TCoT: Trajectory Chain-of-Thoughts for Robotic Manipulation with Failure Recovery in Vision-Language-Action Model

Xiang Li[1,2], **Ya-Li Li**[1,2*], **Yuan Wang**[1,2], **Huaqiang Wang**[1,2], **Shengjin Wang**[1,2]

[1]Department of Electronic Engineering , Tsinghua University, China
[2]Beijing National Research Center for Information Science and Technology (BNRist), China
l-xiang24@mails.tsinghua.edu.cn {liyali13, wgsgj}@tsinghua.edu.cn

Abstract

Recent advances in vision-language-action (VLA) models have demonstrated impressive generalization for robotic manipulation. However, these models often operate by directly mapping visual and linguistic inputs to subsequent actions, lacking intermediate task planning, along with failure detection and recovery ability. These limitations prevent them from effectively decomposing complex tasks, recognizing problems, and correcting erroneous actions, ultimately resulting in complete task failure. This significantly hinders their ability to perform long-horizon tasks and generalization ability. To this end, we introduce **TCoT**: **T**rajectory **C**hain-**o**f-**T**hought, a unified VLA framework that enhances this direct mapping with trajectory planning as well as failure detection and recovery. TCoT leverages hierarchy trajectories as a precise and compact representation of CoT reasoning for manipulation: global planning provides a high-level, goal-oriented trajectory to guide the robot toward its task objective, while local planning focuses on real-time adjustments to address dynamic changes. Moreover, we designed the Global-Local Switching Recovery algorithm that detects and effectively recovers from failures. Experimental results reveal that TCoT surpasses the state-of-the-art methods across both real and simulated scenarios and exhibits superior generalization capabilities.

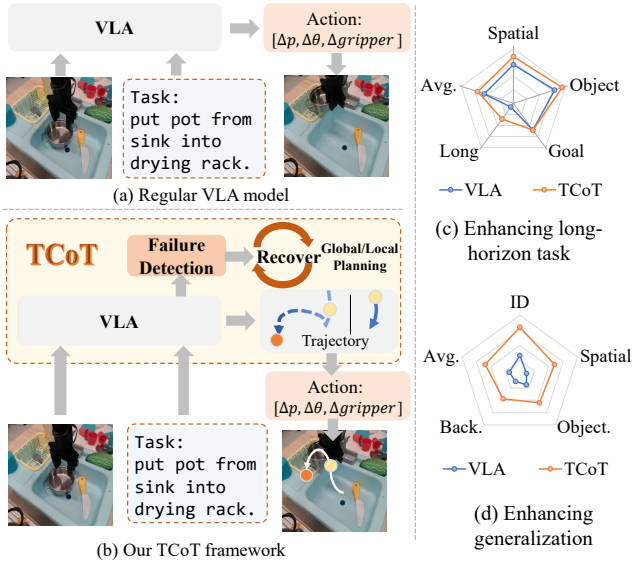

Figure 1: (a) Regular VLA models directly map inputs to actions, (b) TCoT introduces a global-local trajectory as intermediate task planning layer.

Introduction

Recent advances in large language models (LLMs)(Achiam et al. 2023; Touvron et al. 2023) and vision-language models (VLMs)(Liu et al. 2024a; Li et al. 2023) have demonstrated remarkable capabilities in perception, understanding, and reasoning over complex data, laying a strong foundation for the development of generalized robotic policies. Vison-language-action models (VLAs) (Zitkovich et al. 2023; Kim et al. 2024; Li et al. 2024a; Brohan et al. 2023a) have extended pretrained VLMs to robotic manipulation tasks by fine-tuning on large-scale cross-embodiment robotic datasets (O'Neill et al. 2024; Khazatsky et al. 2024). These generalist robot policies exhibit flexibility in adapting to changes in the environment, objects, and instructions, enabling efficient task adjustment through fine-tuning.

Despite the progress made, existing VLA models often focus on architecture design (Liu et al. 2025; Black et al. 2024a; Wen et al. 2025) and computational efficiency (Bu et al. 2024a), while overlooking two key challenges: (1) the absence of intermediate task planning between high-level instructions and low-level actions, and (2) the inability to detect and recover from failures. These issues remain inadequately addressed and require further exploration:

The Absence of Intermediate Task Planning: Current VLA models (Black et al. 2024a; Liu et al. 2025; Octo Model Team et al. 2024) typically predict robot actions for the next time step directly from vision and language inputs, as shown in Fig. 1 (a) However, this approach often neglects intermediate task planning and reasoning, limiting its effectiveness, especially in scenarios requiring long-horizon manipulation. When humans perform long-horizon tasks, they first plan and decompose the entire task into smaller, manageable steps and then execute them sequentially. Moreover, the lack of task planning limits the model's ability to retain historical context, such as past actions and trajectories. Thus, it may repeat completed actions due to poor progress estimation,

*Corresponding author

hindering performance on long-horizon tasks.

