Fast-ThinkAct: Efficient Vision-Language-Action Reasoning via Verbalizable Latent Planning

Chi-Pin Huang[1], **Yunze Man**[2], **Zhiding Yu**, **Min-Hung Chen**, **Jan Kautz**, **Yu-Chiang Frank Wang**[1], **Fu-En Yang**
NVIDIA

Abstract

Vision-Language-Action (VLA) tasks require reasoning over complex visual scenes and executing adaptive actions in dynamic environments. While recent studies on reasoning VLAs show that explicit chain-of-thought (CoT) can improve generalization, they suffer from high inference latency due to lengthy reasoning traces. We propose Fast-ThinkAct, an efficient reasoning framework that achieves compact yet performant planning through verbalizable latent reasoning. Fast-ThinkAct learns to reason efficiently with latent CoTs by distilling from a teacher, driven by a preference-guided objective to align manipulation trajectories that transfers both linguistic and visual planning capabilities for embodied control. This enables reasoning-enhanced policy learning that effectively connects compact reasoning to action execution. Extensive experiments across diverse embodied manipulation and reasoning benchmarks demonstrate that Fast-ThinkAct achieves strong performance with up to 89.3% reduced inference latency over state-of-the-art reasoning VLAs, while maintaining effective long-horizon planning, few-shot adaptation, and failure recovery.

Links: Project Page

1. Introduction

Recent large vision-language models (VLMs) ????????? have achieved remarkable capabilities in visual-language understanding across diverse multimodal tasks. To extend these capabilities to embodied-centric tasks, recent works leverage large-scale robot demonstrations ? to develop Vision-Language-Action (VLA) foundation models ????????. These VLA tasks require agents to perceive complex visual scenes, reason over spatial and temporal contexts, and execute adaptive actions within dynamic environments, demanding robust long-horizon planning and contextual adaptation. However, as these VLA models primarily rely on supervised training from action data, they excel at basic skills (e.g., pick-and-place) but struggle to generalize beyond training distributions, such as long-horizon planning, self-correction from failures, and adaptation to novel scenarios, due to the impracticality of collecting exhaustive robot demonstrations.

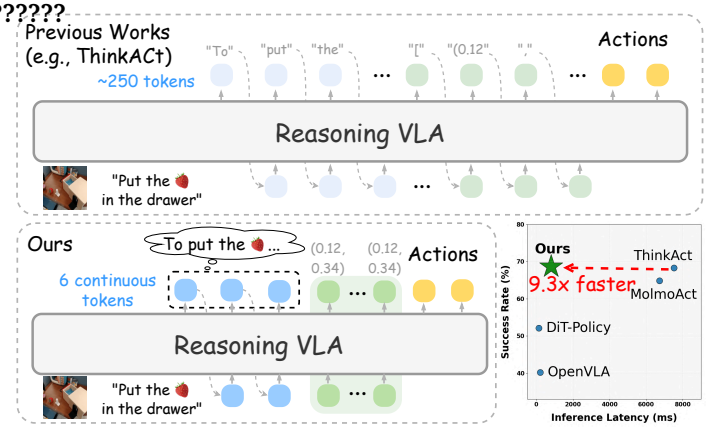

Figure 1: **Overview of Fast-ThinkAct.** Previous reasoning VLAs generate lengthy reasoning traces (∼250 tokens). Our approach learns compact continuous tokens (e.g., 6) (blue) and parallel spatial tokens (green) as internal reasoning. The bottom-right plot shows that we achieve 9.3× faster inference than ThinkAct-7B ?, while delivering improved performance on the SimplerEnv-Google benchmark.

Additional affiliations: [1] National Taiwan University. [2] University of Illinois Urbana-Champaign.

Reasoning VLAs **???????** address these limitations by incorporating intermediate thinking processes, improving generalization and task-solving capability. Supervised chain-of-thought (CoT) methods **????** address this by learning from intermediate reasoning annotations. These approaches can be categorized into textual reasoning methods that leverage off-the-shelf LLMs and VLMs to generate pseudo CoT labels **?**, and visual reasoning methods that generate structured visual reasoning representations such as sub-goal images, image depth, and 2D visual traces **??**. However, these supervised approaches require substantial reasoning annotations and remain limited by training data coverage. To address this, ThinkAct **?** employs RL-based reasoning **?** to generate long textual CoTs guided by action-aligned visual rewards. While these reasoning methods effectively improve task generalization and planning capabilities, they require generating lengthy chain-of-thought steps that introduce substantial reasoning latency, which hampers embodied applications with *real-time* requirements.

In embodied AI applications such as robotic manipulation and autonomous driving, agents must make rapid decisions at high frequencies (e.g., 1-15 Hz) **?**. However, generating lengthy reasoning traces can take several seconds per decision (e.g., 0.1 Hz) **??**, creating a critical bottleneck that limits real-time performance **??** and poses safety risks in time-critical scenarios **?**. To mitigate this efficiency bottleneck while preserving reasoning capabilities, very recent works **???** have explored approaches to reduce inference latency in embodied reasoning. For instance, ECoT-Lite **?** proposes reasoning dropout to accelerate inference, yet directly reducing textual reasoning length risks performance degradation due to critical information loss. How to preserve reasoning capability while enabling compact representations that properly capture essential spatial-temporal dynamics remains a crucial challenge for reasoning VLA models.

In this paper, we propose *Fast-ThinkAct*, an efficient embodied reasoning framework for Vision-Language-Action tasks that achieves compact yet expressive planning through verbalizable latent reasoning. As depicted in Figure 1, unlike prior reasoning VLAs that generate lengthy explicit textual CoT traces, we introduce reward-guided preference distillation with visual trajectory alignment to compress linguistic and visual planning into compact continuous latents that enable implicit internal reasoning. Our student VLM encodes reasoning into compact latents decodable by a verbalizer, enabling preference-based optimization that leverages RL-derived reward signals to distill high-quality reasoning patterns from a textual teacher VLM while suppressing low-quality ones. We further align trajectory latents between teacher and student to transfer visual planning capabilities essential for embodied control. Once trained, the student VLM enables reasoning-enhanced policy learning that bridges implicit multimodal planning with action execution, achieving significantly faster inference while outperforming existing reasoning VLAs.

Our contributions can be summarized as follows:

- We propose *Fast-ThinkAct*, an efficient reasoning framework that compresses reasoning into verbalizable latent thoughts while maintaining expressive planning abilities.
- We introduce preference-guided distillation with manipulation trajectory alignment that compresses linguistic and visual planning into compact continuous latents.
- We bridge high-level visual planning to low-level action execution through reasoning-enhanced policy learning guided by manipulation trajectory latents.
- We achieve up to 89.3% inference latency reduction over state-of-the-art reasoning VLAs while maintaining strong performance across diverse embodied benchmarks.

2. Related Works

2.1. Vision-Language-Action (VLA) Models

Foundation VLAs. Vision-Language-Action (VLA) models **???????????** have recently emerged as a promising paradigm for embodied AI by training vision-language backbones on large-scale robot demonstrations. Works such as OpenVLA **?** and π_0 **?** achieve language-conditioned manipulation through end-to-end policy learning, while Magma **?** co-trains on heterogeneous human and robot data. HAMSTER **?** and TraceVLA **?** further

leverage 2D visual trajectories to boost spatial-action connections. Despite success on routine manipulation, these imitation-based approaches struggle with long-horizon planning and generalization to novel scenarios due to limited training data coverage.

Reasoning VLAs. To overcome these limitations, recent works **?????????** integrate explicit reasoning mechanisms into VLA architectures. Supervised approaches **????** introduce intermediate reasoning through chain-of-thought annotations. Embodied CoT **?** and Hi-Robot **?** synthesize reasoning labels via pretrained foundation models. To perform vision-centric reasoning **??** beyond pure text, CoT-VLA **?** employs visual goal generation and MolmoAct **?** structures reasoning by spatial representations. Additionally, EO-1 **?** introduces interleaved vision-language-action pre-training to bridge reasoning and interaction. Recent works **??** alternatively leverage reinforcement fine-tuning to generate reasoning chains with designed rewards. Despite improved generalization, these reasoning VLAs suffer from high inference latency and inevitably introduce extraneous information that degrades action quality.

