LaST-R1: Reinforcing Robotic Manipulation via Adaptive Physical Latent Reasoning

Hao Chen[*1,2] **Zhonghao Yan**[*2] **Jiaming Liu**[*,†2] **Nuowei Han**[*2] **Renrui Zhang**[†1]
Chenyang Gu[2] **Jialin Gao**[1] **Peng Jia**[3] **Shanghang Zhang**[2 ✉] **Pheng-Ann Heng**[1]

[1]The Chinese University of Hong Kong [2]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University [3]Simplexity Robotics

Project page: `https://siriyep.github.io/last-r1/`

Abstract

Robotic foundation models require reasoning over complex visual scenes to execute adaptive actions in dynamic environments. While recent studies on latent-reasoning Vision-Language-Action (VLA) models have demonstrated the capability to capture fine-grained physical dynamics, they remain predominantly confined to static imitation learning, severely limiting their adaptability and generalization. In this paper, we present **LaST-R1**, a novel reinforcement learning (RL) post-training framework designed to effectively harness "latent reasoning-before-acting" policies. Specifically, we propose **Latent-to-Action Policy Optimization (LAPO)**, a core RL algorithm that jointly optimizes the latent reasoning process and the action generation. By explicitly embedding latent Chain-of-Thought (CoT) reasoning directly within the RL optimization loop, LAPO stimulates profound physical world modeling, which in turn drives robust execution in interactive environments. Furthermore, an **adaptive latent CoT mechanism** is introduced, allowing the policy to dynamically modulate its reasoning horizon based on diverse environment states. Experiments show that LaST-R1 achieves a near-perfect 99.9% average success rate on the LIBERO benchmark with only one-shot warm-up, significantly improving convergence speed and performance over prior state-of-the-art (SOTA) methods. In real-world deployments, LaST-R1 achieves an average improvement of 22.5 percentage points over SOTA supervised fine-tuning approach across four complex tasks, covering both single-arm and dual-arm settings. Finally, LaST-R1 demonstrates strong generalization across simulated and real-world environments.

1 Introduction

Driven by large-scale pre-trained VLMs [1, 2, 3], Vision-Language-Action (VLA) models [4, 5, 6, 7, 8, 9, 10] have emerged as a promising paradigm for general robotic manipulation. Moving beyond direct observation-to-action mapping, recent architectures increasingly draw inspiration from Chain-of-Thought (CoT) reasoning [11]. While explicitly generating linguistic traces [12, 13, 14, 15] or future states [16, 17, 18, 19, 20] provides structured guidance before acting, it incurs non-negligible inference latency and discretization bottlenecks. This inherently restricts the model's ability to capture continuous, high-frequency physical dynamics. Therefore, recent research [21, 22] has pivoted toward reasoning in a compact latent space that accommodates fine-grained, hard-to-verbalize physical knowledge, establishing a highly expressive reasoning foundation for complex manipulation.

*Equal contribution, †Project lead, ✉Corresponding author.

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

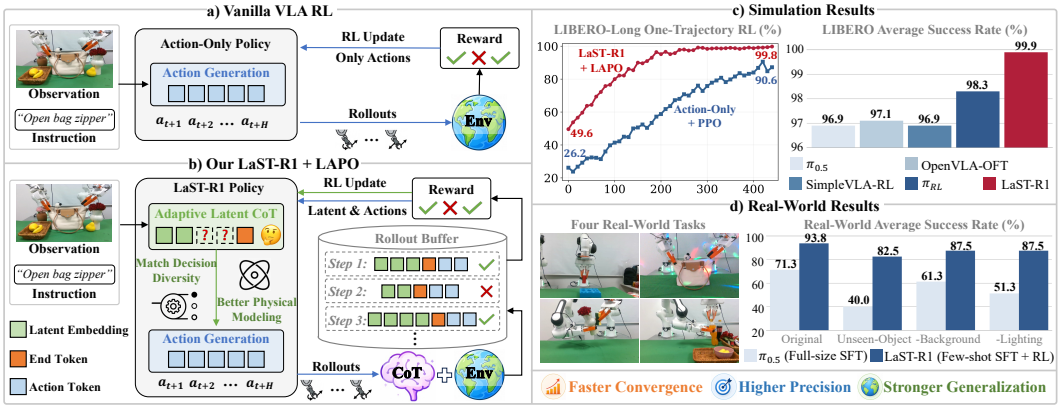


Figure 1: **The Overview of LaST-R1.** (a) Unlike vanilla RL baselines that strictly optimize actions, (b) our approach utilizes LAPO to jointly optimize an adaptive latent CoT alongside physical execution. By bridging cognitive reasoning and control, LaST-R1 achieves (c) faster convergence speed, higher success rate in simulation, and (d) stronger generalization capabilities in real-world.

Despite the advantages of latent reasoning, existing frameworks remain grounded in imitation learning, requiring massive, high-cost expert demonstrations. Their reliance on static datasets prevents closed-loop environmental interaction, leading to compounding errors and limited generalization. Concurrently, a nascent line of research [23, 24, 25, 26, 27, 28, 29, 30] has introduced online reinforcement learning (RL) for VLA post-training, improving exploration and robustness through trial-and-error interaction with the environment. However, as shown in Figure 1 (a), current methods are largely restricted to vanilla architectures that operate directly in the action space, bypassing the underlying physical reasoning process. Consequently, they lack interaction-driven learning to accurately model physical dynamics, limiting their ability to adapt to complex environments.

To this end, we propose **LaST-R1**, a novel RL post-training framework specifically developed to optimize "latent reasoning-before-acting" policies, as shown in Figure 1 (b). We first present the LaST-R1 VLA model, which tightly integrates latent CoT reasoning with action generation. Specifically, the model autoregressively produces latent reasoning embeddings to capture structured physical dynamics, serving as conditions for parallel action decoding. To ensure stable and expressive reasoning, these latent embeddings are grounded in global feature representations from a vision foundation model [31], providing a robust semantic and spatial prior. Building on this architecture, we propose **Latent-to-Action Policy Optimization (LAPO)**, a RL algorithm that jointly optimizes the latent reasoning process and action generation. Unlike prior RL approaches that operate solely within the action space [24, 23, 25], LAPO treats latent reasoning embeddings as implicit decision variables. By bridging this implicit deliberation with action generation through a unified step-level likelihood ratio, our method allows environmental rewards to directly shape the latent reasoning space, thereby concurrently optimizing the model's physical comprehension and interactive robustness. Finally, an **adaptive latent CoT mechanism** is introduced to augment the framework, enabling the dynamic adjustment of the reasoning horizon based on task diversity. This allows the policy to allocate more reasoning steps for intricate manipulations, while maintaining fast execution for reactive responses, effectively balancing cognitive capacity and computational efficiency.

To empirically validate our framework, LaST-R1 VLA is first initialized with pre-trained Qwen3-VL-4B [2] and undergoes large-scale pre-training across diverse robotic manipulation datasets [32, 33, 34]. For specific downstream tasks, the policy is adapted through a supervised fine-tuning (SFT) warm-up before engaging in online RL post-training. During evaluation, our approach achieves a near-perfect 99.9% average success rate on the LIBERO benchmark with one-shot warmup, outperforming prior state-of-the-art (SOTA) baselines, with a particularly notable improvement of 5.73 percentage points on the LIBERO-Long suite (long-horizon tasks). In real-world deployments, LaST-R1 achieves an average improvement of 22.5 percentage points over SOTA SFT method across four complex tasks, achieving a 93.75% average success rate across the four real-world tasks. Furthermore, systematic evaluations demonstrate that the proposed framework exhibits strong generalization in simulation environments, consistently outperforming vanilla action-only RL policies trained with standard PPO [35]. Notably, LaST-R1 achieves zero-shot generalization to unseen objects, backgrounds, and lighting conditions after LAPO post-training. In summary, our main contributions are as follows:

1) We present **LaST-R1**, a specialized RL post-training framework for "latent reasoning-before-acting" policies to enhance physical world modeling and robust action generation.
2) We propose **Latent-to-Action Policy Optimization (LAPO)**, a novel RL algorithm that jointly optimizes latent reasoning and action generation via a unified step-level likelihood ratio.
3) We design an **adaptive latent CoT mechanism** that dynamically adjusts the reasoning horizon based on task diversity, balancing reasoning capacity and computational efficiency.

2 Related Work

Vision-Language-Action (VLA) Models. Based on pretrained vision-language models (VLMs), VLA models [8, 6, 10, 36, 37] have demonstrated promising performance in robotic control with discrete action generation [4, 38] or continuous action representation. Recent frameworks have diversified into regression-based [5], diffusion-based [39, 9, 17, 36], and flow-matching-based architectures [40, 7, 41]. To further enhance precision and spatial awareness, several studies incorporate 3D spatial information or point-cloud reasoning into the VLA backbone [42, 43], significantly boosting performance in complex manipulation tasks. Inspired by Chain-of-Thought (CoT) reasoning in VLMs [11], recent research focuses on endowing VLAs with powerful reasoning capabilities. This includes generating intermediate text [7, 14], visual [22, 16], or multimodal plans [17], notably through methods like Embedded CoT [15] to strengthen spatial reasoning. However, these explicit decoding processes inevitably introduce inference latency. To alleviate this, further research explores introducing latent reasoning [44, 45] into the action generation process [46, 21, 22], balancing performance gains with execution efficiency. While these VLA frameworks have evolved significantly, the field remains predominantly confined to the imitation learning paradigm, where the lack of environmental interaction hinders the models' ability to generalize beyond static datasets or further refine their internal reasoning for robust control.

Reinforcement Learning (RL) for VLA Models. To overcome the limitations of static imitation learning, Reinforcement Learning (RL) has been increasingly adopted to fine-tune VLA models through environmental feedback [47]. Early efforts focus on offline preference alignment [48, 49], such as GRAPE [49], which employs Direct Preference Optimization (DPO) [50] to align robot trajectories with human intent. To further enhance closed-loop robustness, online RL frameworks have emerged [28, 24, 23, 25, 27, 51, 26, 52]. For example, VLA-RL [25] and RL4VLA [28] utilize Proximal Policy Optimization (PPO) [35] to improve generalization in unseen environments, while SimpleVLA-RL [24] and TGRPO [53] apply Group Relative Policy Optimization (GRPO) [54] to achieve efficient post-training without complex reward modeling. Depending on the action space, these methods optimize either discrete token probabilities [24, 25] or continuous action sequences [23]. However, a critical gap remains: current RL-based VLA methods predominantly support vanilla architectures, focusing exclusively on action-space supervision. They fail to address how to conduct RL post-training for reasoning-based VLAs, leaving the intrinsic link between the internal "thought" process and the final "act" unoptimized. Therefore, we present LaST-R1, a RL post-training framework with a unified VLA model that sequentially performs latent reasoning and action generation. Building on this, we propose Latent-to-Action Policy Optimization (LAPO) to concurrently optimize both the reasoning trajectory and the action generation, achieving superior robustness and precision in complex manipulation tasks.

3 Methodology

This section presents the proposed **LaST-R1** framework, as shown in Figure 2. We first review the problem formulation (Section 3.1), then introduce a unified VLA architecture with latent Chain-of-Thought (CoT) reasoning (Section 3.2). Next, we present **Latent-to-Action Policy Optimization (LAPO)**, which jointly optimizes latent reasoning and action generation (Section 3.3). Finally, we describe an **adaptive mechanism** that dynamically adjusts the reasoning horizon (Section 3.4).

