E-TTS: A New Embodied Test-Time Scaling Framework for Robotic Manipulation

Wen Ye[1,2*], Peiyan Li[1,2*], Tingyu Yuan[2,3], Yuan Xu[1,2], Xiangnan Wu[1,2], Chaoyang Zhao[3], Jing Liu[4], Nianfeng Liu[4], Yan Huang[1,2,4†], and Liang Wang[1,2†]

[1] New Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] Foundation Model Research Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China [4] FiveAges, Beijing, China
yewen2025@ia.ac.cn, peiyan.li@cripac.ia.ac.cn
{yhuang, wangliang}@nlpr.ia.ac.cn

Abstract. Recently, a few works have made early attempts to study test-time scaling for embodied tasks. However, two major challenges remain unsolved: (1) reasoning can effectively improve the performance of the policy, but its scaling mechanism has seldom been studied; (2) historical information is essential, as embodied tasks are inherently long-horizon and sequential, making sole reliance on current observations for action scaling inadequate due to the lack of historical context utilization. To address these challenges, we introduce E-TTS, a modular and plug-and-play Embodied Test-Time Scaling framework that unifies reasoning and action scaling for robotic manipulation via history-aware iterative refinement with vision-language verifiers. To support joint reasoning-action scaling, E-TTS performs reasoning-action joint sampling and scoring in a pairwise manner. To better utilize historical information, E-TTS uses a history buffer to store historical context, which is then used by reasoning and action verifiers to evaluate the sampled candidates. Unlike conventional open-loop TTS methods, E-TTS introduces feedback generation into the sampling process to form a closed-loop iterative refinement mechanism, enhancing both inference efficiency and environmental adaptability. Each component functions as an independent and composable module, allowing flexible and adaptive configuration depending on task requirements. To evaluate the advantages of our framework, we conduct experiments across 4 different benchmarks, 6 environments, 3 embodiments, and 4 base vision-language-action models. The experimental results demonstrate that, without requiring additional expert data collection or retraining, E-TTS consistently improves performance, achieving up to a 33.14% increase in simulation and 26.62% in real-world scenarios. Our project page is https://27yw.github.io/E-TTS-Web/.

*Equal contribution.
†Corresponding authors.

Keywords: Vision–Language–Action models · Test time scaling · Robotic manipulation

1 Introduction

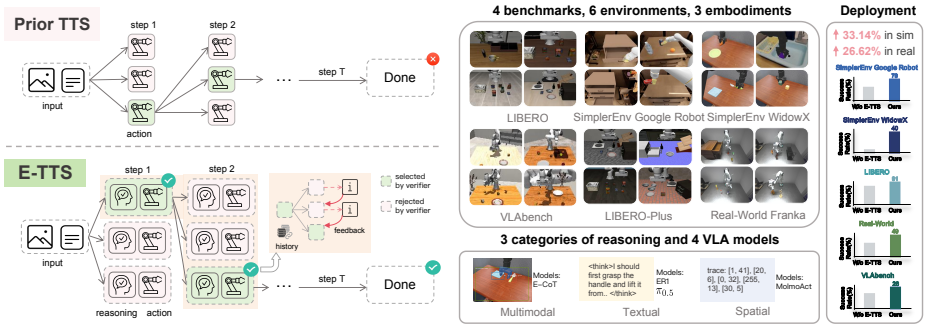

Fig. 1: Overview. E-TTS is an embodied test-time scaling framework that integrates reasoning and action scaling for robotic manipulation through history-aware, closed-loop interactions with vision-language verifiers. When combined with standard VLA models, E-TTS consistently enhances performance, achieving up to a 33.14% improvement in simulation and 26.62% in real-world scenarios.

Test-time scaling (TTS) has gained significant attention in fields such as computer vision [41] and natural language processing as it can improve model's performance without requiring additional data or retraining. Such an advantage is even more appealing for the embodied domain, as collecting real-world robotic data is substantially more expensive than obtaining internet vision–language data. Fundamentally, TTS trades additional inference-time computation for higher output quality. In the embodied domain, there is a wide range of latency-insensitive tasks (e.g., object rearrangement), which prioritize task success over millisecond-level latency. For these scenarios, TTS offers a promising path toward robust execution. Consequently, a pivotal question arises: how can we apply Test-Time Scaling (TTS) in the embodied domain with an optimal balance between computation and performance?

Recent works [11,15,17] explore test-time scaling (TTS) for embodied systems (see Fig. 1), as shown in the upper part of Fig. 1, yet they overlook two intrinsic challenges of robotic manipulation. First, it has become popular to incorporate a reasoning component before action prediction [14,19,27,45,46]. However, existing TTS methods focus solely on scaling actions, causing misalignment between high-level planning and execution, and thus suboptimal manipulation performance. Unlike conventional vision-language tasks, manipulation is a sequential decision-making process. Historical trajectory context is vital for candidate evaluation, while iterative feedback is necessary to refine reasoning and actions. Current TTS

approaches lack mechanisms to integrate this history, limiting their effectiveness in complex embodied scenarios.

To effectively address these two challenges, we propose a unified embodied test-time scaling framework (E-TTS) that jointly scales reasoning and action through history-aware, closed-loop interaction and refinement with vision–language verifiers (Fig. 1). Specifically, to tackle the first challenge, E-TTS performs joint reasoning–action sampling, where the reasoning process and low-level action generation are sampled and selected based on the unified score that is jointly modeled from both dimensions. For reasoning selection, we prompt the VLM to perform zero-shot scoring of each reasoning candidate. For action selection, we construct an action preference dataset, based on which a separate VLM is trained as a *verifier* to score candidate actions. To address the second challenge, we use a history buffer to store historical information, which is then incorporated into the input context to improve the verifier's ability to model the temporal dependency. Moreover, our framework integrates feedback generation at sampling step. This generated feedback is then incorporated into subsequent sampling rounds, creating an iterative refinement process that enhances both efficiency and adaptability. E-TTS features a highly flexible architecture where every component is optional, independently configurable, and can be combined with others, providing an adaptable framework that accommodates a wide range of tasks and different VLA models without additional training.