2.2. Efficient Reasoning

To address the inference latency of reasoning, recent LLM research explores various efficiency techniques **??????**. For example, RL-based approaches **????** introduce length penalties to encourage shorter reasoning chains, though such methods can suffer from training instability. Beyond length control, latent reasoning methods **?????** enable reasoning in continuous spaces, such as Coconut **?** using hidden states as continuous thoughts, CODI **?** distilling explicit CoT into continuous space via teacher-student alignment, and Soft Thinking **?** generating weighted concept tokens. However, these LLM techniques cannot directly transfer to VLA tasks due to the need for spatial-temporal understanding and bridging semantic reasoning with embodied control. Recently, ECoT-Lite **?** proposes reasoning dropout to accelerate embodied reasoning by skipping test-time reasoning traces. However, reasoning dropout can lead to inconsistent planning as it builds on supervised embodied CoT. Our proposed Fast-ThinkAct distills reasoning into compact latent representations that naturally encode multimodal information, enabling robust reasoning-enhanced policy learning.

3. Method

3.1. Problem Formulation

We first define the setting and notations. At each timestep t, given a language instruction l, the model observes a visual input o_t and generates an action chunk a_t, represented as a sequence of continuous robot control vectors (e.g., 7- or 14-DOF for single- or bimanual robots, respectively).

To address this problem, we propose Fast-ThinkAct, an efficient reasoning framework that bridges high-level planning with low-level action execution. Our approach employs a VLM $\mathcal{F}_\theta$ to perform reasoning in *continuous latent space*, integrated with an action model π_ϕ for executable action generation. Specifically, $\mathcal{F}_\theta$ processes observation-instruction pairs (o_t, l) through latent chain-of-thought (CoT) reasoning to produce a compact visual plan latent c_t that encapsulates the intended trajectory in visual space (Sec. 3.2). This c_t subsequently guides π_ϕ to predict executable actions a_t (Sec. 3.3). By distilling reasoning into a continuous latent space rather than discrete text, Fast-ThinkAct achieves significantly improved inference efficiency while enhancing action performance through better preservation of spatial and visual information.

3.2. Efficient Embodied Reasoning

To enable efficient embodied reasoning that meets the real-time requirements of embodied AI tasks, we aim to compress long textual CoTs into a compact set of continuous latent representations. However, compressing reasoning traces into latents is challenging, as there is no direct supervision signal in the latent space to guide what reasoning patterns should be encoded.

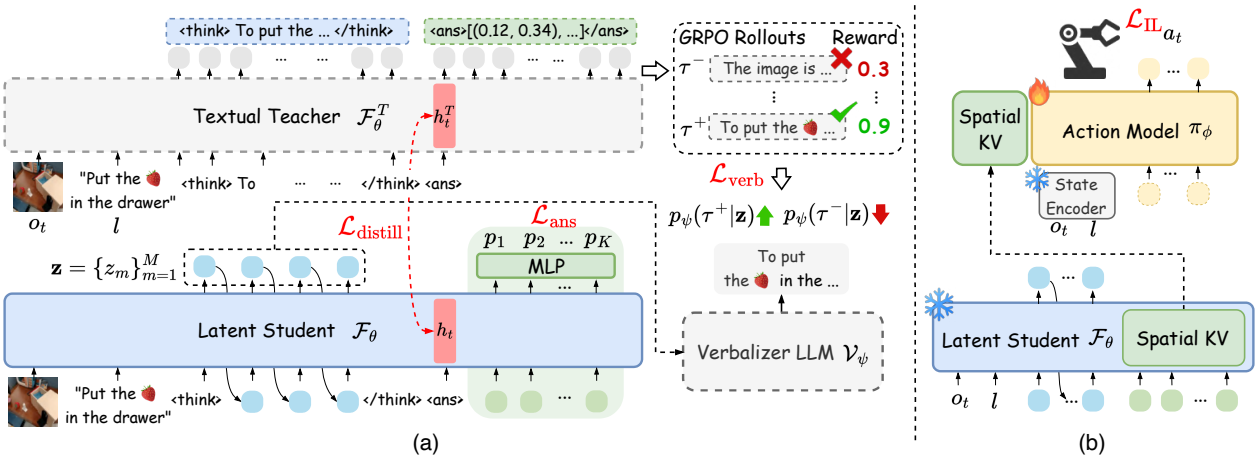

Figure 2: **Overview of Fast-ThinkAct.** (a) Given observation o_t and instruction l, the Textual Teacher VLM $\mathcal{F}_\theta^T$ generates explicit reasoning chains. The Latent Student VLM $\mathcal{F}_\theta$ distills these into compact latent tokens $\mathbf{z}$ guided by reward preferences. Verbalizer LLM $\mathcal{V}_\psi$ decodes latents to text for preference-based learning via $\mathcal{L}_{\text{verb}}$, while $\mathcal{L}_{\text{distill}}$ transfers visual planning capability from teacher, and spatial tokens enable parallel visual trajectory prediction via $\mathcal{L}_{\text{ans}}$, ensuring latents are verbalizable and grounded in visual planning. (b) Reasoning-Enhanced Policy Learning. The Action Model π_ϕ is trained with $\mathcal{L}_{\text{IL}}$ while freezing the latent student $\mathcal{F}_\theta$ and state encoder.

3.2.1. Verbalizable Latent CoT by Reward Preferences

To address this challenge, we propose to perform distillation in natural language space by introducing a verbalizer LLM that decodes latents into verbalizable reasoning. This approach grounds latent learning in an interpretable textual form, ensuring that the learned latents faithfully preserve the underlying reasoning structure. Since reasoning traces generated by the teacher model $\mathcal{F}_\theta^T$ exhibit varying quality, we adopt a preference-based learning framework that exploits reward signals from the teacher's GRPO training to guide the latent student $\mathcal{F}_\theta$ toward high-quality reasoning patterns while suppressing low-quality ones.

Specifically, we employ a teacher-student framework where a textual teacher model $\mathcal{F}_\theta^T$ first learns explicit reasoning through GRPO **?** training by maximizing:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\tau\sim\mathcal{F}_\theta^T}\Big[\min\big(r_\theta(\tau)A(\tau), \text{clip}(r_\theta(\tau), 1-\epsilon, 1+\epsilon)A(\tau)\big)\Big], \tag{1}$$

where τ denotes a reasoning trace and $r_\theta(\tau) = \frac{\mathcal{F}_\theta^T(\tau)}{\mathcal{F}_{\text{old}}^T(\tau)}$ is the probability ratio. The advantage function for group rewards $\{R_i\}_{i\in G(\tau)}$ is represented as:

$$A(\tau) = \frac{R_\tau - \text{mean}(\{R_i\}_{i\in G(\tau)})}{\text{std}(\{R_i\}_{i\in G(\tau)})}. \tag{2}$$

This training process produces textual CoTs with varying quality, where the advantage function $A(\tau)$ naturally serves as a quality indicator. To construct preference pairs for distillation, we select the highest and lowest advantage traces from each rollout group:

$$\tau^+ = \arg\max_{\tau\in G} A(\tau) \text{ and } \tau^- = \arg\min_{\tau\in G} A(\tau). \tag{3}$$

Instead of generating textual tokens, the student model $\mathcal{F}_\theta$ performs latent reasoning by autoregressively generating M continuous latent vectors $\mathbf{z} = \{z_m\}_{m=1}^M$ with $z_m \in \mathbb{R}^d$, where d is the hidden size. We then train

the verbalizer LLM $\mathcal{V}_\psi$ to decode these latents $\mathbf{z}$ into natural language. The training objective encourages the verbalizer to assign a higher likelihood to decoding latents into high-quality reasoning τ^+ than low-quality reasoning τ^-. Inspired by DPO **?**, we formulate this as an optimization guided by the reward preferences:

$$\mathcal{L}_{\text{verb}} = -\mathbb{E}\Big[\log \sigma\Big(\beta\big(\log \tfrac{p_\psi(\tau^+|\mathbf{z})}{p_{\text{ref}}(\tau^+)} - \log \tfrac{p_\psi(\tau^-|\mathbf{z})}{p_{\text{ref}}(\tau^-)}\big)\Big)\Big], \tag{4}$$

where p_{ref} is the reference model (i.e., $\mathcal{V}_\psi$ without latent conditioning), σ is the sigmoid function, and $\beta = 0.1$ controls preference strength. This encourages the student VLM $\mathcal{F}_\theta$ to encode latents that the verbalizer decodes into high-quality reasoning while suppressing low-quality patterns.

3.2.2. Action-Aligned Visual Plan Distillation

While the verbalizer loss (Eq. 4) enables the student $\mathcal{F}_\theta$ to capture high-level reasoning patterns, it does not explicitly ensure that latent representations encode the visual planning capability crucial for embodied control. To address this, we introduce action-aligned visual plan distillation to transfer the teacher $\mathcal{F}_\theta^T$'s spatial reasoning ability to the student $\mathcal{F}_\theta$.

