predicted waypoint paths are generally object-directed and broadly consistent with the ground-truth paths across scenes and viewpoints, and the predicted grasp points fall on or near the target objects. This suggests a concrete future direction: augmenting the language CoT with a spatial reasoning channel—e.g., conditioning the parallel action decoder on both the generated subgoal text and a predicted spatial plan (waypoint path and grasp point)—so that the reasoning segment conveys not only *what* to do next but also *where* and *how* to move. Integrating such spatially grounded CoT into the SFT-then-RL pipeline, and testing whether it further strengthens Causal Alignment under action-execution-sensitive OOD, is left to future work.

We emphasize the scope of this study: it is an exploratory, qualitative experiment built on a separate model (RoboBrain 2.5 with LoRA adaptation) rather than on the DeepThinkVLA policy evaluated in the main paper, and none of the claims, conditions, or quantitative results in the main paper depend on it.