To validate its effectiveness, we integrate the proposed E-TTS with four representative vision-language-action models, all of which take vision-language inputs and output actions, but with different types of intermediate reasoning. The evaluation experiments are conducted across six environments: SIMPLER WidowX, SIMPLER Google Robot, LIBERO, LIBERO-Plus, VLAbench, and the real world. The average success rates are improved with a maximum gain of 33.14% and an average gain of 13.52%. Additionally, we conduct extensive ablation studies on the key components to identify the sources of improvement and explore the trade-off between reasoning scaling and action scaling, finding that both are important for embodied test-time scaling.

To summarize, our main contributions are threefold:

- We introduce a novel plug-and-play *embodied test-time scaling* framework that can be seamlessly integrated with various different vision-language-action models, enhancing their success rates without the need for additional expert data collection or retraining.
- We tackle the unique challenges of the embodied domain by jointly scaling reasoning and action through history-aware, iterative refinement with vision–language verifiers, leading to significant performance improvements over conventional test-time scaling methods.
- We conduct extensive experiments in both simulated and real-world environments to validate the effectiveness of our framework. The results consistently show that our approach significantly boosts task success rates across a variety of environments and embodiments.

2 Related Work

2.1 Vision-Language-Action Models

With the rapid development of embodied domain [5, 7–9, 22, 23, 30, 31, 37, 38, 43], vision-Language-Action (VLA) models [14, 16, 21] have recently become a popular research direction, aiming to unify visual perception, language understanding, and decision-making within a single framework. However, simply coupling vision, language and motor signals is insufficient for addressing the reasoning demands in complex embodied environments. Thus, recent research emphasizes embedding reasoning as explicit action tokens, which is able to externalize internal cognitive processes before generating concrete actions [48]. Some reasoning-based VLAs, such as E-CoT [45] and RAD [10], incorporate multimodal structured chain-of-thought (CoT) representations into policy learning by synthesizing large-scale reasoning-action datasets. Some other works, such as ThinkAct [13], $\pi_{0.5}$ [14], and Embodied-R1 [44], generate textual reasoning through reinforcement learning and planning, leveraging multimodal latent planning [13], heterogeneous data co-training [14], and embodied pointing representations [44] to improve long-horizon generalization and spatial reasoning. More recently, EMMA-X [32] and MolmoAct [19] extend this reasoning-enhanced paradigm toward spatially grounded multimodal action reasoning, in which models predict sub-tasks, scene descriptions, and fine-grained motor trajectories grounded in visual context, demonstrating strong generalization in real-world robotic manipulation. Despite significant progress, most VLAs rely on scaling dataset and model sizes during training to improve final performance, which limits their adaptability in environments with constraints on data and model size.

2.2 Test Time Scaling

Recent LLM progress has shifted focus to test-time scaling, which allocates more inference-phase computation via three primary strategies: Parallel scaling [4, 18, 20, 28] improves reliability by generating multiple candidates and aggregating them through selection mechanisms. Sequential scaling [6, 26, 36, 42, 49] enables iterative refinement, allowing models to progressively build and revise intermediate reasoning. Hybrid scaling [2, 35, 40] unifies these strengths, exemplified by Tree-of-Thought and Graph-of-Thought, to provide structured, adaptive exploration and deliberation. Inspired by these language-centric efforts, we explore test-time scaling for the embodied domain. Our framework jointly scales reasoning and action spaces while leveraging history-aware joint verification and selection to coordinate exploration and refinement during execution.

2.3 Embodied Test Time Scaling

Few studies have extended the concept of test-time scaling to the embodied domain, aiming to enhance inference-time performance without retraining the underlying policy. For example, Hume [29] introduces a dual-system framework

that augments a Vision-Language-Action (VLA) model with a value head for repeated sampling and cascaded denoising, enabling the model to re-evaluate and refine its action proposals during inference. Similarly, RoboMonkey [17] explores large-scale reward modeling and synthetic data generation to support the external evaluation of action candidates. RoVer [11] designs a robot process-reward model for VLA systems, enabling them to score and refine candidate actions at inference time. TACO [39] designs a test-time scaling framework using pseudo-count estimation. However, these methods primarily scale within the action space, neglecting the tight coupling between reasoning and action. In contrast, our E-TTS jointly scales reasoning and action, significantly improving the success rate. Furthermore, these approaches fail to fully leverage historical context and the feedback, neglecting the sequential decision-making characteristics of the embodied domain. In contrast, we propose a history-aware, framework-level closed-loop framework that effectively addresses these unique challenges.

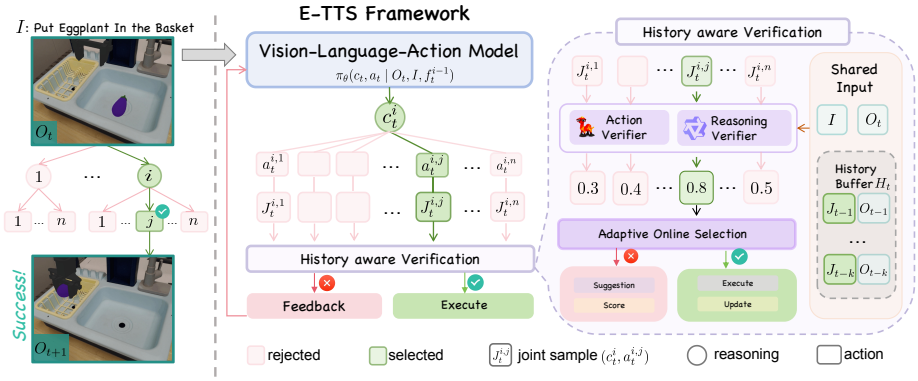

Fig. 2: Overview of the proposed E-TTS framework. At timestep t, given an instruction I and the current observation O_t, the Vision-Language-Action model jointly samples multiple reasoning-action pairs $J_t^{i,j} = (c_t^i, a_t^{i,j})$. Each candidate is evaluated by the History-aware Verification module based on history buffer H_t, O_t and I, which applies both reasoning and action verifiers to assess consistency. Through Adaptive Online Selection, the best pair is executed; otherwise, Feedback-Guided Refinement is triggered to resample improved candidates.