We distill spatial reasoning from the teacher, which is trained with trajectory-level rewards (e.g., goal completion and trajectory alignment **?**) for grounded visual planning. We align the trajectory-level representations by minimizing the L2 distance between hidden states of the `<answer>` token that encodes the visual plan:

$$\mathcal{L}_{\text{distill}} = \|h_t^T - h_t\|_2^2, \tag{5}$$

where h_t^T and h_t are the hidden states from teacher (corresponding to τ^+) and student, respectively.

To enable efficient parallel trajectory prediction, unlike the textual teacher that autoregressively generates verbose text sequences of waypoints $\{p_k\}_{k=1}^K$ with $p_k \in [0,1]^2$ (tokenized into 60-70 tokens when $K = 5$), the student uses K learnable spatial tokens $\{\mathbf{s}_i\}_{i=1}^K$ appended to the reasoning latent sequence, with each output hidden state simultaneously projected to a waypoint via an MLP. The total objective for training $\mathcal{F}_\theta$ combines all three components:

$$\begin{aligned} \mathcal{L}_{\text{student}} &= \mathcal{L}_{\text{verb}} + \mathcal{L}_{\text{distill}} + \mathcal{L}_{\text{ans}}, \quad \text{where} \\ \mathcal{L}_{\text{ans}} &= \sum_{i=1}^{K} \|p_i - \hat{p}_i\|_2^2, \ \text{with } p_i = \text{MLP}(h'(\mathbf{s}_i)), \end{aligned} \tag{6}$$

where $h'(\mathbf{s}_i)$ denotes the output hidden state of the i-th spatial token and $\hat{p}_i$ are ground-truth waypoints. Through this unified framework, the student model $\mathcal{F}_\theta$ performs compact yet expressive latent reasoning and generates visual trajectory plans efficiently.

3.3. Reasoning-Enhanced Policy Learning

After the student VLM $\mathcal{F}_\theta$ performs compact latent reasoning and generates visual trajectory planning through spatial tokens, we leverage these representations to guide a diffusion Transformer-based action model π_ϕ (e.g., RDT **?**) for action prediction. To bridge the high-level visual planning with low-level action generation, we connect the visual latent planning c_t encoded in the key-value cache corresponding to the spatial tokens to the action model.

Specifically, we extract visual latent planning c_t from the KV cache of spatial tokens in earlier VLM layers (since $\mathcal{F}_\theta$ has more layers than π_ϕ) and concatenate with KV pairs from the action model's state encoder. The action model's cross-attention then attends to both the visual planning context and state observations. We post-train on action-annotated robot data by *freezing* $\mathcal{F}_\theta$ and the state encoder while updating only π_ϕ with the imitation learning objective:

$$\mathcal{L}_{\text{IL}}(\phi) = \ell\left(\pi_\phi(o_t, l, c_t), \hat{a}_t\right), \tag{7}$$

where ℓ denotes the denoising objective for diffusion policy and $\hat{a}_t$ is the ground-truth action. Through this post-

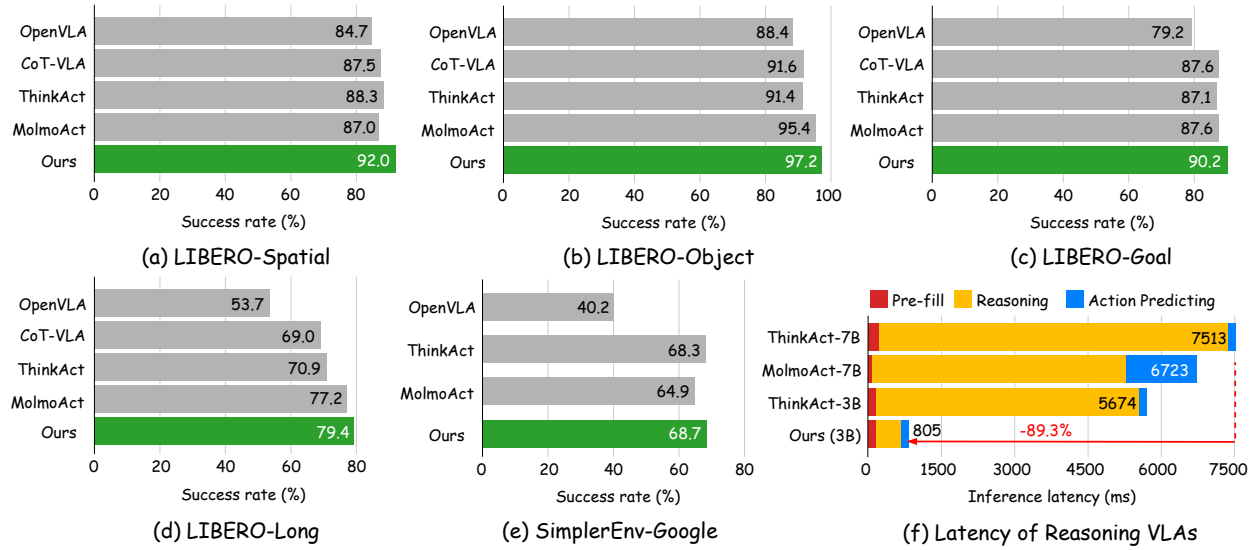

Figure 3: **Evaluation of robot manipulation and reasoning efficiency.** (a)-(e) Success rates on LIBERO **?** and SimplerEnv **?** benchmarks compared with state-of-the-art 7B reasoning VLAs. (f) Latency comparison across 3B and 7B reasoning VLAs. Our approach achieves up to 89.3% inference latency reduction while maintaining superior task success rates.

training, the action model effectively translates visual planning from compact latent reasoning into low-level robot actions.

3.4. Learning Strategy and Inference

Training Strategy. We initialize both teacher $\mathcal{F}_\theta^T$ and student $\mathcal{F}_\theta$ from the same checkpoint obtained through SFT and CoT-SFT on a pre-trained VLM. The teacher is trained with GRPO using action-aligned rewards **?**, while the student is trained with $\mathcal{L}_{\text{student}}$ to compress reasoning into compact latents. We then connect the trained $\mathcal{F}_\theta$ with action model π_ϕ (initialized from **?**) by freezing $\mathcal{F}_\theta$ and the state encoder while updating the latent projector and π_ϕ with $\mathcal{L}_{\text{IL}}$ on large-scale robotic data. For target environment adaptation (e.g., LIBERO **?**, RoboTwin2.0 **?**), we fine-tune on environment-specific demonstrations.

Inference. The $\mathcal{F}_\theta$ processes (o_t, l) by compact latent reasoning, generating visual trajectories via K spatial tokens. The visual latent planning c_t, extracted from the spatial tokens' KV cache, conditions π_ϕ to predict actions a_t. Inference requires only $\mathcal{F}_\theta$ and π_ϕ; the verbalizer $\mathcal{V}_\psi$ is used solely during training and optionally for interpretability.

4. Experiment

4.1. Experimental Setup

Implementation Details.

We use Qwen2.5-VL 3B **?** as the VLM backbone. The SFT stage runs for 1 epoch with batch size 64 and learning rate $1e{-}5$, followed by CoT-SFT for 15K iterations with the same hyperparameters. For teacher-student training, both $\mathcal{F}_\theta^T$ and $\mathcal{F}_\theta$ are initialized from the CoT-SFT checkpoint and trained for 4,500 iterations with batch size 128 and learning rate $1e{-}6$. The teacher is optimized with GRPO **?** using action-aligned visual rewards **?** and QA-style rewards (detailed in supplementary material). For the first 3,000 iterations of the student training, we train the verbalizer $\mathcal{V}_\psi$ with standard language modeling loss, then switch to $\mathcal{L}_{\text{verb}}$ for the remaining 1,500 iterations. For reasoning-enhanced policy learning, we initialize π_ϕ from DiT-Policy **?** pre-trained on OXE **?** for SimplerEnv, and from RDT **?** for LIBERO and RoboTwin2.0. A linear projection adapts the VLM's KV cache to the action model dimension (1,024 for DiT-Policy and 2,048 for RDT). Training runs for 20K iterations with

Table 1: **Quantitative evaluation on RoboTwin2.0 ?.** E and H denote easy and hard settings (without/with domain randomization). Background colors indicate task length based on expert demonstrations: short (80-100), medium (110-220), long (270-470) steps.