3.1 Preliminaries

We formulate the Vision-Language-Action (VLA) model as a parameterized policy π_θ that maps multimodal observations (i.e., visual inputs and language instructions) to an action chunk $\mathbf{a}_{t:t+H}$ in

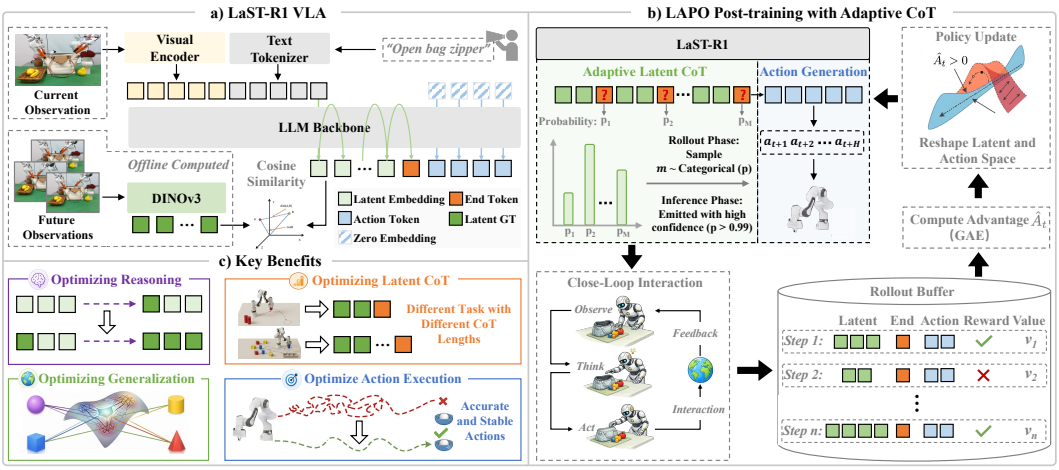


Figure 2: **The Framework of LaST-R1.** (a) LaST-R1 VLA is a unified model that takes visual observations and language instructions as input, where a vision foundation model provides physically grounded latent targets to guide latent CoT reasoning before action generation. (b) During LAPO RL post-training, LaST-R1 interacts with the environment in a closed loop manner, storing latents, actions, and rewards in a rollout buffer for jointly reshaping the latent and action spaces. It further enables adaptive reasoning by learning to emit the `<latent_end>` token based on predicted probabilities, dynamically adjusting the reasoning horizon across tasks. (c) Through LAPO, LaST-R1 achieves adaptive reasoning lengths across diverse tasks, improving generalization and execution stability.

$SE(3)$ space. For single-arm robots such as the Franka Research 3, the action is a 7-DoF end-effector control vector $\mathbf{a}_t \in \mathbb{R}^7$, comprising 3-DoF positional offsets, a 3-DoF orientation (represented as Euler angles), and a 1-DoF gripper state. For dual-arm systems, we extend this formulation to 14-DoF via vector concatenation. The policy can be optimized either through supervised fine-tuning (SFT) using expert demonstrations or via reinforcement learning (RL) via environment interaction.

SFT Formulation. In supervised fine-tuning, we have access to a dataset of expert demonstrations $\mathcal{D} = \{(\mathbf{s}_t, \mathbf{a}_{t:t+H})\}$, where $\mathbf{s}_t$ denotes the multimodal observations at timestep t. The policy is trained to imitate expert behavior by maximizing the conditional log-likelihood:

$$\mathcal{J}_{\text{SFT}}(\theta) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_{t:t+H}) \sim \mathcal{D}} \left[\log \pi_\theta(\mathbf{a}_{t:t+H} \mid \mathbf{s}_t) \right]. \tag{1}$$

While this objective enables effective behavioral cloning, the model's capability and generalization remain inherently bounded by the quality, scale, and diversity of the training data.

RL Formulation. To surpass the limitations of static imitation, RL casts the sequential decision-making problem as an interactive process where the policy engages with the environment over trajectories of length $T+1$. At each timestep t, the agent observes a state $\mathbf{s}_t$ and samples an action chunk $\mathbf{a}_{t:t+H} \sim \pi_\theta(\cdot \mid \mathbf{s}_t)$. The environment then transitions to the next state $\mathbf{s}_{t+1}$ and yields a scalar reward r_t. This interaction induces a trajectory τ, whose distribution is governed by the initial state and the environment dynamics. The objective is to maximize the expected discounted return:

$$\mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \gamma^t r_t \right], \quad \nabla_\theta \mathcal{J}_{\text{RL}}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_{t:t+H} \mid \mathbf{s}_t) \, \hat{A}_t \right], \tag{2}$$

where $\gamma \in [0, 1)$ is the discount factor, and $\hat{A}_t$ denotes the advantage estimate, capturing the relative utility of the sampled action chunk $\mathbf{a}_{t:t+H}$ compared to the expected value at state $\mathbf{s}_t$.

3.2 LaST-R1 Model Architecture

As shown in Figure 2 (a), we propose a unified LaST-R1 VLA model that jointly models latent reasoning and action generation. We first present the overall pipeline, followed by a novel method for constructing compact latent representations of future dynamics.

Overall Pipeline. LaST-R1 VLA builds upon the Qwen3-VL-4B [2] architecture, integrating a SigLIP2-Large [55] visual encoder with an LLM backbone. Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, the

visual encoder first employs 2D-RoPE with interpolated absolute positional embeddings to transform spatial information into dense visual embeddings $f_{\text{v}} \in \mathbb{R}^{N_{\text{v}} \times 2560}$. Subsequently, these embeddings are concatenated with language embeddings $f_{\text{l}} \in \mathbb{R}^{N_{\text{l}} \times 2560}$ and processed by the LLM backbone, where N_{v} and N_{l} denote the respective sequence lengths. The model first generates latent reasoning embeddings, followed by action prediction. To fully leverage the strong reasoning capability of the LLM, latent embeddings are generated in an autoregressive manner, with their representation detailed in the next paragraph. To support action generation, we normalize and discretize continuous actions into tokens, extending the LLM vocabulary accordingly [4]. We adopt a parameter-free action tokenizer to map between continuous actions and discrete tokens, while employing parallel decoding to improve inference efficiency [5]. To enable RL post-training, we introduce a value head composed of a 4-layer MLP to estimate state values, which shares the same backbone with the VLA [28, 23].

Latent Representation. To model temporal environmental dynamics, LaST-R1 VLA autoregressively infers future states over a horizon N_z, yielding N_z latent reasoning embeddings. Prior methods extract these latents via average pooling [21] or auxiliary learnable parameters [22], which often sacrifices fine-grained spatial structures or injects undesired inductive biases into the learned latent space. To overcome this, we construct latent embeddings using DINOv3 [31], a state-of-the-art vision foundation model. Specifically, we extract its `<CLS>` token ($f_{\text{d}} \in \mathbb{R}^{1 \times 4096}$) as a holistic image embedding and apply top-k (k = 2560) selection based on feature magnitude along the channel dimension. These selected latent targets align with the hidden size of the VLA embedding while preserving the most salient visual components. By harnessing DINOv3's structurally rich and semantically dense feature space, the extracted `<CLS>` token serves as a strong semantic prior, facilitating accurate modeling of physical dynamics. Crucially, these latent targets are precomputed offline, effectively decoupling the foundation model from the policy to ensure no additional overhead during both training and inference. We systematically compare this approach with prior latent representation formats in Section 4.2 to further validate its effectiveness.

3.3 Latent-to-Action Policy Optimization

To effectively train the proposed *latent reasoning-before-acting* policy LaST-R1 VLA, we introduce **Latent-to-Action Policy Optimization (LAPO)**, a RL framework that jointly optimizes latent reasoning and action generation, as shown in Figure 2 (b). In our formulation, the policy first generates latent embeddings, which condition subsequent action prediction. While prior works [28, 52] have established PPO [35] as an effective post-training method for VLA models, they operate solely in the action space and overlook intermediate reasoning processes that are integral to decision-making. To address this limitation, we propose LAPO, which treats latent embeddings as *implicit decision variables* and introduces a unified optimization framework that enables environment reward signals to shape both reasoning and action generation spaces across the entire trajectory.

Rollout Collection. At each environment step t, given the current state $\mathbf{s}_t$ comprising visual observations and language instructions, the policy first deterministically generates a sequence of latent embeddings $\mathbf{Z}_t^{\text{old}} = \{\mathbf{z}_{t,i}^{\text{old}}\}_{i=1}^{N_z}$ autoregressively: $\mathbf{z}_{t,i}^{\text{old}} \sim \pi_\theta(\cdot \mid \mathbf{s}_t, \mathbf{z}_{t,<i}^{\text{old}})$. Conditioned on these latents, a sequence of discrete action tokens $\mathbf{C}_t = \{\mathbf{a}_{t,j}\}_{j=1}^{N_a}$ is then produced, where N_a denotes the total number of tokens representing the H-step action chunk. Specifically, we reuse the KV cache from the latent generation phase and decode the action tokens in parallel via a single forward pass with bidirectional attention over N_a placeholder vectors. We further insert a special `<latent_end>` token between the latent and action sequences to signify the completion of reasoning. The final hidden embedding of this token is routed into a value head to output the state value estimation v_t. To formulate a unified action distribution for decision step t, we compute the joint log-probability of the action sequence by summing over the individual tokens: $\log \pi_\theta(\mathbf{C}_t \mid \cdot) = \sum_{j=1}^{N_a} \log \pi_\theta(\mathbf{a}_{t,j} \mid \cdot)$. During rollout, we store the sampled latent sequences $\mathbf{Z}_t^{\text{old}}$, action token sequences $\mathbf{C}_t$, their corresponding log-probabilities, and the estimated state values. Ultimately, a scalar reward is assigned based on task success or failure, yielding complete trajectories that are subsequently used to compute the advantage estimates $\hat{A}_t$ via Generalized Advantage Estimation (GAE) for policy optimization.

Policy Optimization. During policy updates, the model processes the identical rollout context to yield updated continuous latent embeddings $\mathbf{Z}_t^\theta = \{\mathbf{z}_{t,i}^\theta\}_{i=1}^{N_z}$ and updated discrete action log-probabilities. Because the advantage $\hat{A}_t$ is computed at the decision-step level, we define a step-level ratio $r_t(\theta)$ for both the action sequence and the continuous latents. For the discrete action sequence,

the ratio is computed via the joint probabilities: $r_t^a(\theta) = \exp\left(\log \pi_\theta(\mathbf{C}_t \mid \cdot) - \log \pi_{\theta_{\text{old}}}(\mathbf{C}_t \mid \cdot)\right)$. For the continuous latent embeddings, we approximate the distribution of rollout latents $\mathbf{Z}_t^{\text{old}}$ using an isotropic Gaussian centered at the current policy output $\mathbf{Z}_t^\theta$. By summing the squared Euclidean distances over the entire latent sequence length N_z, we define the step-level latent ratio surrogate as:

$$r_t^z(\theta) = \frac{\pi_\theta(\mathbf{Z}_t^{\text{old}} \mid \cdot)}{\pi_{\theta_{\text{old}}}(\mathbf{Z}_t^{\text{old}} \mid \cdot)} = \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N_z} \|\mathbf{z}_{t,i}^{\text{old}} - \mathbf{z}_{t,i}^\theta\|^2\right), \tag{3}$$

where σ is a fixed hyperparameter regulating the latent variance. For a more intuitive, step-by-step derivation of $r_t^z(\theta)$, please refer to Appendix A. By jointly optimizing reasoning latents and actions, we formulate the joint LAPO clipped surrogate loss as an expectation over all valid decision steps t:

$$\mathcal{L}_{\text{policy}}(\theta) = -\mathbb{E}_t\left[\sum_{k\in\{z,a\}} \min\left(r_t^k(\theta)\hat{A}_t, \text{clip}(r_t^k(\theta), 1-\epsilon_{\text{min}}, 1+\epsilon_{\text{max}})\hat{A}_t\right)\right], \tag{4}$$

where ϵ_{min} and ϵ_{max} are the clipping thresholds. This objective enables the environment reward to shape both the action space and the internal reasoning space simultaneously, as shown in Figure 2 (b). Specifically, when $\hat{A}_t > 0$, the optimization explicitly minimizes the latent distance $\sum_{i=1}^{N_z} \|\mathbf{z}_{t,i}^{\text{old}} - \mathbf{z}_{t,i}^\theta\|^2$, effectively pulling the current policy's latent representations toward the "good-reasoning" manifolds that facilitated successful trajectories. In practice, to achieve better optimization flexibility, we decouple the joint policy objective $\mathcal{L}_{\text{policy}}(\theta)$ into two separate components: the action loss $\mathcal{L}_{\text{action}}(\theta)$ (for $k = a$) and the latent loss $\mathcal{L}_{\text{latent}}(\theta)$ (for $k = z$). Finally, the total training loss unifies these policy objectives with state-value estimation, governed by weighting coefficients λ_1 and λ_2:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{action}}(\theta) + \lambda_1\mathcal{L}_{\text{latent}}(\theta) + \lambda_2\mathcal{L}_{\text{value}}(\theta), \tag{5}$$

where $\mathcal{L}_{\text{value}}(\theta) = \mathbb{E}_t\left[(v_t - \hat{R}_t)^2\right]$ is the mean squared error (MSE) against the target return $\hat{R}_t$.