3 Embodied Test-Time Scaling

3.1 Framework Overview

As shown in Fig. 2, given a task instruction I and the current observation O_t, our method first performs **Reasoning-Action Joint Sampling (Sec. 3.2)**. This step involves sampling a diverse set of candidate pairs $J_t^{i,j} = (C_t^i, a_t^{i,j})$,

where C_t^i represents an intermediate reasoning step and $a_t^{i,j}$ is the corresponding action. These joint samples, along with previous observations, are stored in a history buffer denoted as H_t. The buffer is then utilized by our **History-aware Verification and Selection (Sec. 3.3)** module, which employs a dual-verifier (**Reasoning (Sec. 3.3)** and **Action (Sec. 3.3)**) to assess the candidates' semantic and physical consistency through a joint scoring strategy. Next, we apply **Adaptive Online Selection (Sec. 3.3)** to evaluate the suitability of the candidate pair. If the confidence of the selected pair exceeds a predefined threshold, the corresponding action is executed. Otherwise, if no satisfactory candidate is found, the **Feedback-Guided Iterative Refinement (Sec. 3.4)** mechanism is triggered, providing corrective feedback to the VLA model. This feedback guides the model to resample more accurate and contextually aligned pairs.

E-TTS is designed as a plug-and-play framework, and all modules are fully modular and configurable: they can be independently enabled, disabled, or combined depending on the task complexity, allowing E-TTS to adapt to different tasks and VLA architectures while maintaining a unified framework. The pseudocode for the entire pipeline is presented in Appendix A.2.

3.2 Reasoning-Action Joint Sampling

Prior works [11, 15, 17] primarily focus on scaling the action space, often overlooking the intermediate reasoning process. However, in complex embodied tasks, reasoning outcomes guide action predictions and, consequently, influence the final action results. To address this, E-TTS introduces a reasoning-action joint scaling strategy that aims to identify the optimal $\langle$reasoning, action$\rangle$ pair rather than selecting a single action.

Specifically, we employ a general robot policy $\pi_\theta(c_t, a_t \mid O_t, I)$, where O_t denotes the current observation, I represents the task instruction, c_t is the reasoning result, and a_t is the executed action. At each timestep t, we first sample a reasoning c_t^i ($i \in [1, M]$), based on which the policy generates N candidate actions $\{a_t^{i,j}\}_{j=1}^N$. We define each $\langle$reasoning, action$\rangle$ pair as a joint unit, and the collection of all pairs corresponding to the same reasoning sample forms a joint sample batch:

$$J_t^{i,j} = (c_t^i, a_t^{i,j}),\ \mathcal{B}_t^i = \{J_t^{i,j}\}_{j=1}^N \tag{1}$$

where M is the maximum number of reasoning samples at each timestep and N is the number of action samples for each reasoning. This joint sampling paradigm generates a structured search space of reasoning-action pairs for the subsequent verification and selection module.

Furthermore, since embodied tasks are inherently sequential decision processes, the final decisions depend not only on the current observation but also on the historical context. To account for this, we store the selected $\langle$reasoning, action$\rangle$ pairs along with past observations in a history buffer, providing historical experience for other modules.

Specifically, at timestep t, we maintain a dynamic buffer:

$$\mathcal{H}_t = \{J_{t-K}, O_{t-K}, \ldots, J_{t-1}, O_{t-1}\} \tag{2}$$

which stores the most recent K reasoning–action pairs. This history buffer operates in a sliding-window manner with a total length of K: when a new joint pair J_{t+1} is generated and verified, it is appended to $\mathcal{H}_t$, while the oldest entry J_{t+1-K} is discarded, resulting in $\mathcal{H}_{t+1}$. More details can be found in Appendix A.1.

3.3 History-aware Verification and Selection

Building upon the joint sampling strategy and history buffer, we introduce a dual-verifier mechanism to evaluate the joint samples.

Reasoning Verifier (V_c) We employ a vision–language foundation model (Qwen2.5-VL-7B [1]) as the reasoning verifier in a zero-shot manner to assess the validity, coherence, and groundedness of candidate reasoning samples.

Formally, for each $J_t^{i,j}$ in the $\mathcal{B}_t^i$, given the history-aware reasoning sequence $\mathcal{H}_t$, the current observation O_t, and the task instruction I, the reasoning verifier generates a confidence score:

$$S_c^{i,j} = V_c(\mathcal{H}_t, J_t^{i,j}, O_t, I) \tag{3}$$

where S_c quantifies the likelihood that the reasoning sample will lead to successful task completion within the current embodied context.

To address the variety of intermediate reasoning results, we categorize them into three complementary types and design specialized processing strategies:

Textual Reasoning. This type of reasoning involves purely linguistic content, such as subtasks. Embodied-R1 [44] is one of the representative manipulation models outputting such intermediate reasoning. Our reasoning verifier processes these samples through contextual embedding to evaluate logical consistency, causal soundness, and task relevance.

Multimodal Reasoning. This reasoning integrates both visual perception (e.g., object detection and segmentation results) and textual understanding (e.g., scene descriptions and task decomposition). E-CoT [45] generates this type of reasoning. Such reasoning requires the reasoning verifier to assess cross-modal consistency. In our framework, we render the visual perception directly on the image as visual prompts and provide these alongside the original textual understanding to avoid losing critical spatial details.

Spatial Reasoning. Spatial reasoning typically involves information related to spatial relationships, such as MolmoAct's [19] trajectory keypoints. In our framework, we transform these spatial reasoning results into visual prompts and overlay them on the original observations. To help the verifier better understand spatial context, we construct representative examples that show what constitutes a valid reasoning prediction in the current environment. This approach enables V_c to learn to differentiate between well-grounded and inconsistent spatial reasoning, ensuring robust evaluation of spatial intent. Further details on reasoning and our corresponding strategies are provided in Appendix A.5.

Efficiency Optimization Introducing reasoning scaling inevitably brings extra computation. To reduce inference latency, we incorporate several system-level optimizations. We deployed the verifier using the vLLM engine and utilize PagedAttention to optimize GPU memory utilization. To further accelerate the process, we adopt a prefix-sharing strategy that caches the KV state of static instructions. Notably, our final approach incurs minimal additional time, with a comprehensive analysis provided in Sec. 4.3.