Model	click alarm		click bell		turn switch		adjust bottle		beat block		handover mic		handover block		hanging mug		stack blocks two		stack bowls three		Average	
	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H	E	H
DP ?	61	5	54	0	36	1	97	0	42	0	53	0	10	0	8	0	7	0	63	0	43.1	0.6
ACT ?	32	4	58	3	5	2	97	23	56	3	85	0	42	0	7	0	25	0	48	0	45.5	3.5
π_0 ?	63	11	44	3	27	23	90	56	43	21	98	13	45	8	11	3	42	1	66	24	52.9	16.3
RDT ?	61	12	80	9	35	15	81	75	77	37	90	31	45	14	23	16	21	2	51	17	56.4	22.8
ThinkAct ?	64	13	84	11	40	19	94	70	79	33	92	40	56	15	31	18	30	5	54	23	62.4	24.7
Fast-ThinkAct	70	17	82	12	37	21	92	72	82	33	99	42	65	15	30	22	45	5	55	25	**65.7**	**26.4**

Table 2: **Quantitative evaluation on EgoPlan-Bench2 ?, RoboVQA ?, and OpenEQA ? benchmarks for embodied reasoning.**

Method	**EgoPlan-Bench2**					**RoboVQA**					**OpenEQA**	**Overall**
	Daily.	Work.	Rec.	Hobbies	Avg.	B-1	B-2	B-3	B-4	B-Avg.	Score	Avg.
GPT-4V ?	36.7	27.7	33.9	32.5	32.6	32.2	26.5	24.7	23.9	26.8	49.6	36.4
Gemini-2.5-Flash ?	44.2	42.3	43.2	39.1	42.4	39.1	31.6	22.9	22.1	28.9	45.3	38.9
InternVL2.5-2B ?	30.9	27.8	28.6	33.1	30.1	36.6	33.7	31.0	29.4	32.7	47.1	36.6
InternVL3-2B ?	36.9	29.9	35.6	31.5	33.4	34.4	33.9	33.5	33.3	33.8	48.8	38.7
NVILA-2B ?	34.6	26.7	33.3	31.6	31.4	38.7	34.3	31.1	29.2	33.3	47.0	37.2
Qwen2.5-VL-3B ?	29.0	27.0	30.2	28.9	28.5	42.5	36.3	28.7	31.8	34.8	43.4	35.6
Magma-8B ?	32.1	25.7	34.4	29.3	29.8	38.6	31.5	28.1	26.7	31.2	49.1	36.7
RoboBrain2.0-3B ?	45.3	37.6	45.9	39.7	41.8	54.4	47.7	43.1	41.0	46.5	50.1	46.1
ThinkAct-3B ?	46.6	41.4	45.9	42.5	44.0	62.4	57.3	52.0	49.6	55.3	48.9	49.4
Fast-ThinkAct-3B	**50.3**	**44.3**	**46.4**	**43.2**	**46.4**	**70.1**	**63.0**	**57.2**	**53.0**	**60.8**	**51.2**	**52.8**

batch size 256 and learning rate $1e{-}4$. All diffusion hyperparameters follow those of the respective action models. All experiments are conducted on 16 NVIDIA A100 GPUs with 80 GB memory.

Training Datasets and Evaluation Benchmarks.
For reasoning VLM training, we utilize single-arm visual trajectories labeled by **?** and dual-arm visual trajectories from the AIST dataset **?**, along with QA tasks from PixMo **?**, RoboFAC **?**, RoboVQA **?**, ShareRobot **?**, EgoPlan **?**, and Video-R1 **?**. For reasoning-enhanced policy learning, we use action data from the OXE dataset **?** (following OpenVLA **?**) when training with DiT-Policy, and augment with bimanual data from the static Aloha dataset **??** when training with RDT.

We evaluate Fast-ThinkAct on four embodied reasoning benchmarks and three robot manipulation benchmarks. For embodied reasoning, we use EgoPlan-Bench2 **?** (accuracy on multiple-choice questions), RoboVQA **?** (BLEU score **?**), OpenEQA **?**, and RoboFAC **?** (both using LLM-based scoring). Notably, RoboVQA and RoboFAC contain videos captured from real robots. For robot manipulation, we evaluate on SimplerEnv **?**, which demonstrates strong correlation with real-world performance, LIBERO **?** covering diverse manipulation tasks including long-horizon scenarios, and RoboTwin2.0 **?** for complex bimanual manipulation. All robot manipulation tasks use task success rate as the metric. Additional details are provided in the supplementary material.

4.2. Quantitative Evaluation

Robot Manipulation.
We evaluate Fast-ThinkAct on robotic manipulation using LIBERO **?** and SimplerEnv **?** benchmarks. LIBERO covers diverse subtasks, including Spatial, Object, Goal, and Long, while SimplerEnv provides a simulated benchmark with strong real-world correlation, featuring variations in lighting, object appearance, and camera viewpoints. As shown in Fig. 3(a)-(e), Fast-ThinkAct consistently outperforms all baselines, achieving the highest

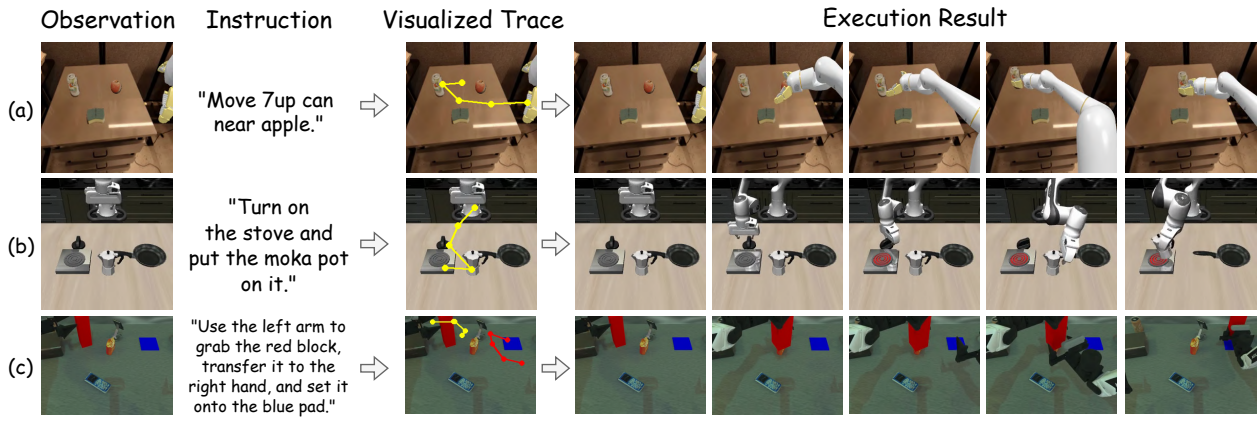

Figure 4: **Visualization of predicted visual trajectories and action execution results on long-horizon tasks.** Examples from (a) SimplerEnv-Google, (b) LIBERO-Long, and (c) RoboTwin2.0-Hard with long (278) steps. Yellow traces indicate single-arm/left gripper trajectories; red traces indicate right gripper trajectories for bimanual tasks.

success rates across all LIBERO subtasks and SimplerEnv-Google. This includes substantial improvements over foundation VLAs such as OpenVLA **?**, and reasoning VLAs including CoT-VLA **?**, ThinkAct **?**, and MolmoAct **?**. Moreover, as shown in Fig. 3(f), our compact latent reasoning achieves 89.3% and 88.0% latency reduction compared to ThinkAct-7B **?** and MolmoAct-7B **?** respectively, and 7× faster inference than ThinkAct-3B, demonstrating substantial efficiency gains without sacrificing performance.

To further validate Fast-ThinkAct on more complex scenarios, we evaluate on RoboTwin2.0 **?**, a challenging bimanual manipulation benchmark requiring long-horizon planning. As shown in Tab. 1, Fast-ThinkAct significantly outperforms previous VLAs including DP **?**, ACT **?**, π_0 **?**, RDT **?**, and ThinkAct **?** across both easy and hard settings. Compared to RDT, Fast-ThinkAct achieves 9.3% and 3.6% higher success rates on easy and hard settings, respectively. Against the reasoning VLA ThinkAct, it improves success by 3.3% and 1.7% while maintaining substantially higher efficiency, as shown in Fig. 3(f). These results demonstrate that our compact reasoning design enables both superior accuracy and computational efficiency on complex bimanual manipulation tasks.

Embodied Reasoning.
In Tab. 2, we evaluate the reasoning capabilities of Fast-ThinkAct in embodied scenarios across three benchmarks: EgoPlan-Bench2 **?**, RoboVQA **?**, and OpenEQA **?**. These benchmarks assess multi-step planning in egocentric everyday scenarios, long-horizon reasoning for robotic manipulation tasks, and zero-shot understanding of embodied scenes in diverse environments, respectively. We observed that, Fast-ThinkAct surpasses all comparison methods, including two proprietary models (i.e., GPT-4V **?** and Gemini-2.5-Flash **?**), exceeding the runner-up by 2.4% on EgoPlan-Bench2, 5.5 BLEU score on RoboVQA, and 1.1 points on OpenEQA. These results demonstrate that Fast-ThinkAct effectively handles complex planning sequences and extended reasoning horizons while generalizing to novel environments, showcasing robust capabilities for scene comprehension and multi-step task execution in embodied AI applications.