3.4 Adaptive Latent CoT Reasoning

While fixed-length reasoning provides a baseline, it imposes a static cognitive horizon that fails to capture the inherent diversity of robotic tasks. Enforcing a fixed latent length N_z incurs unnecessary computational overhead on highly predictable motions and restricts the reasoning capacity required for maneuvers demanding intricate cognitive planning. To address this, we introduce an **adaptive latent CoT mechanism** optimized via our proposed RL framework. By allowing the model to adapt its reasoning length based on reward signals, we enable an "early-exit" strategy that dynamically tailors the latent reasoning horizon to the specific demands of each task.

Dynamic Latent Generation. To enable adaptive reasoning, we shift the `<latent_end>` token from a deterministic sequence terminator to a dynamically emitted transition signal within the latent generation process. During autoregressive generation, if the policy predicts the `<latent_end>` token with a sufficiently high confidence probability $p \geq 0.99$ (where $p \in [0, 1]$), indicating that the model is highly certain the reasoning phase should conclude at the current decision step, the latent generation terminates and the model transitions to action prediction. To stabilize the training process and prevent erratic fluctuations in latent reasoning length, we restrict the emission of the `<latent_end>` token to a predefined set of M candidate positions, bounded by a maximum reasoning length N_{max}.

Exploration via Length Sampling. During RL rollouts, exploring various reasoning lengths is crucial for discovering the optimal cognitive horizon across diverse environmental states. We treat the reasoning length as a stochastic decision to encourage environmental interaction. Specifically, we extract the pre-softmax logits l associated with the `<latent_end>` token across all M predefined candidate positions, and then apply a temperature parameter β to control the exploration entropy, yielding a normalized categorical distribution over the candidate lengths:

$$p_m = \frac{\exp(l_m/\beta)}{\sum_{i=1}^{M}\exp(l_i/\beta)}, \quad \forall m \in \{1, \ldots, M\}. \tag{6}$$

The reasoning length index $m \sim \texttt{Categorical}(p_1, \ldots, p_M)$ is sampled to dictate the number of generated latent embeddings, facilitating exploration during rollout to discover adaptive cognitive horizons. During inference, the model adopts a confidence-based exit strategy for efficient execution.

Table 1: **Comparison on the LIBERO.** For RL training, we use single-trajectory warm-up and single-view data. Specifically, † denotes full-trajectory warm-up, and ‡ indicates two-camera views training. Results for LaST-R1 are averaged over 3 runs, with standard deviations reported.

Models	Paradigm	Spatial		Object		Goal		Long		Average	
		SR↑	Rank↓	SR↑	Rank↓	SR↑	Rank↓	SR↑	Rank↓	SR↑	Rank↓
OpenVLA [4]	SFT	84.7	12	88.4	12	79.2	12	53.7	12	76.5	12
GR00T-N1 [40]	SFT	94.4	8	97.6	8	93.0	8	90.6	7	93.9	8
π_0 [6]	SFT	96.8	7	98.8	4	95.8	7	85.2	8	94.2	7
$\pi_{0.5}$ [7]	SFT	98.8	4	98.2	7	98.0	5	92.4	5	96.9	5
OpenVLA-OFT [5]	SFT	97.6	6	98.4	6	97.9	6	94.5	2	97.1	4
GRAPE† [49]	RL	88.5	11	92.1	10	83.1	9	57.2	11	80.2	11
TGRPO† [53]	RL	90.4	9	92.2	9	81.0	11	59.2	10	80.7	10
VLA-RL† [25]	RL	90.2	10	91.8	11	82.2	10	59.8	9	81.0	9
SimpleVLA-RL [24]	RL	98.2	5	98.7	5	98.8	3	91.7	6	96.9	5
RLinf-GRPO [52]	RL	98.9	3	99.7	3	98.3	4	93.6	4	97.6	3
π_{RL}‡ [23]	RL	99.6	2	**100.0**	1	99.6	2	94.0	3	98.3	2
LaST-R1 (Ours)	RL	**99.87** ± 0.12	1	**100.00** ± 0.00	1	**100.00** ± 0.00	1	**99.73** ± 0.12	1	**99.90** ± 0.05	1

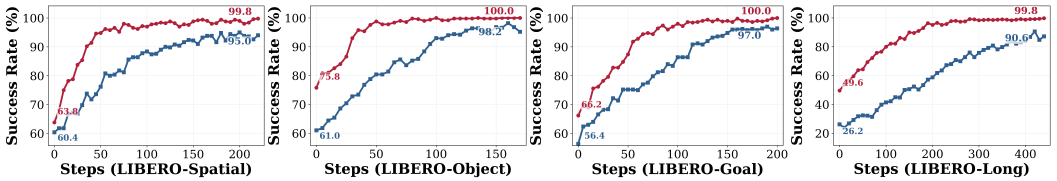


Figure 3: **One-shot Online RL Learning Curves on LIBERO.** We compare our proposed LaST-R1 optimized via LAPO (red), against the standard Action-Only baseline optimized via PPO (blue).

Adaptive Length Optimization. To optimize the adaptive length selection, the decision of ***when*** to stop reasoning must be explicitly supervised. We introduce an additional policy loss term for the `<latent_end>` token. Let $\mathbf{z}_t^{\text{end}}$ denote the transition token emitted at the sampled position m. We record its log-probability $\log \pi_\theta(\mathbf{z}_t^{\text{end}} \mid \cdot)$ during rollout. During the policy update, we construct a discrete likelihood ratio for the transition token: $r_t^{\text{end}}(\theta) = \exp\left(\log \pi_\theta(\mathbf{z}_t^{\text{end}} \mid \cdot) - \log \pi_{\theta_{\text{old}}}(\mathbf{z}_t^{\text{end}} \mid \cdot)\right)$. For models equipped with adaptive reasoning, the total training objective is augmented to include this transition-specific term $\mathcal{L}_{\text{end}}(\theta)$, weighted by a coefficient λ_3:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{action}}(\theta) + \lambda_1 \mathcal{L}_{\text{latent}}(\theta) + \lambda_2 \mathcal{L}_{\text{value}}(\theta) + \lambda_3 \mathcal{L}_{\text{end}}(\theta). \tag{7}$$

Guided by $\hat{A}_t$, this objective optimizes reasoning efficiency by rewarding shorter paths in predictable states while penalizing insufficient reasoning in scenarios requiring more cognitive planning.

4 Experiments

Following prior works [8, 9], LaST-R1 is first pre-trained on a custom-designed large-scale dataset, detailed in Appendix B.1. We then systematically evaluate our framework: Section 4.1 compares our method against SOTA baselines on the LIBERO benchmark, and Section 4.2 provides ablation studies on our core designs. Furthermore, Section 4.3 validates the performance of LaST-R1 in real-world robotic deployments, while Section 4.4 thoroughly investigates its generalization capabilities.

4.1 Simulation Experiment

Settings. We validate our method on four LIBERO task suites [56]: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long, each comprising 10 distinct tasks. All setups feature a simulated Franka Panda with a single front-view observation at 256 × 256 resolution. Following [24], we adopt a two-stage training pipeline. We first perform an SFT warm-up using a single randomly selected data per task, followed by online RL to iteratively collect rollouts and update the policy within the simulator. More training details are placed in Appendix C.2 and C.3. Finally, performance is measured by the average Success Rate (SR) across 50 held-out test scenarios for each task.

Baselines. We conduct a comprehensive comparison against both SFT-only and RL-trained models. All SFT baselines are trained on the full training dataset (50 trajectories per task). Among the RL-trained baselines, GRAPE [49] utilizes DPO [50], VLA-RL [25] and π_{RL} [23] rely on PPO [35], TGRPO [53], RLinf-GRPO [52], and SimpleVLA-RL [24] are optimized via GRPO [54]. All RL-trained baselines follow their official warm-up strategies, as detailed in Table 1.

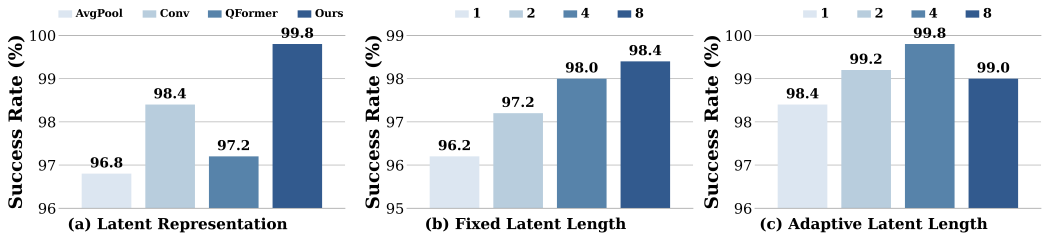


Figure 4: **Ablation Studies.** We evaluate (a) latent representation methods, (b) different fixed latent CoT lengths, and (c) adaptive CoT length with varying `<latent_end>` emission strategy.

Result Analysis. Table 1 presents the quantitative comparison across the LIBERO benchmark. Our proposed LaST-R1 achieves state-of-the-art performance, recording a near-perfect average success rate of 99.9% and ranking first across all four task suites. Notably, despite using only a single trajectory for warm-up, our method outperforms strong SFT baselines such as $\pi_{0.5}$ (96.9%) and OpenVLA-OFT (97.1%), which heavily rely on complete expert datasets. When compared to the leading RL-based method π_{RL} under the identical one-trajectory setting, LaST-R1 maintains a consistent advantage. This gap becomes particularly pronounced on the highly challenging LIBERO-Long suite (99.73% vs. 94.0%), demonstrating that our framework possesses superior capabilities in handling complex and long-horizon manipulation tasks. Ultimately, these results demonstrate that LaST-R1 substantially improves the physical world modeling capability of VLA models, leading to near-perfect action performance even with one-shot training data. Meanwhile, Figure 3 presents the learning curves across four task suites in LIBERO. LaST-R1 trained with LAPO achieves significantly faster convergence and higher success rates compared to the Action-Only baseline optimized via PPO. The baseline policy struggles with optimization efficiency, especially in the LIBERO-Long task suite, requiring significantly more steps yet still converging to sub-optimal performance. The results show that introducing latent CoT optimization during post-training acts as an effective cognitive buffer, smoothing the RL optimization landscape and improving sample efficiency. Furthermore, additional analyses are provided in Appendix D.2 and D.3, showing how this optimized reasoning mechanism yields both adaptive reasoning lengths across tasks and optimized execution episode lengths.

4.2 Ablation Studies

To validate our core designs, we conduct ablation studies evaluated on the LIBERO-Spatial suite unless otherwise specified.

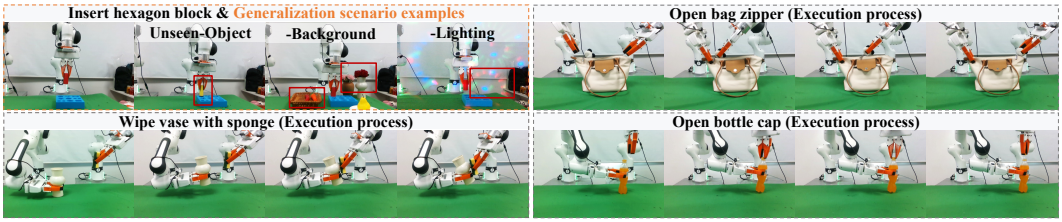
(1) Effectiveness of Latent Reasoning in SFT and RL Post-Training. As shown in Figure 3, one-shot SFT with latent reasoning (LaST-R1 Warm-up) achieves a 63.9% average SR, surpassing the Action-Only baseline (51.0%). The improvement is particularly pronounced on LIBERO-Long (long-horizon tasks), where incorporating latent CoT boosts performance from 26.2% to 49.6%. After RL post-training, our full method (LaST-R1 + LAPO) further amplifies this advantage, achieving 99.9% SR across all task suites and significantly outperforming Action-Only + PPO (95.2%). These results demonstrate that our latent reasoning mechanism enhances physical understanding, leading to more effective action optimization, as further evidenced by action-to-vision attention (Appendix E.1).

(2) Design Choices for Latent Representation. As shown in Figure 4 (a), we study the impact of different latent representation construction methods by comparing our DINOv3-based approach with three alternatives: averaging VLA visual encoder features [57] via global pooling [21], downsampling them with convolution [22], and extracting latents using a Q-Former [58]. Detailed implementations of these variants are deferred to Appendix C.1. Evaluated after RL post-training, our DINOv3 representation achieves the highest SR of 99.8%, outperforming Convolution (98.4%), Q-Former (97.2%), and Global Pooling (96.8%). This validates that leveraging pre-trained global representations from DINOv3 better preserves intricate spatial and semantic structures, while the top-k selection scheme maximally retains dynamic information.