Action Verifier ($V_\mathbf{a}$) Inspired by [17], our action verifier evaluates the low-level feasibility and task relevance of candidate actions. The action verifier (LLaVA-7B [33]) is trained on 90k paired demonstrations that include successful and failed policy rollouts. The data are collected from both SimplerEnv and LIBERO by sampling paired good and bad actions. The quality labels are assigned by comparing the sampled actions with ground-truth actions using an MSE-based criterion. It learns through a modified Bradley–Terry [3] objective that incorporates graded preference levels between action pairs. Specifically, it minimizes the difference between the ground-truth and predicted preference margins, enhancing sensitivity to varying action quality. Following RT-2 [50], we discretize each dimension of the robot's continuous actions into 256 bins during training.

During inference, for action in the pair $J_t^{i,j}$ in a joint sample batch, the verifier V_a predicts a scalar score:

$$S_a^{i,j} = V_\mathrm{a}(J_t^{i,j}, O_t, I) \tag{4}$$

which reflects the action feasibility under the current embodied context. More details can be found in Appendix A.6.

Adaptive Online Joint Selection To evaluate the $\langle \text{reasoning}, \text{action} \rangle$ pair $J_t^{i,j}$, we combine the normalized scores from both verifiers. The final joint score $S_t^{i,j}$ for each pair $(c_t^i, a_t^{i,j})$ is computed as:

$$S_t^{i,j} = S_c^{i,j} \times \hat{S}_a^{i,j} \tag{5}$$

where $S_c^{i,j}$ represents the score of the reasoning trace c_t^i, and $\hat{S}_a^{i,j}$ is the normalized score of the action sample $a_t^{i,j}$ for reasoning c_t^i. This ensures that only $\langle \text{reasoning}, \text{action} \rangle$ pairs with both coherent reasoning and feasible actions receive high joint scores. The mathematical justification is provided in Appendix A.3.

While this joint scoring mechanism effectively measures the consistency between reasoning and action, greedily searching to select the pair with the highest score can limit the model's exploration of the solution space, potentially leading to a sub-optimal or local optimum solution. To address this issue, we adopt an adaptive online selection strategy based on an ϵ-greedy policy, which balances exploration and exploitation during inference.

At each timestep t, a random variable $u \sim \mathrm{Uniform}(0, 1)$ determines the selection mode: with probability ϵ, a random reasoning–action pair is sampled from

the current batch $\mathcal{B}_t^i$ for exploration; otherwise, the pair with the highest joint score is chosen greedily: $J_t^{\mathcal{B}_i} = J_t^{i,j^*}, \quad j^* = \arg\max_j S_t^{i,j}$, where $S_t^{i,j}$ denotes the joint score of pair $J_t^{i,j}$.

To further avoid low-quality selections, a threshold η is applied:

$$J_t^{\mathcal{B}_i} = \begin{cases} \text{Random}(\mathcal{B}_t^i), & u \leq \epsilon \\ J_t^{i,j^*}, & u > \epsilon \text{ and } S_t^{i,j^*} > \eta \\ \text{next batch } \mathcal{B}_t^{i+1}, & u > \epsilon \text{ and } S_t^{i,j^*} \leq \eta \end{cases} \tag{6}$$

This adaptive policy dynamically allocates computation to promising candidates while maintaining the model's ability to explore the solution space. More details can be found in Appendix A.7.

3.4 Feedback-Guided Iterative Refinement

To help the model find a solution more quickly and avoid repeatedly visiting suboptimal regions, the rejection of an entire joint batch triggers a feedback-guided iterative refinement mechanism.

We reapply our reasoning verifier to analyze the failure causes for the rejected samples and generate structured textual feedback. Specifically, the model is prompted with a set of guided questions aimed at eliciting targeted critiques.

Given the current observation O_t, instruction I, and the failed sample $J_t^{i,j}$, the verifier produces a textual suggestion F_t^i that describes the reasoning flaw and offers actionable corrections. This feedback F_t^i is then inserted into the original instruction prompt, forming a revised context that conditions the next round of joint batch sampling $\mathcal{B}_t^{i+1}$:

$$I_t^{i+1} = \text{Concat}(I, F_t^i) \tag{7}$$

where I_t^{i+1} denotes the refined instruction that conditions the base policy.

This closed-loop refinement allows the model to learn from its own errors, progressively improving reasoning coherence and action feasibility until a high-score solution is obtained or a maximum batch limit M is reached.

We provide additional details, including prompts (Appendix A.4), refinement strategies (Appendix A.1), and examples of iterative refinement (Appendix A.8) and detailed suggestions (Appendix A.9).

4 Experiments

We conduct extensive experiments to evaluate the effectiveness of E-TTS. Our experiments are designed to address the following research questions:

Q1: Is E-TTS general enough to enhance the performance of various manipulation methods that rely on different intermediate reasoning results?

Q2: Are specific designs for embodied scenarios, such as jointly scaling reasoning and action and incorporating feedback and history, truly beneficial?

Table 1: Performance comparison on three embodied manipulation tasks on Simpler widowx robot (Visual Matching). The proposed method (E-CoT + **Ours**) consistently outperforms all baselines and ablation variants. Gray rows denote ablation settings removing specific components such as feedback, scaling, or history buffer.

Method	Spoon on Towel	Eggplant in Basket	Carrot on Plate	Average
E-CoT	20.00	0.00	0.00	6.67
E-CoT + naive TTS	41.67	4.17	25.00	23.61
E-CoT + Robomonkey	33.33	8.33	37.50	26.38
w/o feedback	50.00	11.50	13.63	25.04
w/o reasoning scaling	50.00	4.17	25.00	26.39
w/o action scaling	45.83	16.67	29.17	30.56
w/o joint scoring	42.85	18.75	31.80	31.13
w/o history buffer	54.17	20.83	20.83	31.94
w/o ϵ-greedy	50.00	20.83	37.50	36.11
E-CoT + **Ours**	**58.33**	**22.22**	**38.89**	**39.81**

Q3: What is the efficiency–performance trade-off of E-TTS, and can it achieve substantial gains without highly excessive latency?

Q4: What is the optimal trade-off between reasoning and action scaling within a fixed computational budget?

Q5: Does E-TTS maintain strong performance in real-world environments?

4.1 Experiments on Different Types of Manipulation Methods

We evaluate E-TTS on four representative vision-language-action models: E-CoT [45], MolmoAct [19], $\pi_{0.5}$ [14] and Embodied-R1 [44]. While all map multimodal inputs to actions, their intermediate reasoning differs. Specifically, E-CoT generates reasoning outputs including object bounding boxes, gripper positions, subtasks, and motion primitives; MolmoAct produces depth predictions and visual traces; $\pi_{0.5}$ and Embodied-R1 generates task plans. More details about these three models are provided in Appendix B.2.