4.3. Analysis of Fast-ThinkAct

Reasoning Enables Long-Horizon Planning.
We analyze Fast-ThinkAct's capability on long-horizon tasks in Tab. 1 and Fig. 4. We focus on long-horizon tasks (average length exceeding 270 steps) in RoboTwin2.0 **?** that require multi-step reasoning and extended planning horizons. As shown in Tab. 1, Fast-ThinkAct achieves average scores of 48.8 and 16.8 on easy and hard settings of long-horizon tasks, respectively, surpassing RDT (35.0/12.3) and ThinkAct (42.8/15.3). Fig. 4 visualizes predicted 2D visual traces and execution results on representative tasks from SimplerEnv-Google **?**, LIBERO-

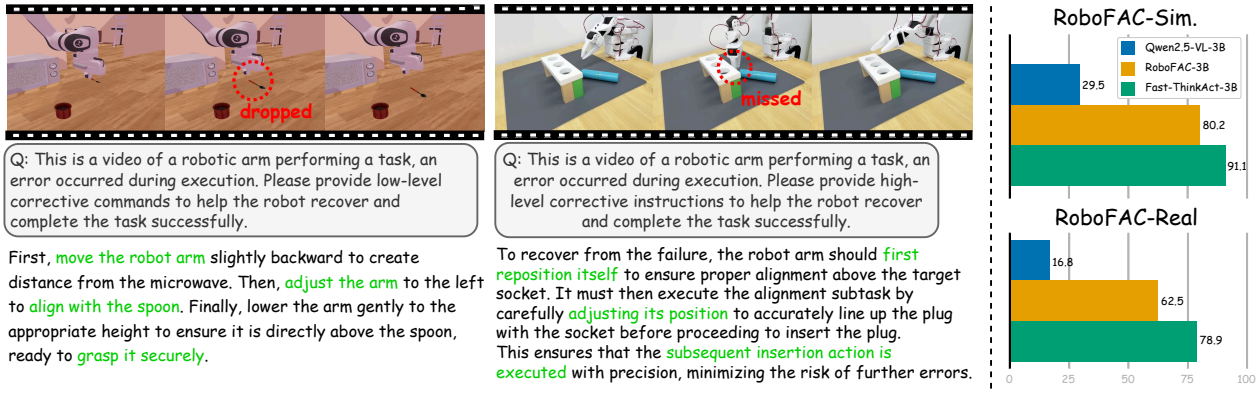

Figure 5: **Failure recovery capability on RoboFAC ?.** Left: Qualitative examples (from both simulation and real robot) of corrective guidance for manipulation errors. Right: Quantitative evaluation on simulation (RoboFAC-Sim) and real-robot (RoboFAC-Real) settings.

Long **?**, and RoboTwin2.0 **?**. For example, the LIBERO-Long task requires sequentially turning on the stove and placing a moka pot on it, while the RoboTwin2.0 handover task requires bimanual coordination to transfer a block between grippers. The visual traces successfully predict feasible solution paths, with their corresponding representations serving as visual planning guidance for successful execution. These results demonstrate that our compact latent reasoning effectively supports long-horizon planning in complex manipulation scenarios.

Reasoning Enables Failure Recovery.
A key advantage of reasoning-based VLAs **??** is their ability to identify runtime failures and provide corrective guidance for recovery. To evaluate this capability, we conduct experiments on RoboFAC **?**, a benchmark specifically designed to assess failure identification and correction in embodied VLMs. As shown in Fig. 5, Fast-ThinkAct substantially outperforms the second-best baseline RoboFAC-3B **?** by 10.9 points on the simulation split and 16.4 points on the real-world split. The qualitative examples demonstrate Fast-ThinkAct's ability to reason over manipulation videos, identify failures, and propose recovery steps. For instance, in the right example where the target object drops mid-execution, Fast-ThinkAct generates a concrete recovery plan: first moving the arm backward to create space, then adjusting laterally to align with the target object, and finally lowering to the appropriate height for a secure grasp. These results demonstrate that our latent reasoning supports both fast task execution and crucial failure analysis capabilities essential for robust robotic manipulation.

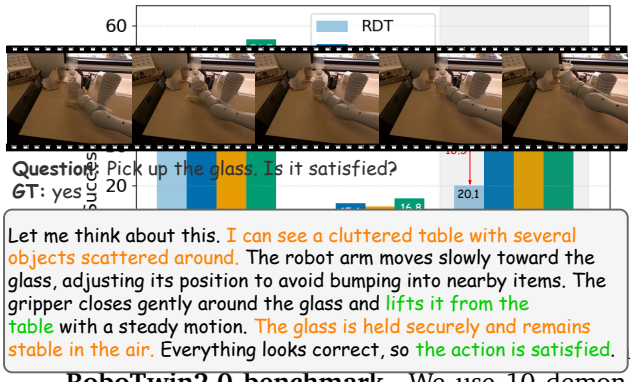

Figure 7: **Reasoning trace comparison on RoboVQA.** (a) Teacher's textual reasoning. (b) Student's verbalized latent reasoning. Green: relevant content; orange: less relevant content.

Reasoning Enables Few-Shot Adaptation.
To assess how reasoning capability improves few-shot adaptation, we conduct few-shot experiments on the RoboTwin2.0 benchmark **?**, fine-tuning models using only 10 demonstrations per task. As illustrated in Fig. 6, Fast-ThinkAct significantly enhances our adopted action model RDT **?** and outperforms the state-of-the-art VLAs, including π_0 **?** and ThinkAct **?** on both medium and long-horizon tasks. Notably, our method achieves

these gains while operating with significantly lower reasoning latency compared to ThinkAct, highlighting the advantage of efficient yet effective reasoning for few-shot action adaptation in complex robot manipulation scenarios.

Visualization of Verbalizable Latent Reasoning.
In Fig. 7, we compare the teacher's textual reasoning with the student's verbalized latent reasoning on RoboVQA. While both capture task-relevant information (green), the teacher generates verbose outputs with less directly relevant content (orange), whereas our student produces more concise and focused responses when verbalized. This demonstrates that our preference-guided distillation not only reduces computational cost but also distills concise reasoning patterns while filtering out redundant information.

Table 3: **Ablation study of training objectives and learning stages.** Note that Fast-ThinkAct w/o $\mathcal{L}_{\text{verb}}, \mathcal{L}_{\text{distill}}$ denotes the student VLM $\mathcal{F}_\theta$ trained without the corresponding loss components.

Method	EgoPlan	RoboVQA	OpenEQA	Average
Fast-ThinkAct	**46.4**	**60.8**	**51.2**	**52.8**
w/o $\mathcal{L}_{\text{verb}}$	42.1	53.8	49.5	48.5
w/o $\mathcal{L}_{\text{verb}}, \mathcal{L}_{\text{distill}}$	41.6	52.7	48.9	47.7
Textual Teacher $\mathcal{F}_\theta^T$	41.7	58.2	49.4	49.8
SFT + CoT-SFT	40.0	46.1	48.8	45.0
SFT only	40.5	53.6	45.3	46.5

Ablation Study.
In Tab. 3, we ablate training stages and loss components. Starting from the full Fast-ThinkAct, removing $\mathcal{L}_{\text{verb}}$ causes performance drops as latent CoTs lack preference-based guidance to align with high-quality reasoning and suppress low-quality patterns. Further removing $\mathcal{L}_{\text{distill}}$ leads to additional decline, indicating that aligning trajectory-level representations is crucial for transferring visual planning capabilities. Comparing training strategies, CoT-SFT underperforms SFT on EgoPlan-Bench2 and RoboVQA but improves on OpenEQA, suggesting naïve chain-of-thought supervision benefits open-ended QA but introduces verbosity that hinders structured reasoning tasks. Our preference-guided approach distills high-quality reasoning while maintaining efficiency. This validates the necessity of our proposed distillation framework and visual trajectory alignment. We provide additional ablation studies in the supplementary material.

5. Conclusion

We presented *Fast-ThinkAct*, an efficient reasoning framework for vision-language-action tasks that achieves compact yet expressive planning through verbalizable latent reasoning. By distilling lengthy textual reasoning into compact latent representations via preference-guided distillation and visual trajectory alignment, our approach bridges high-level embodied reasoning with low-level action execution through reasoning-enhanced policy learning. Exten-

sive experiments across diverse robotic manipulation and embodied reasoning benchmarks demonstrate that Fast-ThinkAct achieves strong performance with significantly reduced inference latency while enabling effective long-horizon planning, few-shot adaptation, and failure recovery capabilities.