(3) Impact of Latent Reasoning Length. We ablate the CoT horizon by varying the fixed latent sequence length $N_z \in \{1, 2, 4, 8\}$, as shown in Figure 4 (b). Compared to the Action-Only baseline (no latent reasoning, 95.0% SR), performance monotonically increases from 96.2% (1 token) to 98.4% (8 tokens). This indicates that a longer cognitive horizon effectively encodes richer condition information for downstream actions. Moreover, we observe that the performance gain from 4 to 8

Table 2: **Comparison on Real-World Tasks.** Evaluation includes the original training scenario and three generalization settings: unseen objects, background variations, and lighting conditions. Example cases are shown below, with changed configurations highlighted in red boxes.

Methods	Insert hexagon block				Open bag zipper			
	Original	Unseen-Object	-Background	-Lighting	Original	Unseen-Object	-Background	-Lighting
$\pi_{0.5}$ [7] (Full-size SFT)	70.00 ± 8.66	35.0 (-35.0)	55.0 (-15.0)	40.0 (-30.0)	75.00 ± 5.00	30.0 (-45.0)	70.0 (-5.0)	60.0 (-15.0)
LaST-R1 (Few-shot SFT → RL)	45.0 → 91.67 ± 2.89	75.0 (-16.67)	85.0 (-6.67)	80.0 (-11.67)	55.0 → 93.33 ± 2.89	80.0 (-13.33)	95.0 (+1.67)	90.0 (-3.33)
Methods	**Wipe vase with sponge**				**Open bottle cap**			
	Original	Unseen-Object	-Background	-Lighting	Original	Unseen-Object	-Background	-Lighting
$\pi_{0.5}$ [7] (Full-size SFT)	71.67 ± 5.77	45.0 (-26.67)	65.0 (-6.67)	50.0 (-21.67)	68.33 ± 2.89	50.0 (-18.33)	55.0 (-13.33)	55.0 (-13.33)
LaST-R1 (Few-shot SFT → RL)	65.0 → 95.00 ± 0.00	80.0 (-15.00)	90.0 (-5.00)	95.0 (-0.00)	45.0 → 95.00 ± 0.00	95.0 (-0.00)	80.0 (-15.00)	85.0 (-10.00)



tokens is marginal. To strike an optimal balance between task accuracy and inference speed, we cap the maximum reasoning sequence at 8 tokens.

(4) Adaptive CoT Length and `<latent_end>` Emission Strategy. Finally, we validate our adaptive latent CoT mechanism in Figure 4 (c). Given a maximum reasoning length $N_{\text{max}} = 8$, we ablate the number of valid candidate positions M for emitting `<latent_end>`. Specifically, we evaluate $M \in \{1, 2, 4, 8\}$ by uniformly distributing the candidate positions across the latent sequence. For example, $M = 1$ allows emission only after the 8th latent token, whereas $M = 8$ permits emission after every latent token. We find that $M = 4$ achieves the best performance (99.8%). While adaptive termination generally improves both efficiency and performance over the fixed-length baseline (98.4%), setting M too large ($M = 8$, 99.0%) slightly degrades performance. Therefore, a moderately constrained adaptive strategy ($M = 4$) offers the best trade-off between reasoning flexibility and optimization stability. More ablation studies are provided in Appendix D.1.

4.3 Real-World Experiment

Settings. We validate our method on four practical real-world manipulation tasks using Franka Research 3 arms, including one single-arm task and three dual-arm tasks: (1) *Insert hexagon block*, (2) *Open bag zipper*, (3) *Wipe vase with sponge*, and (4) *Open bottle cap*. For each task, we collect expert demonstrations using a space mouse. Given the significant domain gap between simulation and real-world deployment, we follow recent real-world VLA RL methods [26, 27] and adopt our proposed *latent reasoning-before-acting* RL framework, namely Off-policy LAPO. To improve RL update efficiency, we incorporate Low-Rank Adaptation (LoRA) [59] into all attention layers of LaST-R1, updating only the LoRA parameters. Additional implementation details are provided in Appendix C.4. For observations, we use one third-person camera and two wrist cameras, with input resolutions matching those in simulation. All policies are evaluated over 20 rollouts in 3 independent runs under varied positions, with task success determined by human evaluation. To highlight the advantages of our RL post-training framework, we compare LaST-R1 with the SOTA VLA model $\pi_{0.5}$, trained via full SFT on 100 expert trajectories.

Result Analysis. The "Original" column in Table 2 demonstrates the remarkable data efficiency of LaST-R1, where an initial warm-up with only 30 trajectories each task followed by online RL significantly outperforms the $\pi_{0.5}$ baseline trained with 100 expert trajectories via full supervised fine-tuning. Overall, this paradigm boosts the average SR substantially from an initial 52.5% to a final 93.75% ($\pm$ 1.25% std), surpassing the 71.25% ($\pm$ 5.0% std) average SR of the baseline $\pi_{0.5}$. The standard deviation (std) is computed over three independent runs. Specifically, the proposed RL method enhances visually guided precision in single-arm tasks by doubling the success rate of *Insert hexagon block* from 45% to 91.67%, while also improving dual-arm coordination for contact-rich manipulation like the *Open bag zipper* and *Wipe vase with sponge* tasks. Furthermore, the dramatic performance surge from 45% to 95% in the *Open bottle cap* task indicates that our post-training fundamentally strengthens the policy's ability to model temporally evolving physical

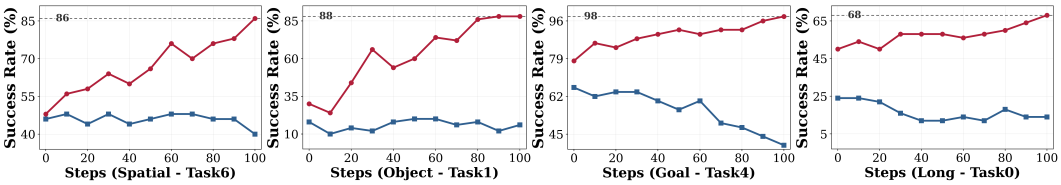


Figure 5: **Generalization Analysis on LIBERO.** While the OOD performance of the Action-Only PPO baseline (blue) stagnates, our LaST-R1 with LAPO (red) demonstrates continuous improvement.

interactions. More visualizations and real-world execution videos are provided in Appendix E.2 and the supplementary material.

4.4 Generalization Analysis

We evaluated LaST-R1's generalization in both simulation and the real world. **(1) Simulation.** Following the setup of prior work [24], for each task suite in LIBERO, we perform online RL on 9 "seen" tasks and hold out 1 task for out-of-distribution (OOD) evaluation, matching our one-trajectory warm-up setting. As shown in Figure 5, the standard Action-Only + PPO baseline suffers from severe overfitting, its OOD performance stagnates and even degrades on the LIBERO-Goal and LIBERO-Long suites. Conversely, our LaST-R1 + LAPO policy demonstrates continuous, significant OOD improvements across all suites. This demonstrates that our latent reasoning with LAPO progressively improve the model's understanding of scene and physical dynamics via environmental interaction, enabling it to extract transferable spatial and semantic concepts rather than overfitting to the training distribution. Additional experiments are presented in Appendix D.4. **(2) Real-World.** This superior adaptability seamlessly transfers to physical deployments. As shown in Table 2, when exposed to severe unseen perturbations, the full-size SFT baseline $\pi_{0.5}$ suffers catastrophic performance degradation, evidenced by up to a 45 percentage points drop when interacting with unseen objects. In stark contrast, LaST-R1 (Few-shot SFT $\rightarrow$ RL) maintains highly resilient execution. It successfully confines performance drops of at most 16.67 percentage points for novel objects, diverse background and lighting variations. This confirms that optimizing latent reasoning through environmental feedback intrinsically fortifies LaST-R1 VLA against environmental uncertainties.

5 Conclusion

In this work, we present LaST-R1, a tailored RL post-training framework for "latent reasoning-before-acting" policies with a unified VLA model coupling latent reasoning and action generation. By introducing latent Chain-of-Thought reasoning grounded in global visual representations, our approach enables structured modeling of physical dynamics prior to action execution. Building on this design, we propose Latent-to-Action Policy Optimization (LAPO), a novel RL paradigm that jointly optimizes reasoning and action spaces, allowing environmental feedback to shape both external behavior and internal reasoning processes. Furthermore, we introduce an adaptive latent CoT mechanism that dynamically adjusts the reasoning horizon based on task diversity, improving both efficiency and performance. Experimental results highlight the importance of integrating physical latent reasoning into VLA learning and show that jointly optimizing reasoning and action through RL is a promising direction for improving generalization and physical understanding. We hope this work provides a foundation for scaling latent reasoning-driven policies in complex robotic manipulation.

Acknowledgement

This study was supported in part by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government via the Hong Kong Centre for Logistics Robotics and by the following grant from the Hong Kong Innovation and Technology Fund (Project No. GHP/252/23SZ). This work was supported by the National Natural Science Foundation of China (62476011) and the Beijing Natural Science Foundation (L252060). This work was also supported by the National Natural Science Foundation of China (625B2007).

References

[1] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025.

[3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[4] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[5] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[7] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. π_0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[8] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

[9] Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning. *arXiv preprint arXiv:2506.01953*, 2025.

[10] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[12] Angen Ye, Zeyu Zhang, Boyuan Wang, Xiaofeng Wang, Dapeng Zhang, and Zheng Zhu. Vla-r1: Enhancing reasoning in vision-language-action models. *arXiv preprint arXiv:2510.01623*, 2025.

[13] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

[14] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

[15] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

[16] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.

[17] Chenyang Gu, Jiaming Liu, Hao Chen, Runzhong Huang, Qingpo Wuwu, Zhuoyang Liu, Xiaoqi Li, Ying Li, Renrui Zhang, Peng Jia, et al. Manualvla: A unified vla model for chain-of-thought manual generation and robotic manipulation. *arXiv preprint arXiv:2512.02013*, 2025.

[18] Zhuoyang Liu, Jiaming Liu, Jiadong Xu, Nuowei Han, Chenyang Gu, Hao Chen, Kaichen Zhou, Renrui Zhang, Kai Chin Hsieh, Kun Wu, et al. Mla: A multisensory language-action model for multimodal understanding and forecasting in robotic manipulation. *arXiv preprint arXiv:2509.26642*, 2025.

[19] 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 preprint arXiv:2506.21539*, 2025.

[20] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

[21] Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. Last _{0}: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026.

[22] Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

[23] Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, et al. πrl: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 2025.

[24] 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 preprint arXiv:2509.09674*, 2025.

[25] 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 preprint arXiv:2505.18719*, 2025.

[26] Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025.

[27] Qinwen Xu, Jiaming Liu, Rui Zhou, Shaojun Shi, Nuowei Han, Zhuoyang Liu, Chenyang Gu, Shuo Gu, Yang Yue, Gao Huang, et al. Twinrl-vla: Digital twin-driven reinforcement learning for real-world robotic manipulation. *arXiv preprint arXiv:2602.09023*, 2026.

[28] Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*, 2025.

[29] Yunfei Li, Xiao Ma, Jiafeng Xu, Yu Cui, Zhongren Cui, Zhigang Han, Liqun Huang, Tao Kong, Yuxiao Liu, Hao Niu, et al. Gr-rl: Going dexterous and precise for long-horizon robotic manipulation. *arXiv preprint arXiv:2512.01801*, 2025.

[30] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

[31] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[32] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[33] 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. *arXiv preprint arXiv:2412.13877*, 2024.

[34] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[36] 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. *arXiv preprint arXiv:2410.07864*, 2024.

[37] Yulin Luo, Hao Chen, Zhuangzhe Wu, Bowen Sui, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Qiuxuan Feng, Jiale Yu, Shuo Gu, et al. Look before acting: Enhancing vision foundation representations for vision-language-action models. *arXiv preprint arXiv:2603.15618*, 2026.

[38] 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. *arXiv preprint arXiv:2501.09747*, 2025.

[39] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Generalizable and interpretable robot foundation model via self-generated reasoning. *arXiv preprint arXiv:2412.03293*, 2024.