We further compare against RoboMonkey [17] and a naive test-time scaling baseline. This naive test-time baseline also scales the reasoning and action, verifying them through the same vision-language foundation model as ours. However, it lacks other embodiment-related adaptations, such as closed-loop feedback or history-based inputs. For fair comparison, RoboMonkey is also integrated with E-CoT in SimplerEnv WidowX. RoboMonkey scales only actions with the same setting of E-TTS. The evaluation interval is 10 steps in the experiments. Detailed hyperparameter settings are presented in Appendix B.1.

Experiments on E-CoT. We integrate E-TTS into E-CoT and evaluate it on the SimplerEnv benchmark [24]. SimplerEnv is a suite of real-to-sim environments designed for evaluating robot policies in simulation. It provides a standardized arena for benchmarking the success rates of robot policies developed for private real-world platforms such as *Google Robot* and *WidowX*. In this paper, we evaluate E-CoT on the *WidowX* platform, using the BridgeData V2 [34], which contains 60,096 trajectories collected from 24 environments. Each method is evaluated over three task types with 24 trials per task. The experimental results are reported in Tab. 1. We observe that the original E-CoT

Table 2: Comparison of visual and variant performance across tasks on SimplerEnv. MolmoAct + **Ours**, significantly outperforms the MolmoAct and baselines, achieving the highest average scores in both Visual (80.00%) and Variant (77.77%) settings.

Method	Visual				Variant			
	Pick Coke Can	Move Near	Open/Close Drawer	Average	Pick Coke Can	Move Near	Open/Close Drawer	Average
MolmoAct	74.50	75.45	63.25	71.05	66.95	52.55	77.75	65.68
MolmoAct + naive TTS	76.47	80.00	66.60	74.36	56.00	50.00	80.00	62.00
MolmoAct + **Ours**	**83.00**	**85.00**	**72.00**	**80.00**	**80.00**	**70.00**	**83.30**	**77.77**

baseline performs poorly, and it often fails to accurately approach the target object. In contrast, when integrated with our E-TTS pipeline, E-CoT produces more appropriate reasoning outputs, resulting in a substantial improvement in average task success rate from 6.67% to 39.81%. Compared to the Robomonkey, our method achieves a significant 50.91% improvement in average success rate, while maintaining a highly competitive inference efficiency with only a marginal 7.8% increase in execution time. More results can be found in Appendix B.5.

Experiments on MolmoAct. We evaluate E-TTS using MolmoAct on SimplerEnv (*Google Robot*), LIBERO [25], and LIBERO-Plus [12]. For SimplerEnv, we test across two settings: (1) Visual Matching, mirroring training setups, and (2) Variant Aggregation, featuring randomized visual conditions. We integrate our framework into both the official zero-shot and fine-tuned MolmoAct checkpoints and the overall success rate is the average across all scene variants. As shown in Tab. 2, E-TTS consistently boosts success rates across all settings, significantly outperforming the naive test-time scaling baseline.

Table 3: Performance comparison on the **LIBERO** benchmark and **LIBERO-Plus** benchmark across four reasoning categories (Spatial, Object, Goal, Long-Horizon). MolmoAct + **Ours** consistently outperforms the baseline and naive variants.

Setting	Spatial	Goal	Object	Long
MolMoAct	88.80	84.30	90.40	77.46
MolMoAct + Ours	**91.20**	**85.04**	**91.36**	**80.99**

(a) LIBERO-Plus Benchmark

Method	Spatial	Object	Goal	Long	Average
MolmoAct	87.00	92.67	87.60	75.00	85.57
MolmoAct + naive TTS	88.90	93.10	70.00	75.00	81.75
MolmoAct + **Ours**	**93.60**	**97.50**	**92.00**	**80.00**	**90.78**

(b) LIBERO Benchmark

In the LIBERO benchmark, a simulated Franka Emika Panda arm performs manipulation tasks conditioned on multimodal demonstrations, which consist of front and wrist camera observations, natural language instructions, and delta end-effector pose actions. LIBERO provides four task suites—*Spatial*, *Object*, *Goal*, and *Long*—each containing ten distinct tasks. During evaluation, we perform 50 rollouts per task. Detailed results are reported in Tab. 3. When integrated with E-TTS, MolmoAct achieves the highest performance across all four task suites, further demonstrating the effectiveness of our approach. More results can be found in Appendix B.7.

We further evaluate generalization on LIBERO-plus [12], which introduces 10,030 tasks across seven perturbation dimensions, including viewpoint changes, object layout shifts, robot initialization variations, instruction rewriting, lighting, background textures, and sensor noise. We evaluate every first 200 tasks in each category, 800 tasks in total, as shown in Tab. 3, our method consistently improves MolMoAct across all categories. The improvements under these diverse perturbations demonstrate enhanced robustness and stronger generalization beyond the standard LIBERO setting.

Table 4: Success Rate (SR) for representative tasks on the **VLABench** benchmark.

Track	IF	SB	SCT	SPo	AC	SD	Avg.
Track 1	0.18	0.48	0.48	0.46	0.50	0.42	0.39
Track 2	-	0.02	0.15	0.40	0.06	0.14	0.20
Track 3	0.08	**0.38**	0.50	-	0.14	0.14	0.19
Track 4	0.04	0.38	0.21	0.06	0.10	0.20	0.15
Track 6	**0.06**	0.45	0.29	0.22	**0.46**	0.30	0.25

(a) $\pi_{0.5}$

Track	IF	SB	SCT	SPo	AC	SD	Avg.
Track 1	**0.20**	**0.53**	**0.53**	**0.63**	**0.54**	**0.48**	**0.42**
Track 2	-	**0.02**	**0.16**	**0.48**	**0.08**	**0.14**	**0.24**
Track 3	**0.10**	0.36	**0.54**	-	**0.18**	**0.16**	**0.21**
Track 4	**0.06**	**0.48**	**0.24**	**0.08**	**0.16**	**0.22**	**0.16**
Track 6	0.04	**0.48**	**0.31**	**0.28**	**0.31**	**0.34**	**0.26**