Limitations and Future Works. As our verbalizer $\mathcal{V}_\psi$ is built upon a pre-trained LLM, it inevitably inherits language model limitations, including hallucination, occasionally producing plausible but inaccurate descriptions. However, this does not affect action execution during inference, as the verbalizer serves only for interpretability while action prediction uses the grounded latent representations from visual plan distillation. To further improve the faithfulness of verbalized reasoning, we can consider incorporating grounding-aware objectives or hallucination suppression techniques in future work.

A. Additional Experimental Setup

A.1. Algorithm

Algorithm 1: Training Fast-ThinkAct (Sec. 3.2)

Input: CoT-SFT checkpoint $\mathcal{F}_{\theta_0}$, training data $\mathcal{D}$, rollout size N, latent reasoning steps M, number of waypoints K, total iterations T_{total}

Output: Trained student model $\mathcal{F}_\theta$

`// Initialize models`

$\mathcal{F}_\theta^T \leftarrow \mathcal{F}_{\theta_0}$, $\mathcal{F}_\theta \leftarrow \mathcal{F}_{\theta_0}$;

Initialize verbalizer $\mathcal{V}_\psi$ from pre-trained LLM;

$t \leftarrow 0$;

while $t < T_{total}$ **do**

  Sample batch $(o, l, \hat{p})$ from $\mathcal{D}$;

  `// Suppose bs=1 for simplicity`

  `// Teacher GRPO training`

  Generate N rollouts $\{\tau_i\}_{i=1}^N$ from $\mathcal{F}_\theta^T(o, l)$;

  Compute trajectory rewards $\{r_i\}_{i=1}^N$;

  Compute group-wise advantages $\{A_i\}_{i=1}^N$;

  Update $\mathcal{F}_\theta^T$ with $\mathcal{J}_{\text{GRPO}}$ (Eq. 1);

  $\tau^+ \leftarrow \arg\max_i A_i$, $\tau^- \leftarrow \arg\min_i A_i$ (Eq. 3) ; `// For student distillation`

  $h_t^T \leftarrow$ hidden state of τ^+ at `<answer>` token from $\mathcal{F}_\theta^T$; `// For distillation loss`

  `// Student latent distillation`

  $\mathbf{z} = \{z_m\}_{m=1}^M \leftarrow \mathcal{F}_\theta(o, l)$; `// Perform auto-regressive latent reasoning`

  Compute $\mathcal{L}_{\text{verb}}$ with $\mathbf{z}$, $\mathcal{V}_\psi$, τ^+, τ^- (Eq. 4);

  Forward K spatial tokens from $\mathcal{F}_\theta(o, l, \mathbf{z})$ to obtain h_t and $\{h'(\mathbf{s}_i)\}_{i=1}^K$;

  Compute $\mathcal{L}_{\text{distill}}$ with h_t^T, h_t (Eq. 5);

  Compute $\mathcal{L}_{\text{ans}}$ with $\{h'(\mathbf{s}_i)\}_{i=1}^K$, $\hat{p}$ (Eq. 6);

  Update $\mathcal{F}_\theta$ with $\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{verb}} + \mathcal{L}_{\text{distill}} + \mathcal{L}_{\text{ans}}$;

  $t \leftarrow t + 1$;

end

return $\mathcal{F}_\theta$;

Algorithm 1 presents the complete training procedure corresponding to Sec. 3.2. It shows how we jointly optimize the teacher model with GRPO and distill its reasoning into the student's compact latent representations.

A.2. Implementation Details

Our implementation follows the setup described in Sec. 4.1 of the main paper. Here we provide additional details. The verbalizer $\mathcal{V}_\psi$ is initialized from a small LLM, Qwen3-0.6B, with cross-attention layers inserted at each layer to condition on latent CoTs $\mathbf{z}$. For the student model training, in the first 3,000 iterations, we replace verbalization loss $\mathcal{L}_{\text{verb}}$ with language modeling loss using τ^+ as ground truth to warm up $\mathcal{V}_\psi$'s alignment with the latent representations $\mathbf{z}$. We then freeze $\mathcal{V}_\psi$ and use the $\mathcal{L}_{\text{verb}}$ for the remaining 1,500 iterations. The student $\mathcal{F}_\theta$ is optimized throughout both phases. For waypoint prediction in Eq. 6, each $p_i \in \mathbb{R}^6$ encodes coordinates in the format $[x_{\text{single}}, y_{\text{single}}, x_{\text{left}}, y_{\text{left}}, x_{\text{right}}, y_{\text{right}}]$, where the first two dimensions are for single-arm and the last four are for bimanual robot. For ground-truth $\hat{p}i$, we fill the corresponding dimensions based on robot type and mask out the unused dimensions when computing $\mathcal{L}_{\text{ans}}$. For GRPO training, we follow the configuration of ThinkAct **?**, using rollout size $N = 5$. Following **?**, we set the number of waypoints in trajectory to $K = 5$. We use $M = 6$ latent reasoning tokens, with ablation study provided in Fig. 8.

During reasoning-enhanced policy learning, for SimplerEnv **?** evaluation, to ensure fair comparison with previous works **??**, we initialize π_ϕ from DiT-Policy **?** pre-trained on the same OXE dataset **??** and conduct reasoning-enhanced policy learning (Sec. 3.3) using the same OXE data. For LIBERO **?** and RoboTwin2.0 **?** evaluations, we initialize π_ϕ from RDT **?**, which has demonstrated strong performance on RoboTwin2.0, and conduct policy learning using OXE **?** and static ALOHA datasets **??**. Our method further enhances RDT's manipulation capabilities on both benchmarks. The use of different action models also demonstrates that our approach is agnostic to the underlying action model choice.

A.3. Training Data Details

A.3.1. Dataset Sources

2D Visual Trace of Manipulation Tasks.
For single-arm manipulation, we utilize 2D visual trajectories labeled by MolmoAct **?** from the Open X-Embodiment (OXE) dataset **?**, comprising approximately 1.3M trajectories. For bimanual manipulation, we extract dual-arm visual trajectories from the AIST dataset **?**, resulting in approximately 92K trajectory samples. Specifically, we first use Molmo-72B **?** to detect left and right gripper positions (following **?**) in the first frame, then apply CoTracker3 **?** to track and parse the manipulation trajectories throughout the video sequences.

RoboFAC ?.
RoboFAC is a robotic failure analysis dataset containing 9,440 erroneous manipulation trajectories across 16 tasks in both simulated and real-world environments. We utilize the training set with 64K QA pairs covering various failure types for developing failure identification and correction planning capabilities.

RoboVQA ?.
RoboVQA contains robot manipulation videos with QA tasks covering task understanding. The dataset includes approximately 5K long-horizon and 92K medium-horizon video sequences from diverse robotic platforms, resulting in total 798K QA pairs. Videos are annotated with multiple questions probing spatial reasoning, action prediction, and task comprehension.

ShareRobot ?.
ShareRobot is a large-scale dataset collected by RoboBrain **?**, containing over 1M QA pairs covering task planning, object affordances, and manipulation strategies across diverse robot embodiments and scenes. The dataset features fine-grained annotations linking task descriptions to frame-level execution details, facilitating learning of transferable manipulation knowledge.

EgoPlan-Bench ?.
EgoPlan-Bench features egocentric videos of daily activities annotated with task planning information including goals, execution history, and current states. The dataset contains approximately 53K video-text pairs for training long-horizon planning and progress tracking capabilities from egocentric view.

Video-R1-CoT ?.
Video-R1 comprises 165K video question-answer pairs with chain-of-thought reasoning annotations generated by large-scale vision-language models. The dataset covers diverse reasoning domains including mathematical logic, spatial understanding, OCR, and visual analytics. All samples are quality-filtered to ensure annotation consistency and correctness.

PixMo ?.
PixMo is a general-purpose vision-language dataset with diverse image captions and question-answer pairs. Following MolmoAct **?**, we incorporate PixMo dataset to preserve general visual understanding and prevent catastrophic forgetting when training on embodied dataset. Specifically, we use approximately 726K samples from the `ask_model_anything`, `cap`, and `cap-qa` splits.