[40] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[41] Songming Liu, Bangguo Li, Kai Ma, Lingxuan Wu, Hengkai Tan, Xiao Ouyang, Hang Su, and Jun Zhu. Rdt2: Exploring the scaling limit of umi data towards zero-shot cross-embodiment generalization. *arXiv preprint arXiv:2602.03310*, 2026.

[42] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

[43] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[44] Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*, 2025.

[45] Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025.

[46] Shuanghao Bai, Jing Lyu, Wanqi Zhou, Zhe Li, Dakai Wang, Lei Xing, Xiaoguang Zhao, Pengwei Wang, Zhongyuan Wang, Cheng Chi, et al. Latent reasoning vla: Latent thinking and prediction for vision-language-action models. *arXiv preprint arXiv:2602.01166*, 2026.

[47] Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.

[48] Hongyin Zhang, Zifeng Zhuang, Han Zhao, Pengxiang Ding, Hongchao Lu, and Donglin Wang. Reinbot: Amplifying robot visual-language manipulation with reinforcement learning. *arXiv preprint arXiv:2505.07395*, 2025.

[49] Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Siwei Han, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. Grape: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024.

[50] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[51] Mingjie Pan, Siyuan Feng, Qinglin Zhang, Xinchen Li, Jianheng Song, Chendi Qu, Yi Wang, Chuankang Li, Ziyu Xiong, Zhi Chen, et al. Sop: A scalable online post-training system for vision-language-action models. *arXiv preprint arXiv:2601.03044*, 2026.

[52] Hongzhi Zang, Mingjie Wei, Si Xu, Yongji Wu, Zhen Guo, Yuanqing Wang, Hao Lin, Liangzhi Shi, Yuqing Xie, Zhexuan Xu, et al. Rlinf-vla: A unified and efficient framework for vla+ rl training. *arXiv preprint arXiv:2510.06710*, 2025.

[53] Zengjue Chen, Runliang Niu, He Kong, Qi Wang, Qianli Xing, and Zipei Fan. Tgrpo :fine-tuning vision-language-action model via trajectory-wise group relative policy optimization, 2025.

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

[55] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[56] 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*, 36:44776–44791, 2023.

[57] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[58] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[59] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[60] Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond images and language. *arXiv preprint arXiv:2511.21395*, 2025.

[61] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. In *RSS*, 2022.

[62] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2023.

[63] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022.

[64] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[65] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023.

[66] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, et al. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[67] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Conference on Robot Learning*, pages 885–897. PMLR, 2020.

[68] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. CLVR jaco play dataset, 2023.

[69] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[70] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023.

[71] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[72] Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, Chelsea Finn, and Abhinav Gupta. Train offline, test online: A real robot learning benchmark, 2023.

[73] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills, 2023.

[74] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.

[75] Jihoon Oh, Naoaki Kanazawa, and Kento Kawaharazuka. X-embodiment u-tokyo pr2 datasets, 2023.

[76] Vikash Kumar, Rutav Shah, Gaoyue Zhou, Vincent Moens, Vittorio Caggiano, Abhishek Gupta, and Aravind Rajeswaran. Robohive: A unified framework for robot learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[77] Tatsuya Matsushima, Hiroki Furuta, Yusuke Iwasawa, and Yutaka Matsuo. Weblab xarm dataset, 2023.

[78] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. FMB: A functional manipulation benchmark for generalizable robotic learning. `https://functional-manipulation-benchmark.github.io`, 2023.

[79] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *CoRL*, 2023.

[80] Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task agnostic offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[81] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[82] Abhishek Padalkar, Gabriel Quere, Antonin Raffin, João Silvério, and Freek Stulp. A guided reinforcement learning approach using shared control templates for learning manipulation skills in the real world. 2023.

[83] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. RoboTurk: A crowdsourcing platform for robotic skill learning through imitation. *CoRR*, abs/1811.02790, 2018.

[84] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. `https://sites.google.com/view/berkeley-ur5/home`.

[85] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`, 2022.

[86] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[87] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[88] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577, 2018.

[89] Jianlan Luo, Zheyuan Hu, Charles Xu, You Liang Tan, Jacob Berg, Archit Sharma, Stefan Schaal, Chelsea Finn, Abhishek Gupta, and Sergey Levine. Serl: A software suite for sample-efficient robotic reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16961–16969. IEEE, 2024.

[90] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025.

[91] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang 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 preprint arXiv:2506.18088*, 2025.

[92] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

A Derivation of the Latent Ratio

In our Latent-to-Action Policy Optimization (LAPO) framework, we optimize both the action tokens and the continuous latent reasoning embeddings. While the likelihood ratio for action tokens is straightforward to compute via categorical probabilities, the ratio for the continuous latent embeddings requires defining a density function over the high-dimensional continuous space to estimate. Here, we provide the detailed mathematical derivation for the decision step-level latent ratio $r_t^z(\theta)$.

1. Distribution Assumption

Let D denote the dimensionality of a single latent reasoning embedding. During policy optimization, we approximate the distribution of the generated continuous latents using an isotropic Gaussian distribution [60] centered at the current policy's output $\mathbf{z}_{t,i}^{\theta}$. We assume a fixed, predefined variance σ^2 across all dimensions. Thus, the probability density function (PDF) for a latent variable $\mathbf{x}$ generated by the current policy θ for the i-th latent embedding is given by:

$$\pi_\theta(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{z}_{t,i}^{\theta}\|^2\right). \tag{8}$$

2. Evaluating the Numerator $\pi_\theta(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot)$ and Denominator $\pi_{\theta_{\text{old}}}(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot)$

To compute the ratio for a single latent embedding sampled during a previous rollout, we must evaluate both the new policy θ and the old policy θ_{old} on the collected old latent embedding $\mathbf{z}_{t,i}^{\text{old}}$.

For the **numerator** $\pi_\theta(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot)$ (the probability of the old latent embedding under the *new* policy), we substitute $\mathbf{x} = \mathbf{z}_{t,i}^{\text{old}}$ into the PDF:

$$\pi_\theta(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{z}_{t,i}^{\text{old}} - \mathbf{z}_{t,i}^{\theta}\|^2\right). \tag{9}$$

For the **denominator** $\pi_{\theta_{\text{old}}}(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot)$ (the probability of the old latent embedding under the *old* policy), the mean of the old policy θ_{old} is, by definition, the sampled latent embedding $\mathbf{z}_{t,i}^{\text{old}}$. Consequently, the squared Euclidean distance becomes zero, i.e., $\|\mathbf{z}_{t,i}^{\text{old}} - \mathbf{z}_{t,i}^{\text{old}}\|^2 = 0$. The exponential term simplifies to $\exp(0) = 1$, leaving only the normalization constant:

$$\pi_{\theta_{\text{old}}}(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot) = \frac{1}{(2\pi\sigma^2)^{D/2}}. \tag{10}$$

3. Computing the Single Latent Embedding Ratio

By dividing the numerator $\pi_\theta(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot)$ by the denominator $\pi_{\theta_{\text{old}}}(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot)$, the complex normalization constant $\frac{1}{(2\pi\sigma^2)^{D/2}}$ perfectly cancels out. The ratio for a single latent embedding i is elegantly reduced to the negative exponentiated distance:

$$r_{t,i}^z(\theta) = \frac{\pi_\theta(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot)}{\pi_{\theta_{\text{old}}}(\mathbf{z}_{t,i}^{\text{old}} \mid \cdot)} = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{z}_{t,i}^{\text{old}} - \mathbf{z}_{t,i}^{\theta}\|^2\right). \tag{11}$$

4. Sequence-Level Latent Reasoning Embeddings Aggregation

The latent reasoning process consists of a sequence of N_z latent embeddings. Assuming the continuous latents are conditionally independent given the rollout context, the joint ratio for the entire latent sequence $\mathbf{Z}_t$ is the product of the individual embedding ratios. Since the product of exponentials equals the exponential of their sum, we arrive at our final formulation:

$$r_t^z(\theta) = \prod_{i=1}^{N_z} r_{t,i}^z(\theta) = \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N_z}\|\mathbf{z}_{t,i}^{\text{old}} - \mathbf{z}_{t,i}^{\theta}\|^2\right). \tag{12}$$

B Dataset Details

B.1 Large Scale Pre-Training Datasets

To ensure LaST-R1 inherits a robust foundation of motor primitives and physical common sense, we curated a diverse corpus of 400K trajectories (28M frames) from the Open-X-Embodiment [32],



(a) Single-arm setup of Franka robot. (b) Dual-arm setup of Franka robot.

Figure 6: **Real-World Robot Experiment Setups of Franka Research 3 Arms.**

DROID [34], and RoboMIND [33] repositories. Table 3 details the proportion of each dataset used in our pre-training mixture. Notably, beyond adhering to standard data quality filtering practices established by prior VLA works [21], we additionally ensure that all robot state annotations are accurate and physically consistent. Crucially, to empower the reasoning process without introducing computational bottlenecks during training, we precompute the DINOv3-based latent tokens for all pre-training frames offline. As detailed in Section 3.2, this involves extracting the <CLS> token from DINOv3 and applying top-k selection to generate dense, semantically rich image representations. Because these spatial and temporal latent targets are processed entirely offline, they are simply loaded alongside the standard visual and textual inputs during this pretraining stage. This strategy provides the model with highly informative cognitive anchors to learn environmental dynamics and spatial reasoning, while adding virtually zero computational overhead to the pre-training pipeline. Ultimately, this stage establishes a shared representation space across diverse robotic datasets, enabling a seamless integration of reasoning and execution within the unified VLA framework.

Table 3: **Datasets for Pre-training.** The names of selected datasets for large-scale pretraining and their sampling ratios (%).

Training Dataset Mixture			
Dataset	**Ratio (%)**	**Dataset**	**Ratio (%)**
BridgeV2 [61, 62]	20.82	Nyu Franka Play [63]	0.24
Kuka [64]	20.22	Stanford Hydra [65]	0.20
Fractal [66]	13.67	RoboMIND [33]	0.20
Robo-Net [67]	11.53	Jaco Play [68]	0.19
Language Table [69]	7.72	Dobb-E [70]	0.18
BC-Z [71]	7.54	Toto [72]	0.17
Maniskill [73]	5.26	Furniture Bench [74]	0.09
DROID [34]	4.82	Utokyo Pr2 Tabletop [75]	0.04
Roboset [76]	3.21	Utokyo Xarm Pap [77]	0.04
FMB Dataset [78]	1.50	CMU Stretch [79]	0.02
Taco Play [80, 81]	1.26	DLR Sara Grid Clamp [82]	0.02
RoboTurk [83]	0.70	Utokyo Pr2 Fridge [75]	0.01
Berkeley Autolab Ur5 [84]	0.35	—	—

B.2 Real-World Data Collection

We configure the dual-arm experimental setup with two Franka Research 3 robotic arms equipped with 3D-printed UMI grippers. A single Intel RealSense D455 camera provides a third-person view, while two D435 cameras are mounted for wrist views. Inference is performed on an NVIDIA RTX 4090 GPU. Both SFT data collection and human-in-the-loop real-world RL are conducted using a 3D SpaceMouse. The main assets used throughout our experiments are shown in Figure 6.

1. Insert hexagon block. This task requires the robot to hold a hexagonal block and accurately insert it into a matching slot on a base. Success depends on the model's ability to perform precise 6-DoF

spatial alignment and maintain accurate trajectory control to handle the tight physical clearance between the object and the target. **Success criterion:** The robot must accurately align and fully insert the block into the slot without toppling the base.

2. Open bag zipper. The robot is required to use dual arms cooperatively, with one arm stabilizing a soft bag while the other grasps and pulls the zipper. This task evaluates the model's capacity for bimanual coordination and its ability to execute a smooth, continuous pulling trajectory while interacting with a deformable object. **Success criterion:** The robot must cooperatively stabilize the bag with one arm and completely pull the zipper from the starting end to the very end with the other.

3. Wipe vase with sponge. This task requires the robot to hold a vase securely with one arm while using a sponge in the other to wipe its exterior. This demands highly accurate bimanual synchronization and the ability to predict and maintain continuous, stable contact along a curved 3D spatial trajectory. **Success criterion:** The robot must maintain continuous, stable physical contact between the sponge and the vase surface along the entire specified wiping trajectory.