(b) $\pi_{0.5}$+Ours

Experiments on $\pi_{0.5}$. We further integrate our framework with $\pi_{0.5}$, a flow-matching-based VLA model with intermediate subtask prediction. It is evaluated on VLABench [47], a benchmark specifically designed to assess long-horizon reasoning, world knowledge transfer, semantic instruction understanding, and physical law awareness. It contains composite tasks requiring multi-step planning and implicit intention understanding. As shown in Tab. 4, integrating our method consistently improves $\pi_{0.5}$ [14] across most tracks and representative tasks (Details can be found in Appendix B.8), with average success rates increasing on all reported tracks. Notably, gains are more pronounced on tasks such as *select_ poker*(from 0.46 to 0.63) and *add_ condiment*(from 0.10 to 0.16), which require multi-step spatial reasoning and semantic grounding. These results indicate that under complex and compositional task settings, it enhances decision consistency across steps and improves long-horizon success rates with E-TTS. The Avg. is of all 12 tasks and full results can be found in Appendix B.8. Moreover, we evaluate E-TTS with TACO [39], which can be found in Appendix B.8.

Experiments on Embodied-R1 (ER-1). We evaluate ER-1 on the SimplerEnv *WidowX* environment using the same evaluation protocol as E-CoT. Due to space constraints, detailed results are provided in Appendix B.6. We observe that integrating E-TTS substantially improves the original model's average success rate from 39.8% to 44.9%.

These results address **Q1**, demonstrating that E-TTS can effectively enhance the performance of manipulation methods regardless of the type of their intermediate reasoning processes.

4.2 Ablation Studies

To validate the effectiveness of our model design and provide insights for the community, we conduct five ablation studies on the SimplerEnv *WidowX* environment using E-CoT as the base manipulation model. The results are in Tab. 1.

w/o feedback removes feedback from the scaling step, leading to a significant performance drop and demonstrating its importance in the scaling process.

Table 5: Ablations on **SimplerEnv**. R/A is reasoning/action.

Method	R/A Samples	SR (%)	Avg Time (s)
E-CoT	–	20.00	8.35
E-CoT + ours	10 / 20	37.50	14.57
E-CoT + ours	10 / 30	41.67	11.46
E-CoT + ours	10 / 50	50.00	12.23
E-CoT + ours	20 / 20	59.10	14.77
E-CoT + ours	50 / 100	50.00	15.65
E-CoT + ours	100 / 100	62.50	20.22

(a) Ablations on sample scale and efficiency

K	SR (%)	η	SR(%)	ϵ	SR(%)
1	37.50	0.1	**70.83**	0.1	**54.17**
3	33.33	0.2	45.83	0.2	50.00
5	45.83	0.3	50.00	0.3	37.50
10	**58.33**	0.4	37.50	0.5	37.50
20	45.83	0.5	33.33	0.7	33.33

(b) Ablations on hyperparameter

w/o action scaling and **w/o reasoning scaling** apply only one scaling component. Using either alone substantially degrades performance, indicating that both are necessary for embodied tasks.

w/o joint scoring evaluates reasoning and action separately. We first select the highest-scoring reasoning, then sample and score actions conditioned on it. This strategy performs poorly, as high-quality reasoning does not guarantee high-quality actions. Effective evaluation requires joint scoring (Sec. 3.2).

w/o history buffer removes historical inputs during evaluation, reducing success rate, demonstrating the importance of trajectory information.

w/o ϵ-greedy disables exploration, resulting in lower performance and highlighting the need to balance exploration and exploitation.

Parameter analysis. We perform a ablation (right part of Tab. 5) on history length K, threshold η, and ϵ exploration. Performance shows that increasing K improves success up to $K = 10$ but drops for longer histories. Moderate η and small ϵ achieve the best trade-off between exploration diversity and reasoning precision, addressing sampling scale and efficiency–performance trade-offs.

Overall, these ablation results address **Q2**, confirming that our design choices are well-motivated and may provide useful insights for future research.

4.3 Trade-off between performance and latency

Test-time scaling inherently trades inference time for performance. However, our empirical results show that this trade-off is favorable. As detailed in Table 5, an optimal scaling configuration yields a 150% relative improvement in success rate with only a 46.6% increase in latency, reflecting high sampling efficiency. By integrating the system-level optimizations from Sec. 3.3, we further reduce this

overhead. Compared to Robomonkey, our method achieves a 51.5% relative gain in average success rate with a marginal 7.8% increase in execution time. These results directly answer **Q3**, yielding a highly favorable efficiency–performance trade-off. Furthermore, E-TTS is model-agnostic and compatible with various VLA architectures. Its modular design allows for composable, selective deployment, enabling flexible joint scaling across reasoning and action dimensions to match task complexity while effectively controlling computational overhead and inference latency.

4.4 Trade-off Between Action Scaling and Reasoning Scaling

One of the main contributions of this paper is the proposal to scale both reasoning and action. A natural question that arises is how to allocate computational resources when they are limited. We test five distribution ratios using the E-CoT setup from Sec. 4.1, with a 60-step sampling limit. Results in Fig. 3 show that over-prioritizing either component yields suboptimal performance. The peak performance at an approximately 1:1 ratio (**Q4**) underscores that both reasoning and action scaling are indispensable for effective embodied test-time scaling.

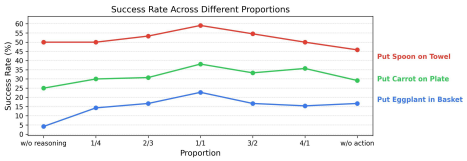

Fig. 3: Success rate across different proportions. The '1/1' proportion achieves peak performance across all tasks.

4.5 Real-robot Experiments

Setup. To validate the effectiveness of our framework in real-world settings, we also conduct real-robot experiments. Our setup features a Franka Research 3 arm with a parallel-jaw gripper, a static ZED 2i depth camera for global views, and a wrist-mounted Realsense D405 for local observations (Fig. 4). We evaluate E-TTS using a finetuned MolmoAct across four tasks, ranging from simple pick-and-place to complex, long-horizon sequences (e.g., placing a coke in a drawer by opening/closing it). These tasks involve both rigid and deformable objects (e.g., a cloth bag). We collected 100 teleoperated expert demonstrations per task via SpaceMouse teleoperation. Each baseline is evaluated over 40 trials per task (160 trials total), with initial object configurations manually standardized for consistency.