Table 4: **Quantitative results with larger model size (7B or 8B) on embodied reasoning benchmarks.**

Method	EgoPlan-Bench2					RoboVQA					OpenEQA	Overall
	Daily.	Work.	Rec.	Hobbies	Avg.	B-1	B-2	B-3	B-4	B-Avg.	Score	Avg.
InternVL2.5-8B ?	36.2	28.7	34.4	35.4	33.5	40.5	33.3	29.6	27.5	32.7	54.4	40.2
InternVL3-8B ?	38.5	32.9	36.1	37.2	36.2	44.3	36.5	31.6	28.9	35.3	55.5	42.3
NVILA-8B ?	35.8	28.7	37.2	35.4	33.7	42.7	39.7	37.6	36.1	39.0	54.0	42.2
Qwen2.5-VL-7B ?	31.4	26.7	29.5	28.6	29.1	47.8	41.2	36.2	33.7	39.7	50.8	39.9
Magma-8B ?	32.1	25.7	34.4	29.3	29.8	38.6	31.5	28.1	26.7	31.2	49.1	36.7
RoboBrain2.0-7B ?	39.4	27.0	33.9	32.2	33.2	44.9	38.2	34.7	33.5	37.8	51.1	40.7
ThinkAct-7B ?	50.1	49.8	44.8	45.2	48.2	69.1	61.8	56.0	52.4	59.8	56.2	54.7
Fast-ThinkAct-7B	51.3	47.3	41.5	45.9	47.5	70.4	63.3	57.3	53.2	61.1	59.0	**55.9**

Table 5: **Results on LIBERO and SimplerEnv benchmarks with additional ThinkAct-3B comparison.**

Method	LIBERO	SimplerEnv-Google	Latency (↓)
OpenVLA-7B ?	76.5	40.2	N/A
CoT-VLA-7B ?	83.9	N/A	N/A
ThinkAct-7B ?	84.4	68.3	7513
MolmoAct-7B ?	86.8	64.9	6723
ThinkAct-3B ?	83.1	64.7	5674
Fast-ThinkAct-3B	**89.7**	**68.7**	**805 (↓7.0×)**

A.3.2. Data Processing and Formatting

Supervised Fine-Tuning (SFT).
To enhance foundational embodied knowledge, we perform supervised fine-tuning on approximately 4M samples combining 2D visual trajectories from MolmoAct ? and AIST ?, along with QA data from PixMo ?, RoboFAC ?, RoboVQA ?, ShareRobot ?, and EgoPlan ?. This stage enables the model to acquire basic visual understanding, task comprehension, and manipulation knowledge across diverse embodiments and scenarios.

Chain-of-Thought SFT (CoT-SFT).
To develop reasoning capabilities while preserving embodied understanding, we sample 5% from the SFT data (approximately 200K samples) and augment with 165K samples from Video-R1-CoT ?. For data with CoT annotations, we format prompts to elicit structured reasoning enclosed in `<think>` tags followed by answers in `<answer>` tags; for data without CoT annotations, we prompt for direct answers only. This enables the model to learn reasoning capabilities from CoT-annotated data and generalize them to embodied tasks.

Teacher-Student Training.
Building upon the CoT-SFT checkpoint, we curate a balanced training set by sampling approximately 5,000 instances from each dataset and data type, totaling nearly 50K samples. We adopt the prompt formatting strategy from CoT-SFT for both teacher GRPO training and student latent distillation. We train both the teacher with GRPO and the student with latent distillation (as detailed in Sec. 3.2) on this data, efficiently transferring high-quality reasoning patterns into compact latent representations.

A.4. Evaluation Setup

A.4.1. Embodied Reasoning Benchmarks

We evaluate on three benchmarks assessing different aspects of embodied reasoning. EgoPlan-Bench2 ? tests egocentric task planning across 24 daily-life scenarios with 1,321 multiple-choice questions, measuring accuracy in predicting next steps given task goals and progress history. RoboVQA ? evaluates visual reasoning in manipulation contexts through 1,893 free-form QA pairs from robot and human demonstrations, assessed via BLEU score. OpenEQA ? assesses spatial and functional understanding through 1,600+ questions spanning

Table 6: **Comparison with efficient textual reasoning methods.**

Method	EgoPlan-Bench2	RoboVQA	OpenEQA	Average
Textual Teacher $\mathcal{F}_\theta^T$	41.7	58.2	49.4	49.8
$\mathcal{F}_\theta^T$ Inference w/o thinking	42.7	55.0	41.7	46.5
$\mathcal{F}_\theta^T$ Inference w/ 6 textual tokens	39.3	53.0	46.5	46.3
$\mathcal{F}_\theta^T$ w/ RL Length-Penalty ?	41.2	57.5	44.7	47.8
Fast-ThinkAct-3B	**46.4**	**60.8**	**52.8**	**53.3**

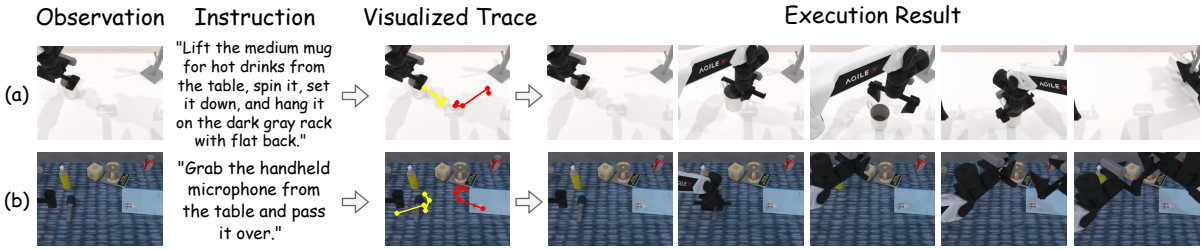

Figure 8: **Visualization of predicted visual trajectories and action execution results on RoboTwin2.0.** Yellow traces indicate left gripper trajectories; red traces indicate right gripper trajectories for bimanual tasks.

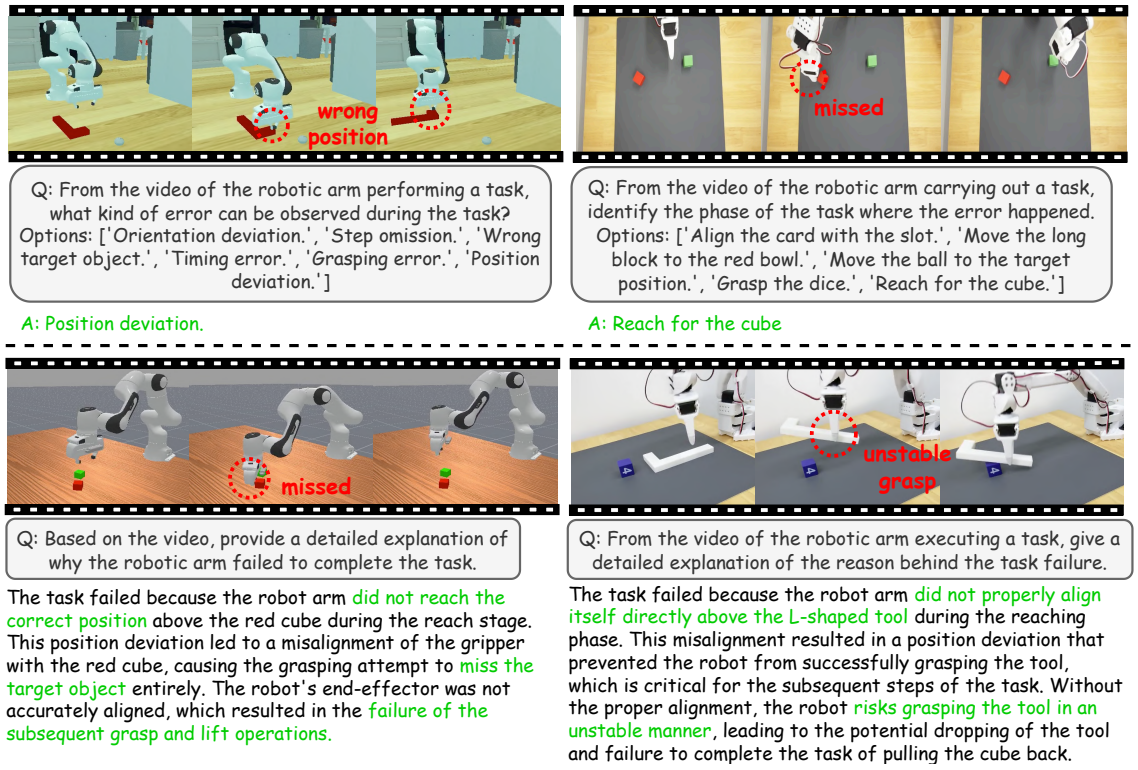

Figure 9: **Failure identification and analysis capabilities on RoboFAC ?.** Top row shows identification of failure types and execution stages. Bottom row demonstrates failure root cause analysis.

180+ real-world environments, evaluated using LLM-based scoring aligned with human preferences. These benchmarks comprehensively evaluate embodied reasoning capability across planning, manipulation, and spatial understanding.

Question: Is this room on the ground floor of the building?
GT: no

Let's carefully analyze the image to determine if this room is on the ground floor of the building.