4. Open bottle cap. The robot must establish a stable grasp on a bottle with one arm while the other arm twists off its cap. This task evaluates the model's ability to execute fine motor control, specifically coordinating precise rotational manipulation and applying appropriate torque while simultaneously stabilizing the base object. **Success criterion:** The robot must securely grasp the bottle and apply sufficient torque to fully twist off and completely separate the cap from the bottle.

C Training Details

C.1 Alternative Latent Implementation Details

To systematically evaluate the architectural design of our latent reasoning tokens, we compare our proposed DINOv3-based representation against three alternative compression strategies: **1) Global Pooling**, **2) Convolutional Downsampling**, and **3) a Lightweight Q-Former** [58]. The most straightforward baseline, Global Pooling, trivially collapses the N_v visual tokens from the VLA vision encoder into a single vector via token-dimensional average pooling, which operates without additional parameters but severely sacrifices fine-grained spatial details. To explicitly model the spatial hierarchy, the Convolutional baseline reshapes the N_v visual tokens back into a 2D feature map and applies a sequence of 1×1 projections, aggressive strided downsampling (stride of 8), and global spatial convolutions with GELU activations to squeeze the map into a single flattened token. Alternatively, the Q-Former baseline employs an attention-based mechanism where a single learnable query adaptively aggregates task-relevant context from the dense N_v visual tokens through stacked Pre-LayerNorm cross-attention and MLP blocks. While these baselines introduce varying degrees of parametric complexity to compress visual features, our proposed approach simply extracts the `<CLS>` token from a pre-trained DINOv3 model and applies top-k dimension selection entirely offline. By directly harnessing the structurally rich and semantically dense feature space of a powerful vision foundation model, our DINOv3-based method circumvents the information loss inherent to naive pooling, spatial downsampling, or from-scratch query optimization. Consequently, this offline extraction strategy provides an optimal, computationally free cognitive anchor, ultimately achieving the best reasoning performance and highest success rates in our empirical evaluations.

C.2 Warm-up Details

Our policy is initialized with pre-trained Qwen3-VL-4B [2] weights, expanding the tokenizer vocabulary to include discrete action tokens (`<action_i>` for $i \in [0, 255]$) and a special transition token (`<latent_end>`). As illustrated in Figure 7, we adopt a hybrid decoding architecture: latent tokens are generated autoregressively to exploit the VLM's robust reasoning, while action tokens are decoded in parallel with a chunk size of 8 for physical execution efficiency. During training, we tailor the latent length strategy to the target domain. For the multi-task LIBERO benchmark, we encourage length exploration by uniformly sampling the reasoning length from $\{2, 4, 6, 8\}$ per sample (anchored on a maximum length of 8 with $M = 4$ candidate positions); conversely, for single-task real-world deployments, we strictly enforce a fixed latent length of 8. The network is optimized using a joint objective comprising a cosine similarity loss for the predicted latents against offline ground-truth (GT) latents, a Cross-Entropy (CE) loss for the `<latent_end>` token, and a standard CE loss for the parallel-decoded action tokens. These components are empirically weighted at $1 : 0.1 : 1$ to

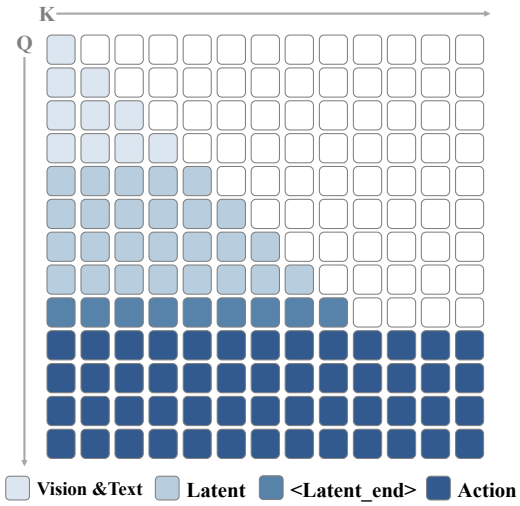


Figure 7: **Hybrid Attention Mask Design.** Our model employs a custom attention mask to unify autoregressive reasoning and parallel action execution. The vision and text prompts, alongside the latent reasoning tokens, utilize a causal lower-triangular mask for sequential generation. After the `<Latent_end>` transition token aggregates the full reasoning context, the action tokens employ a bidirectional mask. This allows all action tokens within a chunk to attend to the entire historical context as well as to each other, enabling efficient parallel decoding.

balance token-level gradients. Training is distributed across 8 $\times$ H20 GPUs using Accelerate [85] and DeepSpeed in `bf16` mixed precision, with a global batch size of 64 and the AdamW [86] optimizer (peak learning rate 1×10^{-5}, cosine decay with a minimum ratio of 0.1). Finally, we customize the training duration per environment: the four LIBERO suites are trained in an extremely low-data regime (1 expert trajectory per task) for 10K iterations each, whereas real-world models are fine-tuned on 30 trajectories per task for 1K iterations.

C.3 RL Training Details on LIBERO.

We implement our Latent-to-Action Policy Optimization (LAPO) framework using `verl` [87] with Ray [88] and Fully Sharded Data Parallel (FSDP) on a single 8 $\times$ H20 GPU node. The VLA policy is initialized from the SFT warm-up checkpoint for continuous online interaction. During rollouts, the policy decodes an action chunk of 8 future action steps in parallel (yielding exactly 56 tokens per chunk) with a sampling temperature of 1.6, while trajectory lengths are capped at 240 steps for LIBERO-Spatial, 320 for LIBERO-Object/LIBERO-Goal, and 576 for LIBERO-Long. A strict verifier mechanism yields a sparse binary reward (0 or 1) for task success, which is scaled by a factor of 5 at the terminal step. Advantages are computed via GAE ($\gamma = 0.99$, $\lambda = 0.95$) with valid-timestep masking applied to accommodate variable-length trajectories. At each RL iteration, a rollout batch of 512 sampled trajectories is evenly partitioned into 4 mini-batches to perform 4 PPO optimization epochs. We train the policy using learning rates of 3×10^{-5} for the VLA actor and 3×10^{-4} for the value head. To ensure stability and efficiency, we apply global gradient clipping at 10, asymmetric PPO clipping bounds ($\epsilon_{\text{min}} = 0.2$, $\epsilon_{\text{max}} = 0.28$), and gradient/optimizer state offloading, alongside evaluating the model's success rate every 5 update steps.

C.4 RL Training Details on Real-World

Building upon the continuous asynchronous actor-learner pipeline [89, 90], policy rollout and model optimization run concurrently without pausing data collection. The actor continuously executes the policy—which predicts actions conditioned on language prompts and three-view RGB observations—and streams transition tuples to the learner. To preserve critical corrective signals, human interventions are routed to a dedicated buffer, allowing the learner to sample mixed mini-batches from both demonstration and rollout streams. We initialize the LaST-R1 model using the supervised

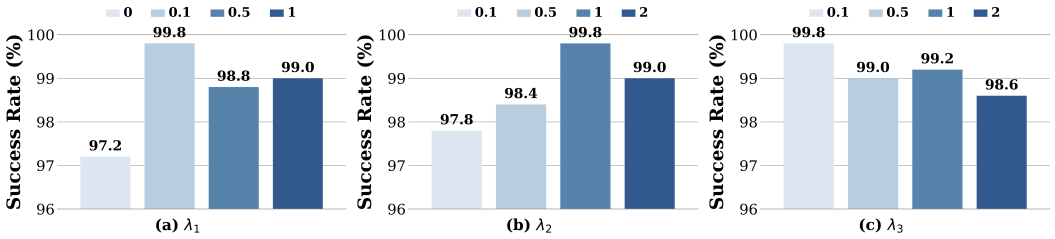


Figure 8: **Ablation Studies on Loss Coefficients.** Performance impact of varying (a) latent loss weight λ_1, (b) state-value weight λ_2, and (c) transition penalty λ_3. In each sub-figure, unlisted coefficients are fixed at their optimal values. The combination ($\lambda_1 = 0.1, \lambda_2 = 1, \lambda_3 = 0.1$) achieves the optimal gradient balance and highest success rate.

warm-up checkpoint, and to ensure stable and computationally tractable online updates, we freeze the base model and exclusively update newly injected Low-Rank Adaptation (LoRA) [59] parameters with rank $r = 32$. Online optimization commences after collecting an initial replay threshold of 500 transitions. The learner optimizes a joint objective combining behavior cloning ($\lambda_{\text{BC}} = 1.0$) and Q-guided policy improvement ($\lambda_Q = 0.5$) using the AdamW [86] optimizer with a learning rate of 1×10^{-5}, zero weight decay, and gradient clipping at a maximum norm of 1.0. We apply a critic-to-actor update ratio of 2:1, where the critic utilizes one-step temporal-difference targets with a discount factor of $\gamma = 0.98$ and target value parameters are softly updated at a rate of $\tau = 0.005$. Gradient accumulation is set to 16 micro-steps with a micro-batch size of 2 per stream, yielding an effective batch size of approximately 32. Finally, to handle sparse real-world feedback, the environment provides a terminal reward of $+10$ upon operator-confirmed task completion, coupled with a constant step penalty of -0.05 to encourage execution efficiency, while the learner periodically broadcasts updated weights back to the actor for hot-loading.

D Additional Quantitative Analysis

D.1 Additional Ablation Studies on Hyperparameters

To systematically evaluate the impact of the weighting hyperparameters in our joint training objective $\mathcal{L}_{\text{total}}(\theta)$, we conduct an ablation study varying the coefficients λ_1, λ_2, and λ_3 on LIBERO-Spatial. As shown in Figure 8, in each experiment, we isolate the effect of one coefficient while fixing the others at their empirically optimal values (i.e., $\lambda_1 = 0.1$, $\lambda_2 = 1$, $\lambda_3 = 0.1$). First, analyzing the latent loss weight λ_1, we observe that relying entirely on implicit gradient backpropagation from the action loss ($\lambda_1 = 0$) yields a suboptimal success rate of 97.2%. Introducing explicit latent supervision significantly improves performance, peaking at 99.8% with $\lambda_1 = 0.1$. However, excessively high values (e.g., $\lambda_1 = 1$) slightly degrade performance to 99.0%, likely because the latent objective begins to overshadow the primary physical execution task. Second, for the state-value estimation weight λ_2, a balanced weight of $\lambda_2 = 1$ achieves the best result (99.8%). Lower weights such as 0.1 and 0.5 degrade the success rate to 97.8% and 98.4% respectively, underscoring the necessity of robust value estimation for accurate advantage computation in our LAPO framework. Finally, evaluating the transition weight λ_3 reveals that a modest penalty of $\lambda_3 = 0.1$ is optimal (99.8%). Increasing this weight up to 2 forces a sharp performance drop to 98.6%, indicating that over-penalizing the `<latent_end>` token disrupts the delicate exploration balance required for the policy to learn state-conditional reasoning lengths. Consequently, the configuration of $\lambda_1 = 0.1$, $\lambda_2 = 1$, and $\lambda_3 = 0.1$ establishes the effective balance across action generation, internal reasoning, and value estimation.

D.2 Analysis of Adaptive Reasoning Length

To investigate the efficacy of our dynamic reasoning mechanism, we analyze the frequency distribution of the chosen latent reasoning lengths (bounded by a maximum of 8 tokens) across the four LIBERO task suites. Figure 9 illustrates the contrast between the policy right after SFT warm-up and the fully RL-optimized policy. During the warm-up phase, where reasoning lengths are sampled randomly, the pre-RL model exhibits a relatively uniform distribution across lengths 2, 4, 6, and 8. This indicates a lack of cognitive prioritization, treating simple and complex states with equal computational budgets.

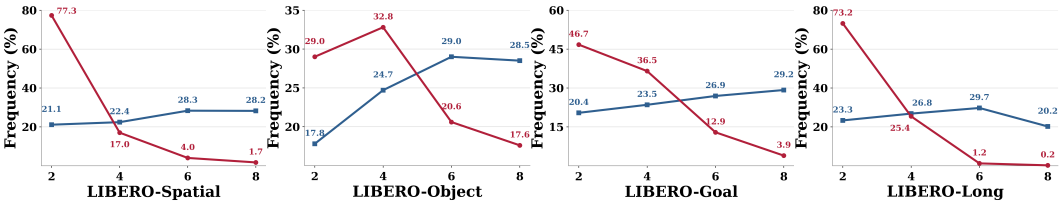


Figure 9: **Frequency Distribution of Adaptive Latent Reasoning Lengths.** Compared to the SFT warm-up (blue), the RL-optimized policy (red) successfully learns the optimal number of latent reasoning steps, achieving an excellent balance between reasoning precision and inference efficiency.