Results. Real-robot results (Fig. 4) show that E-TTS boosts the original model's average success rate from 22.13% to 48.75%, achieving great performance across all tasks and addressing **Q5**. Notably, we observed emergent self-correction behaviors: for instance, after a missed grasp, the robot may autonomously reattempt the action, a capability absent in training. This rollout is visualized in Appendix C. We hypothesize that this stems from the vision-language foundation

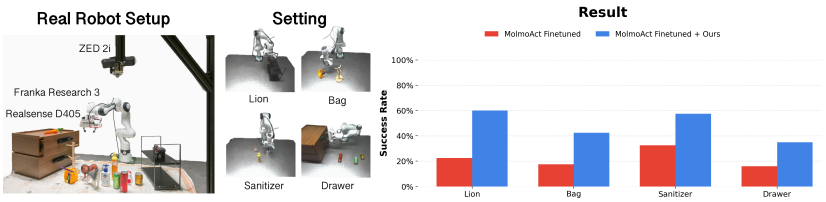

Fig. 4: Real-Robot Experiments. Overview of the real-world robotic setup and evaluation results. Left: Hardware configuration including ZED 2i camera, Franka Research 3 arm, and RealSense D405. Middle: Representative manipulation tasks. Right: Success rate comparison, where our method (blue) significantly outperforms the MolmoAct-Finetuned baseline, achieving an average improvement of 26.62%.

model's extensive pretrained knowledge, enabling effective reasoning in out-of-distribution states. More details about task and results are in Appendix C.

5 Conclusion and Future Work

We propose an embodied test-time scaling framework that addresses the unique challenges of embodied tasks. It can be integrated with various manipulation methods to enhance performance without requiring additional expert teleoperation data. Through experiments across different environments and embodiments, we validate our method's effectiveness. Though our integrated efficiency optimizations mitigate the latency, future research will explore more acceleration techniques.

Acknowledgements

This work was jointly supported by National Natural Science Foundation of China (62322607, 62236010 and 62276261), Beijing Natural Science Foundation (L252033).

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
2. Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., Gajda, J., Lehmann, T., Niewiadomski, H., Nyczyk, P., et al.: Graph of thoughts: Solving elaborate problems with large language models. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 17682–17690 (2024)
3. Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika **39**(3/4), 324–345 (1952)

4. Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q.V., Ré, C., Mirhoseini, A.: Large language monkeys: Scaling inference compute with repeated sampling. arXiv preprint arXiv:2407.21787 (2024)
5. Cai, R., Guo, J., He, X., Jin, P., Li, J., Lin, B., Liu, F., Liu, W., Ma, F., Ma, K., et al.: Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. arXiv preprint arXiv:2602.12684 (2026)
6. Chen, X., Lin, M., Schärli, N., Zhou, D.: Teaching large language models to self-debug. arXiv preprint arXiv:2304.05128 (2023)
7. Chen, Y., Huang, Y., He, K., Li, P., Wang, L.: Verm: Leveraging foundation models to create a virtual eye for efficient 3d robotic manipulation. IEEE Robotics and Automation Letters **11**(3), 2482–2489 (2026)
8. Chen, Y., Li, P., Huang, Y., Yang, J., Chen, K., Wang, L.: Ec-flow: Enabling versatile robotic manipulation from action-unlabeled videos via embodiment-centric flow. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11958–11968 (2025)
9. Chen, Y., Li, P., Yang, J., He, K., Wu, X., Xu, Y., Wang, K., Liu, J., Liu, N., Huang, Y., et al.: Bridgev2w: Bridging video generation models to embodied world models via embodiment masks. arXiv preprint arXiv:2602.03793 (2026)
10. Clark, J., Mirchandani, S., Sadigh, D., Belkhale, S.: Action-free reasoning for policy generalization. arXiv preprint arXiv:2502.03729 (2025)
11. Dai, M., Liu, L., Bai, Y., Liu, Y., Wang, Z., Su, R., Chen, C., Lin, L., Wu, X.: Rover: Robot reward model as test-time verifier for vision-language-action model. arXiv preprint arXiv:2510.10975 (2025)
12. Fei, S., Wang, S., Shi, J., Dai, Z., Cai, J., Qian, P., Ji, L., He, X., Zhang, S., Fei, Z., et al.: Libero-plus: In-depth robustness analysis of vision-language-action models. arXiv preprint arXiv:2510.13626 (2025)
13. Huang, C.P., Wu, Y.H., Chen, M.H., Wang, Y.C.F., Yang, F.E.: Thinkact: Vision-language-action reasoning via reinforced visual latent planning. arXiv preprint arXiv:2507.16815 (2025)
14. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: π0. 5: a vision-language-action model with open-world generalization, 2025. URL https://arxiv. org/abs/2504.16054 **1**(2), 3
15. Jang, S., Kim, D., Kim, C., Kim, Y., Shin, J.: Verifier-free test-time sampling for vision language action models. arXiv preprint arXiv:2510.05681 (2025)
16. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
17. Kwok, J., Agia, C., Sinha, R., Foutter, M., Li, S., Stoica, I., Mirhoseini, A., Pavone, M.: Robomonkey: Scaling test-time sampling and verification for vision-language-action models. arXiv preprint arXiv:2506.17811 (2025)
18. Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L.J.V., Liu, A., Dziri, N., Lyu, S., et al.: Tulu 3: Pushing frontiers in open language model post-training. arXiv preprint arXiv:2411.15124 (2024)
19. Lee, J., Duan, J., Fang, H., Deng, Y., Liu, S., Li, B., Fang, B., Zhang, J., Wang, Y.R., Lee, S., et al.: Molmoact: Action reasoning models that can reason in space. arXiv preprint arXiv:2508.07917 (2025)
20. Li, D., Cao, S., Cao, C., Li, X., Tan, S., Keutzer, K., Xing, J., Gonzalez, J.E., Stoica, I.: S*: Test time scaling for code generation. arXiv preprint arXiv:2502.14382 (2025)