First, I notice that there is a window with a view of greenery outside. This suggests we are at a level that allows natural light to filter in, which is typical for ground floor windows.

Next, I see a white desk and a chair, which seem like standard office furniture for a ground floor workspace. The desk appears to be at a height that would be comfortable for someone to sit and work, which is also consistent with a ground floor layout.

Additionally, there's a white bookshelf that looks like it could be used for storage and decoration, which is a common feature in office spaces on the ground floor.

However, there is a cabinet in the background that seems a bit unusual for a ground floor room. It could be a storage cabinet for a specific purpose, but it might not be typical for a ground floor space.

Let's also consider the lighting. The natural light coming through the window seems to be coming from the side, which is more likely if we are on the ground floor and looking towards a window that is part of the building's exterior.

After considering all these factors, I think the room is likely on the ground floor, but it's not impossible for it to be on an upper floor with a unique window design or a specialized storage cabinet.

Final thought: Given the typical layout of office spaces and the visible elements, it's reasonable to conclude that this room is on the ground floor.

(a) Textual reasoning (from teacher model)

Let's break this down. The image shows a study room scene with a desk, a chair, and some shelves. There's no clear indication that this is on the ground floor. It's obviously indoors, but I should check if anything suggests a different level. Wait, I should also consider if there's any sign of elevation. Hmm, it seems like the view outside the window shows that the room is positioned above the ground. Oh, I see, the perspective through the window suggests it's not on the ground floor. Let's confirm that. Yes, it checks out.

(b) Verbalized reasoning

Figure 10: **Reasoning trace comparison on OpenEQA.** (a) Teacher's textual reasoning. (b) Student's verbalized latent reasoning. Green: reasonable reasoning trace; red: incorrect trace.

A.4.2. Robotic Manipulation Benchmarks

We evaluate on three simulation benchmarks covering diverse manipulation scenarios. SimplerEnv **?** provides manipulation tasks with strong sim-to-real correlation, featuring diverse visual variations in lighting, textures, backgrounds, and camera poses. Following MolmoAct **?**, we evaluate on the Google Robot tasks using the standard protocol **??** of directly evaluating on SimplerEnv after training on OXE. LIBERO **?** targets different generalization challenges through four task suites: spatial layout variation (LIBERO-Spatial), object diversity (LIBERO-Object), goal variation (LIBERO-Goal), and long-horizon planning with mixed variations (LIBERO-Long). We evaluate each suite over 500 trials using 3 random seeds following prior works **??**. RoboTwin2.0 **?** features challenging bimanual manipulation with easy and hard difficulty settings, where the hard setting introduces domain randomization including clutter, lighting variations, diverse textures, and height changes. Following the original protocol, we train on 50 clean expert demonstrations per task and evaluate with 100 rollouts under both settings. We assess 10 tasks categorized into short, medium, and long horizons based on demonstration lengths.

B. Additional Experiment Results

B.1. Additional Quantitative Results

Results of Larger Model Size.
To demonstrate the scalability of our approach, we apply Fast-ThinkAct to a larger backbone, Qwen2.5-VL-7B, and evaluate its performance on embodied reasoning benchmarks. As shown in Tab. 4, Fast-ThinkAct consistently achieves strong performance across EgoPlan-Bench2 **?**, RoboVQA **?**, and OpenEQA **?**, validating that our latent reasoning distillation method effectively scales to larger model backbones.

Performance Comparison with ThinkAct-3B.
Tab. 5 presents detailed numerical results corresponding to Fig. 3 with additional ThinkAct-3B results. At the same 3B model size, Fast-ThinkAct achieves notable performance gains (89.7 vs. 83.1 on LIBERO, 68.7 vs. 64.7 on SimplerEnv-Google) while dramatically improving efficiency with 7× faster inference (805ms vs. 5674ms).

Comparison with Efficient Reasoning Baselines.
Table 6 compares our method with efficient textual reasoning alternatives applied to the textual teacher $\mathcal{F}_\theta^T$. We evaluate three baselines: removing reasoning during inference entirely (0 tokens), constraining the teacher to generate only 6 textual tokens during inference, and applying RL training with a length penalty **?** to encourage concise reasoning (∼50 tokens). These achieve 46.5, 46.3, and 47.8 respectively, all degrading from the teacher's 49.8. In contrast, Fast-ThinkAct uses only 6 latent tokens and achieves **53.3**, demonstrating superior efficiency and performance.

B.2. Additional Qualitative Results

Qualitative Robot Execution.
We provide qualitative robot execution comparisons between the base action model RDT **?** and Fast-ThinkAct in the supplementary video `Fast-ThinkAct.mp4`. Our method shows substantial improvements on challenging robotic execution tasks, where reasoning capabilities provide better spatial understanding and coordination for successful manipulation.

Bimanual Manipulation Results.
In Fig. 8, we present visualized trajectories and execution results for `hanging mug` and `handover mic` tasks under easy and hard settings in RoboTwin2.0 **?**. The hard setting includes different backgrounds and distractor objects. These examples show successful bimanual coordination where predicted waypoints accurately guide both grippers through the manipulation sequence, demonstrating Fast-ThinkAct's spatial reasoning ability across varied visual conditions.

Failure Identification and Recovery.
In Fig. 9, we demonstrate Fast-ThinkAct's failure identification and analysis capabilities, complementing the recovery planning shown in the main paper. The top row shows that Fast-ThinkAct identifies failure types (e.g., position deviation) and execution stages (e.g., reaching for the cube). The bottom row illustrates root cause analysis, for instance, in the bottom-right example, the model correctly infers that the failure to push the cube with an L-shaped tool stems from an improper initial grasp. These results demonstrate Fast-ThinkAct's comprehensive understanding of manipulation failures beyond recovery planning.

Verbalized Latent Reasoning.
Fig. 10 visualizes teacher textual reasoning and student verbalized reasoning. While the student generates compact and correct (green) reasoning, the teacher's lengthy output sometimes contains erroneous steps (red) that might degrade the performance.

Table 7: **Additional ablation study of training objectives and learning stages on robot manipulation benchmarks.**

Method	LIBERO	SimplerEnv Google	RoboTwin2.0	Average
Fast-ThinkAct	**89.7**	**68.7**	**46.1**	**68.2**
w/o $\mathcal{L}_{\text{verb}}$	88.6	67.3	44.9	66.9
w/o $\mathcal{L}_{\text{verb}}$, $\mathcal{L}_{\text{distill}}$	86.3	65.7	42.6	64.9
Textual Teacher	88.5	67.3	45.8	67.2
SFT + CoT-SFT	87.2	65.8	43.3	65.4
SFT only	86.9	64.5	42.8	64.7

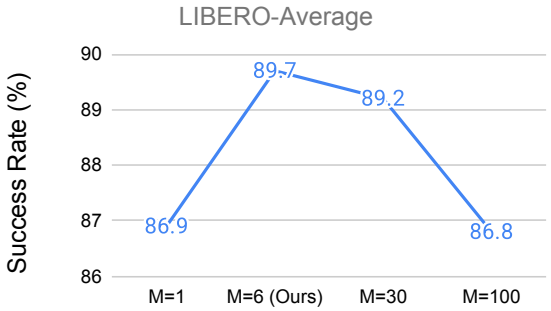

Table 8: **Ablation of Latent Reasoning Steps M.**

B.3. Additional Ablation Study and Analysis

Additional Ablation Results on Manipulation Benchmarks.
Table 7 shows ablation results on LIBERO **?**, SimplerEnv-Google **?**, and RoboTwin2.0 **?**. Removing $\mathcal{L}_{\text{verb}}$ or $\mathcal{L}_{\text{distill}}$ progressively degrades performance, confirming their contributions. Our full model consistently outperforms the textual teacher and models without teacher-student training (CoT-SFT, SFT only), demonstrating the benefits of compact latent reasoning distillation.

Ablation Study on Action Model Conditioning.
In Sec. 3.3, we extract visual latent planning c_t from early-layer KV cache of spatial tokens to condition the action model. We compare this against using late-layer KV cache (last N layers, where N is the action model depth) and directly using spatial tokens' output hidden states. Our approach achieves **89.7** on LIBERO, outperforming late-layer KV at 88.3 and output hidden states at 87.1, demonstrating that early-layer representations better capture visual planning information for action prediction. Therefore, we adopt early-layer KV conditioning as our default configuration.

Ablation Study on Latent Reasoning Steps.
In Fig. 8, we study the effect of latent reasoning steps M. We observe that too few steps ($M = 1$) limit reasoning capacity, while excessive steps ($M = 30, 100$) might introduce redundant or noisy information. Therefore, we adopt $M = 6$, which achieves optimal performance, as our default.