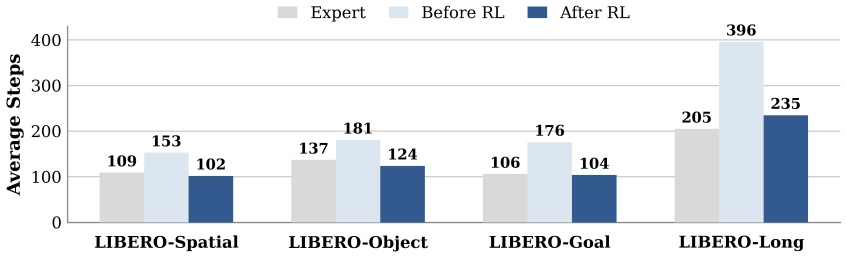


Figure 10: **Comparison of Average Execution Steps Across LIBERO Task Suites.** We report the average steps taken by the expert demonstrations, the pre-RL policy, and the RL-optimized policy. After optimization with LAPO, LaST-R1 not only drastically reduces the execution steps compared to the imitation baseline, but even surpasses the temporal efficiency of experts in most scenarios.

Following Latent-to-Action Policy Optimization (LAPO), the length distribution undergoes a dramatic shift. Guided by the environment reward and our transition-specific optimization objective, the model effectively learns an "early-exit" strategy. Across all environments, the RL-optimized policy heavily gravitates toward shorter cognitive horizons, with reasoning lengths of 2 or 4 tokens dominating the decision steps. This empirical behavior strongly validates our design: rather than passively executing fixed-length reasoning, LaST-R1 proactively learns to terminate its internal deliberation once sufficient state representations are formed. By drastically reducing the token count per step without compromising the near-perfect success rates, the model achieves an optimal balance between robust physical reasoning and inference efficiency.

D.3 Episode Length Comparisons

To further demonstrate the efficacy of our framework, we evaluate the average number of execution steps across 500 test trajectories for each LIBERO suite. Notably, these averages are computed unconditionally over all rollouts, regardless of ultimate task success or failure. As illustrated in Figure 10, the policy trained after warm-up SFT (Before RL) requires significantly more steps than the simulator-provided expert demonstrations. This inflation in trajectory length stems from compounding errors and purely reactive behaviors; when the pre-RL model encounters out-of-distribution (OOD) states, it often gets trapped in inefficient loops or prolonged, unsuccessful wandering that approaches the maximum step limit. However, following online interaction, the RL-optimized policy (After RL) exhibits a dramatic reduction in execution steps, notably outperforming even the original expert trajectories in the Spatial, Object, and Goal task suites. We attribute this exceptional temporal efficiency directly to the optimized latent reasoning process. Rather than passively mimicking the scripted expert data, which often relies on conservative motion planning and rigid, waypoint-based trajectories, the RL-driven reasoning strategy empowers the policy with a deeper, more generalized understanding of the task dynamics. By jointly optimizing the reasoning and action spaces, the model learns to synthesize highly efficient internal representations, thereby discovering optimal shortcuts and formulating more direct, precise, and decisive trajectories than the original algorithmic experts.

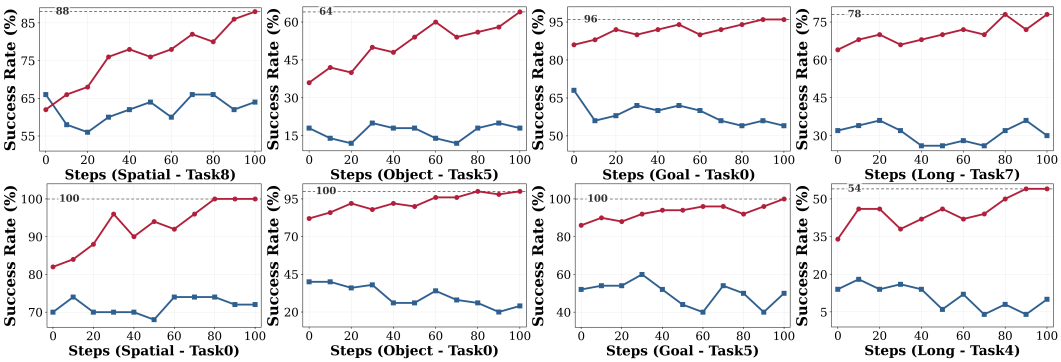


Figure 11: **Generalization Analysis on LIBERO.** For each task suite, models are warmed up with one trajectory per task, followed by online RL training on 9 tasks, with the remaining 1 task for evaluation. While the out-of-distribution performance of the Action-Only PPO baseline (blue) stagnates, our LaST-R1 with LAPO (red) shows continuous generalization improvements across all task suites.

D.4 Additional Generalization Analysis

To provide a more granular understanding of the out-of-distribution (OOD) learning process, Figure 11 details the step-by-step generalization curves for two representative held-out tasks from each LIBERO suite. The learning dynamics reveal a stark contrast between the two paradigms throughout the online interaction phase. The Action-Only PPO baseline (blue) exhibits a classic overfitting pathology. Across almost all tasks, its OOD success rate flatlines early in training and frequently oscillates or even collapses (e.g., Spatial-Task8 and Long-Task7). This empirical evidence confirms that optimizing purely in the action space forces the model to memorize the specific kinematic trajectories of the 9 training tasks, leaving it entirely brittle and incapable of adapting when faced with the unseen spatial configurations or objects of the held-out task. In sharp contrast, our LaST-R1 optimized with LAPO (red) displays a remarkably stable and continuous upward trajectory across the entire training process. The curves demonstrate no signs of catastrophic forgetting or OOD performance degradation. Notably, our policy not only rapidly converges to a perfect 100% OOD success rate on several tasks (Spatial-Task0, Object-Task0, and Goal-Task5), but also shows robust, monotonic growth on the most challenging multi-stage tasks. For instance, in Long-Task4, while the action-only baseline completely fails to surpass a 20% success rate due to compounding errors, our method steadily climbs to 54%. This extended analysis clearly illustrates that jointly optimizing the latent reasoning space empowers the policy to abstract high-level semantic and physical principles, allowing it to dynamically compose learned skills for novel scenarios rather than rigidly overfitting to the training distribution.

D.5 Comparison with Additional RL Baseline

To further evaluate our method, we introduce an additional contrastive baseline, ConRFT [26], which uses an action-only model under the exact same real-world hardware setup. Table 4 presents the quantitative comparison between ConRFT and our proposed LaST-R1 across the four evaluated tasks.

Table 4: Success rates of ConRFT and LaST-R1 on real-world tasks.

Method	Insert hexagon block	Open bag zipper	Wipe vase with sponge	Open bottle cap
ConRFT [26]	76.67 ± 2.89	60.00 ± 5.00	76.67 ± 2.89	68.33 ± 7.64
LaST-R1 (Ours)	**91.67 ± 2.89**	**93.33 ± 2.89**	**95.00 ± 0.00**	**95.00 ± 0.00**

As the results demonstrate, LaST-R1 significantly outperforms ConRFT under the exact same online interaction paradigm. This substantial performance gap further validates the algorithmic superiority of jointly optimizing the latent reasoning and action spaces, as opposed to relying purely on action-only optimization.

D.6 Analysis of Human Interventions during Real-World Policy Optimization

In our framework, the "human-in-the-loop" component functions primarily as an early-stage exploration strategy [26]. A common challenge in reinforcement learning (RL) is that a low initial success rate can significantly hinder effective exploration. To address this, we utilize a budget for human interventions to guide the policy during its initial training phases. To quantify the extent of human involvement during the training process, we tracked the intervention ratio across different stages of policy optimization. Table 5 presents the intervention ratio at various training iterations for the *Insert Hexagon Block* task as a representative example.

Table 5: Human intervention ratio at different iterations for the *Insert Hexagon Block* task.

Iterations	**400**	**800**	**1200**	**1600**
Intervention Ratio	34.04%	20.88%	4.55%	2.83%

As shown in the results, the reliance on human interventions is highest during the early stages of optimization (34.04% at 400 iterations) to kickstart exploration. However, as the policy improves through RL, the need for human guidance drops significantly, falling to just 2.83% by 1600 iterations. This trend is consistent across all real-world tasks evaluated in our study, demonstrating that our approach effectively leverages brief human guidance without requiring long-term human involvement.

D.7 Evaluation on Additional Benchmark: RoboTwin 2.0

To validate generalization beyond the LIBERO benchmark and to ensure that our observed gains are not subject to benchmark-specific overfitting, we conducted additional evaluations on the RoboTwin 2.0 benchmark [91]. To maintain a fair and rigorous assessment of the generalization capabilities of our model, we strictly adhered to the experimental protocols established in SimpleVLA-RL [24]. The evaluation results across four challenging tasks are summarized in Table 6.

Table 6: Success rates on the RoboTwin 2.0 benchmark. Best results are highlighted in bold.

Method	`move_can_pot`	`beat_block_hammer`	`place_phone_stand`	`handover_mic`
π_0 [6]	41.0	59.0	22.0	96.0
RDT [36]	33.0	22.0	13.0	95.0
OpenVLA-OFT [5]	28.1	28.1	17.1	45.3
SimpleVLA-RL [24]	61.2	87.5	39.6	89.2
LaST-R1 (Ours)	**69.5**	**92.2**	**55.1**	**97.3**

As demonstrated in the results, LaST-R1 significantly outperforms both the SimpleVLA-RL baseline and other strong SFT-based methods under the same evaluation protocol. These findings firmly validate that the representational benefits of our latent reasoning and the algorithmic superiority of the LAPO framework are not overfitted to the LIBERO benchmark. Instead, they successfully transfer to diverse and complex dual-arm tasks, yielding a highly robust policy.

E Additional Qualitative Analysis

E.1 Action-to-Vision Attention

To gain deeper insights into the internal decision-making mechanisms of our policy, we utilize Grad-CAM [92] to visualize the cross-attention weights from the action tokens to the visual tokens. As illustrated in Figure 16, we compare spatial attention maps across four representative trajectories from the LIBERO task suites to evaluate the impact of explicit latent reasoning and our LAPO framework. Examining the SFT-trained models, the Action-Only policy frequently exhibits diffused and scattered attention, struggling to consistently localize task-relevant entities and often lagging behind the robot's end-effector. In stark contrast, by generating latent reasoning tokens prior to action decoding, our proposed LaST-R1 establishes a strong semantic anchor, resulting in highly concentrated and precise visual grounding on critical objects of interaction. These advantages become even more pronounced after reinforcement learning post-training. While applying standard PPO to the Action-Only model

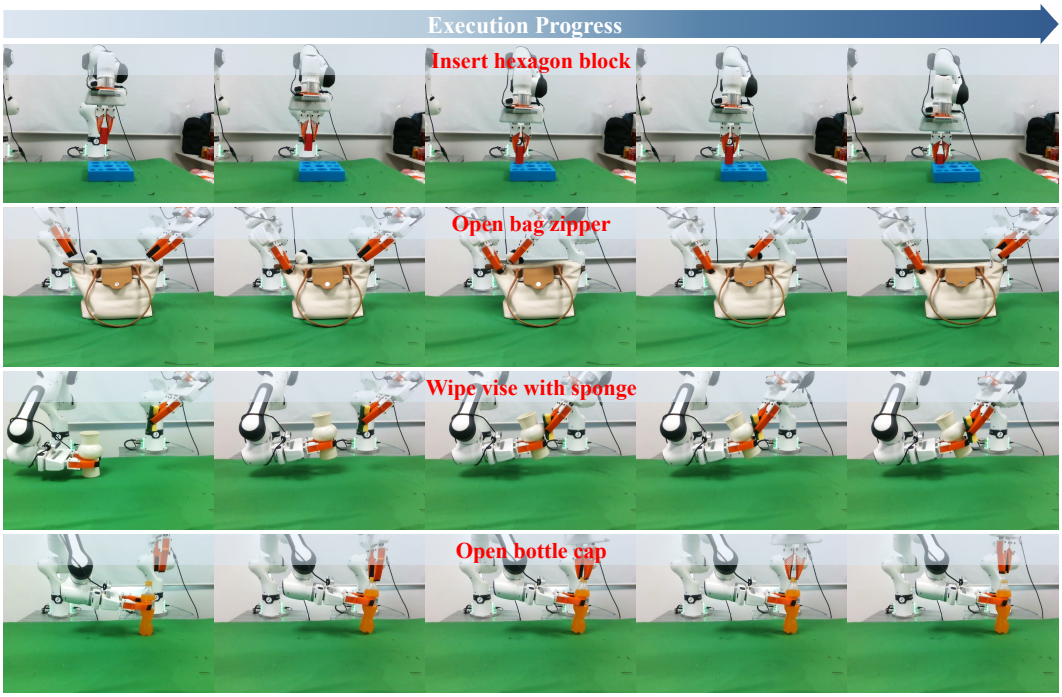


Figure 12: **Real-World Execution Trajectories of the Proposed Policy.** The sequences illustrate the continuous execution progress across diverse robotic manipulation tasks.