21. Li, P., Chen, Y., Wu, H., Ma, X., Wu, X., Huang, Y., Wang, L., Kong, T., Tan, T.: Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. arXiv preprint arXiv:2506.07961 (2025)
22. Li, P., Chen, Y., Xu, Y., Yang, J., Wu, X., Guo, J., Sun, N., Qian, L., Li, X., Xiao, X., et al.: Multi-view video diffusion policy: A 3d spatio-temporal-aware video action model. arXiv preprint arXiv:2604.03181 (2026)
23. Li, P., Wu, H., Huang, Y., Cheang, C., Wang, L., Kong, T.: Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy. IEEE Robotics and Automation Letters **10**(2), 1912–1919 (2025)
24. Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., Levine, S., Wu, J., Finn, C., Su, H., Vuong, Q., Xiao, T.: Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941 (2024)
25. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)
26. Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al.: Self-refine: Iterative refinement with self-feedback. Advances in Neural Information Processing Systems **36**, 46534–46594 (2023)
27. Qu, D., Song, H., Chen, Q., Chen, Z., Gao, X., Ye, X., Lv, Q., Shi, M., Ren, G., Ruan, C., et al.: Eo-1: Interleaved vision-text-action pretraining for general robot control. arXiv preprint arXiv:2508.21112 (2025)
28. Renze, M.: The effect of sampling temperature on problem solving in large language models. In: Findings of the association for computational linguistics: EMNLP 2024. pp. 7346–7356 (2024)
29. Song, H., Qu, D., Yao, Y., Chen, Q., Lv, Q., Tang, Y., Shi, M., Ren, G., Yao, M., Zhao, B., et al.: Hume: Introducing system-2 thinking in visual-language-action model. arXiv preprint arXiv:2505.21432 (2025)
30. Sun, N., Li, Y., Wang, C., Li, H., Liu, H.: Collabvla: Self-reflective vision-language-action model dreaming together with human. arXiv preprint arXiv:2509.14889 (2025)
31. Sun, N., Zhang, Y., Yang, Y., Zhao, W., Li, P., Guo, J., Song, W., Ding, P., Suo, R., Su, Y., et al.: Revisiting embodied chain-of-thought for generalizable robot manipulation. arXiv preprint arXiv:2606.03784 (2026)
32. Sun, Q., Hong, P., Pala, T.D., Toh, V., Tan, U., Ghosal, D., Poria, S., et al.: Emma-x: An embodied multimodal action model with grounded chain of thought and look-ahead spatial reasoning. arXiv preprint arXiv:2412.11974 (2024)
33. Sun, Z., Shen, S., Cao, S., Liu, H., Li, C., Shen, Y., Gan, C., Gui, L., Wang, Y.X., Yang, Y., et al.: Aligning large multimodal models with factually augmented rlhf. In: Findings of the Association for Computational Linguistics: ACL 2024. pp. 13088–13110 (2024)
34. Walke, H., Black, K., Lee, A., Kim, M.J., Du, M., Zheng, C., Zhao, T., Hansen-Estruch, P., Vuong, Q., He, A., Myers, V., Fang, K., Finn, C., Levine, S.: Bridgedata v2: A dataset for robot learning at scale. In: Conference on Robot Learning (CoRL) (2023)
35. Wang, J., Wang, J., Athiwaratkun, B., Zhang, C., Zou, J.: Mixture-of-agents enhances large language model capabilities. arXiv preprint arXiv:2406.04692 (2024)
36. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)

37. Xu, Y., Yang, J., Wang, X., Chen, Y., Zhu, Z., Fang, B., Huang, G., Chen, X., Ye, Y., Zhang, Q., et al.: Egodemogen: Novel egocentric demonstration generation enables viewpoint-robust manipulation. arXiv preprint arXiv:2509.22578 (2025)
38. Yang, J., Chen, Y., Xu, Y., Li, P., Wu, X., Wen, Z., Fang, B., Yu, T., Zhang, Z., Li, Y., et al.: Uaor: Uncertainty-aware observation reinjection for vision-language-action models. arXiv preprint arXiv:2602.18020 (2026)
39. Yang, S., Zhang, Y., He, H., Pan, L., Li, X., Bai, C., Li, X.: Steering vision-language-action models as anti-exploration: A test-time scaling approach. arXiv preprint arXiv:2512.02834 (2025)
40. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models. Advances in neural information processing systems **36**, 11809–11822 (2023)
41. Ye, W., Liu, Z., Gui, Y., Yuan, T., Su, Y., Fang, B., Zhao, C., Liu, Q., Wang, L.: Genpilot: A multi-agent system for test-time prompt optimization in image generation. arXiv preprint arXiv:2510.07217 (2025)
42. Yu, P., Xu, J., Weston, J., Kulikov, I.: Distilling system 2 into system 1. arXiv preprint arXiv:2407.06023 (2024)
43. Yuan, T., Guan, B., Ye, W., Tian, Z., Yang, Y., Zhou, W., Li, Z., Huang, Y., Wang, P., Zhao, C., et al.: Unibyd: A unified framework for learning robotic manipulation across embodiments beyond imitation of human demonstrations. arXiv preprint arXiv:2512.11609 (2025)
44. Yuan, Y., Cui, H., Huang, Y., Chen, Y., Ni, F., Dong, Z., Li, P., Zheng, Y., Hao, J.: Embodied-r1: Reinforced embodied reasoning for general robotic manipulation. arXiv preprint arXiv:2508.13998 (2025)
45. Zawalski, M., Chen, W., Pertsch, K., Mees, O., Finn, C., Levine, S.: Robotic control via embodied chain-of-thought reasoning. arXiv preprint arXiv:2407.08693 (2024)
46. Zhai, A., Liu, B., Fang, B., Cai, C., Ma, E., Yin, E., Wang, H., Zhou, H., Wang, J., Shi, L., et al.: Igniting vlms toward the embodied space. arXiv preprint arXiv:2509.11766 (2025)
47. Zhang, S., Xu, Z., Liu, P., Yu, X., Li, Y., Gao, Q., Fei, Z., Yin, Z., Wu, Z., Jiang, Y.G., et al.: Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11142–11152 (2025)
48. Zhong, Y., Bai, F., Cai, S., Huang, X., Chen, Z., Zhang, X., Wang, Y., Guo, S., Guan, T., Lui, K.N., et al.: A survey on vision-language-action models: An action tokenization perspective. arXiv preprint arXiv:2507.01925 (2025)
49. Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., et al.: Least-to-most prompting enables complex reasoning in large language models. arXiv preprint arXiv:2205.10625 (2022)
50. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023)