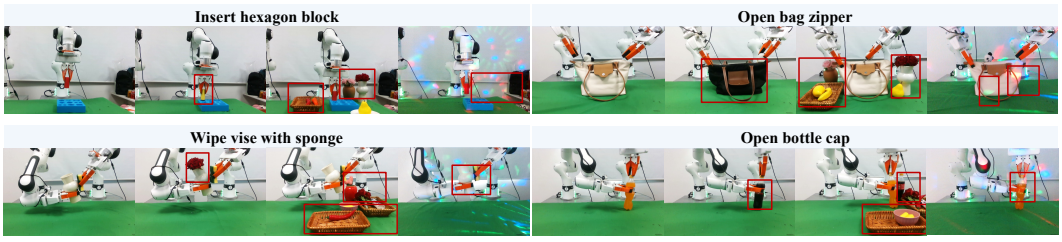


Figure 13: **Policy Robustness Under Unseen Configurations.** The model consistently maintains task success when subjected to severe visual variations, effectively ignoring out-of-distribution distractors, unseen object instances, and dynamic lighting conditions (indicated by red bounding boxes).

tends to over-focus on the gripper's immediate vicinity and lacks long-horizon awareness, LaST-R1 + LAPO achieves the most robust, goal-directed attention. Because LAPO jointly optimizes the latent reasoning space and the action space using the environment reward, the policy learns to dynamically shift its intense visual focus from the manipulated object to the target receptacle as the trajectory progresses, confirming that our framework successfully aligns internal visual cognition with precise physical execution.

E.2 Real-World Visualization

To evaluate the practical deployment capabilities of LaST-R1, we visualize its execution trajectories across diverse real-world manipulation tasks, as shown in Figure 12. As demonstrated, the policy exhibits stable closed-loop control during standard execution (e.g., precise insertion, bimanual coordination). Crucially, the model demonstrates remarkable robustness in visually cluttered environments, as shown in Figure 13. Despite severe out-of-distribution perturbations—such as dynamic lighting, novel distractor objects, and background variations—LaST-R1 consistently maintains execution stability, highlighting its strong generalization capabilities in complex, unconstrained scenarios.

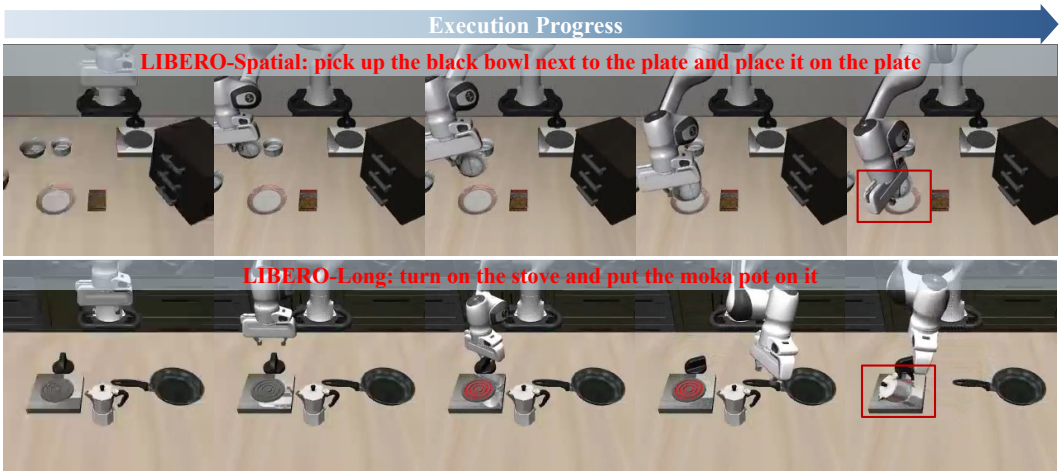


Figure 14: **Failure Cases in the LIBERO.** Red boxes highlight the moments of failure.



Figure 15: **Visualizations of Real-World Failure Cases.** The sequences illustrate typical execution errors across four tasks, with red bounding boxes highlighting the specific moments of failure.

F Failure Case Analysis

F.1 Simulation.

In the rare failure cases observed, the model's errors stem from fine-grained physical execution. As shown in Figure 14, in the LIBERO-Spatial task: *pick up the black bowl next to the plate and place it on the plate*, the robot successfully grasps the bowl but releases it slightly off-center, causing it to slip off the edge of the plate due to imperfect placement precision. Similarly, in the LIBERO-Long task: *turn on the stove and put the moka pot on it*, while the robot correctly turns on the stove, its arm inadvertently knocks over the moka pot during the approach phase. This irreversible change

in the object's pose prevents subsequent grasping, highlighting the inherent long-tail challenge of maintaining strictly collision-free trajectory planning in complex, multi-step environments.

F.2 Real-World.

In real-world deployments, the observed failure cases primarily stem from fine-grained execution imperfections and complex contact dynamics. As shown in Figure 15, for the *Insert hexagon block* task, the failure is caused by a slight positional deviation during the final insertion phase, preventing the block from fitting into the slot. In the *Open bag zipper* task, the robot struggles with dynamic friction, causing the gripper to slip and detach from the zipper mid-pull. For the *Wipe vase with sponge* task, the failure occurs because the robot fails to establish actual physical contact between the sponge and the vase surface, resulting in an ineffective wiping motion. Finally, in the *Open bottle cap* task, the robot successfully grasps and twists the cap but terminates the action prematurely, failing to unscrew the cap completely. These cases indicate that although LaST-R1 achieves remarkable performance in handling continuous, high-precision physical feedback in real-world contact-rich scenarios, a certain probability of failure still exists.

G Limitations and Future Work

While LaST-R1, optimized via the proposed LAPO framework, exhibits exceptionally strong performance across both the LIBERO benchmark and real-world deployments, it has several limitations that present exciting avenues for future research. First, to strike an optimal balance between reasoning capacity and inference speed, our current adaptive CoT mechanism caps the maximum latent sequence length at 8 and restricts the early-exit transitions to a predefined set of M candidate positions. Moving forward, we aim to refine the latent representation mechanism and explore more flexible, fully dynamic reasoning architectures that can halt deliberation without manually designed constraints. Second, our current real-world evaluations are conducted exclusively on single-arm and dual-arm manipulation platforms. To demonstrate the scalability and universality of our framework, we plan to deploy LaST-R1 to more complex robotic embodiments, extending our latent reasoning paradigm to multi-fingered dexterous hands and whole-body humanoid robot control.

H Broader Impact

Our work proposes LaST-R1 VLA, a foundation model for robotic manipulation that seamlessly integrates adaptive-length latent reasoning and parallel action execution within a unified framework. While our method significantly enhances spatial awareness and dynamically allocates cognitive compute through Latent-to-Action Policy Optimization (LAPO), deploying high-capacity VLA models in physical environments inherently introduces potential risks. These risks primarily involve unpredictable physical behaviors under out-of-distribution scenarios or misinterpretation of ambiguous human instructions. To mitigate these challenges, future real-world deployments should incorporate rigorous safety guardrails and strict hardware-level operational constraints. Broadly, our post-training framework delivers strong performance across diverse downstream tasks, facilitating the transition of robotic systems from controlled laboratory settings to real-world deployment. Meanwhile, by enabling robots to fluidly balance deliberate cognitive planning with rapid physical reflexes, our approach advances the development of reliable, general-purpose robotic assistants for complex, unstructured domains such as smart manufacturing, elder care, and domestic automation.

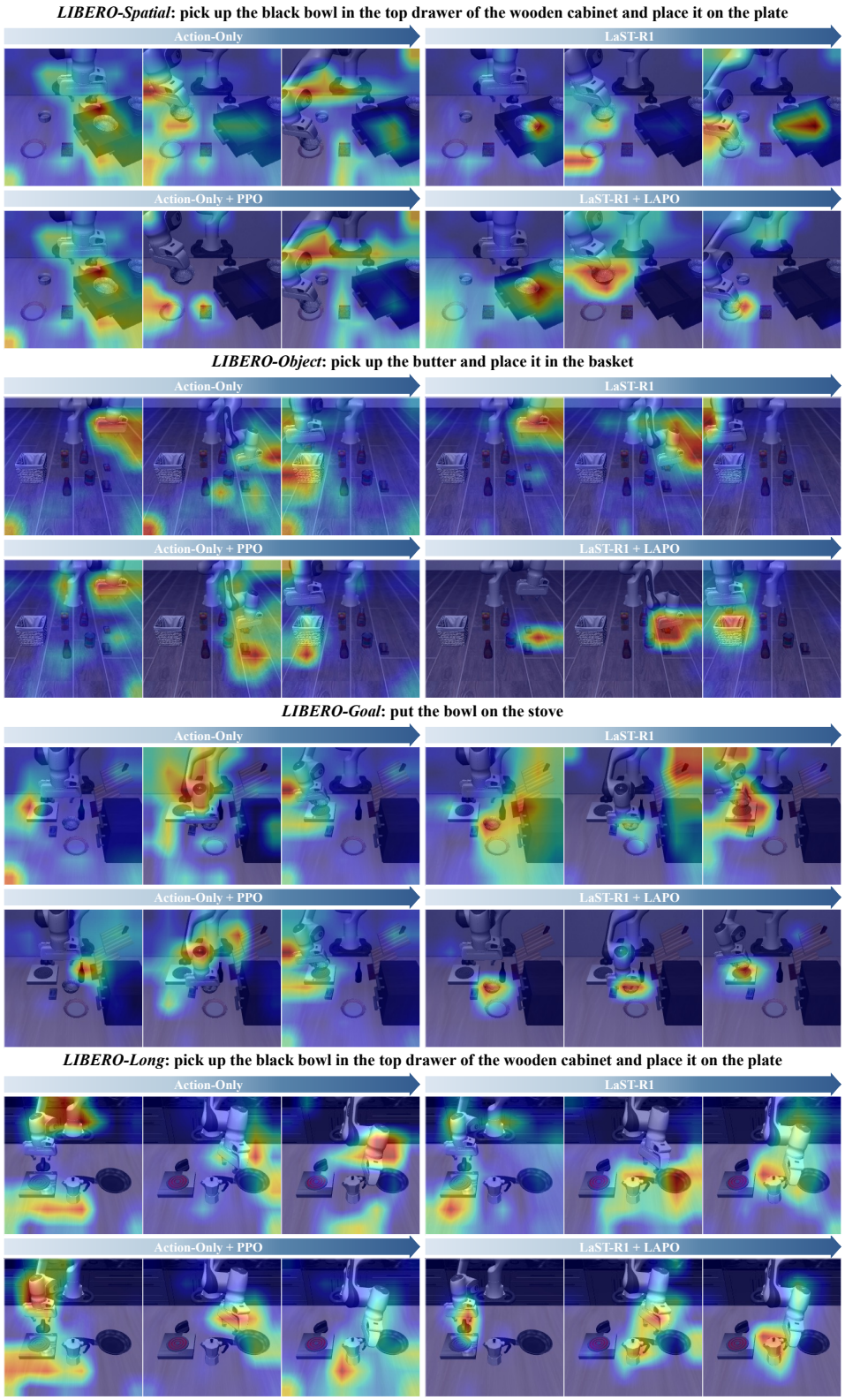


Figure 16: **Visualizations of Action-to-Vision Cross-Attention.** We compare the attention maps across four representative trajectories from four task suites in LIBERO.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We did.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We did.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A].

Justification: We do not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We did.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We did.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We did.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We did.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We did.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We did.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We did.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We do not pose such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We did.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We do not involve with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We do not involve with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

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