Inability to Detect and Recover from Failures: Task failures are common in robot-environment interactions, making detection and recovery crucial for success. However, current VLA models (Kim et al. 2024; Liu et al. 2025; Black et al. 2024a) lack this capability, as they are trained primarily on successful demonstrations and ignore failure cases (Dai et al. 2024). While VLAs may exhibit implicit retry behavior, such retries are often limited, unguided, and unreliable. In contrast, an explicit failure recovery mechanism enables structured detection and targeted re-execution, allowing the robot to identify errors, replan, and adapt strategies.

To address these challenges, we introduced TCoT (Fig. 1), a unified VLA framework that enhances the model with hierarchy trajectory planning, as well as failure detection and recovery. Trajectories provide a concrete intermediate task planning that bridges the gap between task understanding and execution, leading to precise and goal-directed actions. In long-horizon tasks, discrete waypoints in trajectory planning decompose and structure the task, representing the target positions the end-effector should reach at each stage. These waypoints enable the robot to break down complex tasks into manageable steps and ensuring smooth progress toward the final goal. We propose a hierarchy planning strategy: (1) global planning provides a goal-oriented long-horizon trajectory that ensures the robot moves toward the task objective, while 2) local planning allows for real-time adjustments to handle dynamic changes with short-horizon trajectory. Moreover, by using predicted trajectories as reference, the model benefits from both future and past way-points, enhancing spatial-temporal reasoning ability. Additionally, we propose a Global Local Switching Recovery (GLSR) algorithm that enables the framework to identify failures and switch strategies when failure occurs. Specifically, we propose a scalable data generation pipeline that augments successful expert demonstrations with failure cases and task completion annotations for training. The model then gains the ability to detect failures by learning from both successful and unsuccessful scenarios. With the support of the proposed hierarchical planning strategy, the model can replan or switch strategies upon failures, improving robustness and success rates, particularly in complex, long-horizon tasks.

We present the first attempt to integrate global-local trajectory planning, action generation, and failure recovery into a unified VLA model, allowing it to handle diverse prompts and generate varied responses. Extensive experiments on both simulated and real-world datasets demonstrate that TCoT surpasses state-of-the-art methods and enhances knowledge sharing between tasks, highlighting its superior performance and robustness. In summary, our contributions include:

- We introduce a unified VLA framework for robotic manipulation, TCoT, capable of action generation, intermediate task planning, as well as failure detection and recovery.
- We propose a trajectory-based CoT reasoning for robot manipulation. It combines global planning for task-level alignment and local planning for action-level adaptability,
- We propose a GLSR algorithm that detects failure and effectively recovers from it, enhancing robustness and success rates in complex, long-horizon tasks.
- Comprehensive experiments have been conducted in both real-world and simulated environments, demonstrating the superior performance and generalizability.

Related Works

Language-Conditioned Visual Manipulation

In language-conditioned manipulation, robots follow language instructions to complete tasks. Existing approaches can be broadly categorized into three paradigms: LLM as planner, specialist, and generalist policies.

In the first paradigm, LLMs are used as planners (Liang et al. 2023; Brohan et al. 2023b; Huang et al. 2024a), where they decompose complex tasks into smaller, more manageable subtasks, each of which is then grounded to a predefined set of skills or policies. However, the textual plans or constraints generated by LLMs or VLMs often suffer from redundancy and unnecessary complexity (Huang et al. 2023a, 2024b), which can make it difficult to describe the scene accurately and concisely. The second paradigm focuses on specialist policies (Fu, Zhao, and Finn 2024; Chi et al. 2024; Ma et al. 2024; Ryu et al. 2024), which are designed to perform a narrow set of tasks. While these policies can achieve high performance within their training context, they cannot generalize beyond the training data.

To address these limitations, recent work has focused on generalist policies(Brohan et al. 2023a; Zitkovich et al. 2023), which leverage extensive cross-embodiment datasets(O'Neill et al. 2024; Walke et al. 2023) and VLAs or by incorporating an action expert (Black et al. 2024a; Liu et al. 2025; Li et al. 2024a; Wang et al. 2025), However, existing VLAs typically predict actions directly from input without fully utilizing the reasoning capabilities of the underlying vision-language models. In contrast, our approach, TCoT, introduces both global and local trajectories as intermediate planning steps before action prediction, enabling a more structured and interpretable reasoning process.

Reasoning for Manipulation Policies

LLMs have demonstrated impressive text-based reasoning capabilities for embodied agents, ranging from task planning (Song et al. 2023; Liang et al. 2023; Sermanet et al. 2024) to embodied feedback (Brohan et al. 2023b; Huang et al. 2023b). However, textual reasoning alone is insufficient to capture the dynamic and context-rich information inherent in environments. Thus, recent works have explored embodied reasoning (Zawalski et al. 2024), which incorporate elements such as gripper position (Li et al. 2024b), low-level motions (Belkhale et al. 2024), and trajectories (Gu et al. 2024; Wen et al. 2024; Zheng et al. 2024; Zhang et al. 2024). These elements are grounded in visual observations of the scene and the agent's state, offering a more comprehensive understanding of the manipulation process.

An alternative approach leverages generative models as visual planners to generate goal images (Ni et al. 2024a; Bu et al. 2024b; Black et al. 2024b; Ni et al. 2024b; Zhao et al. 2025), which are then used by a low-level controller to guide the robot's action. However, the generated images or

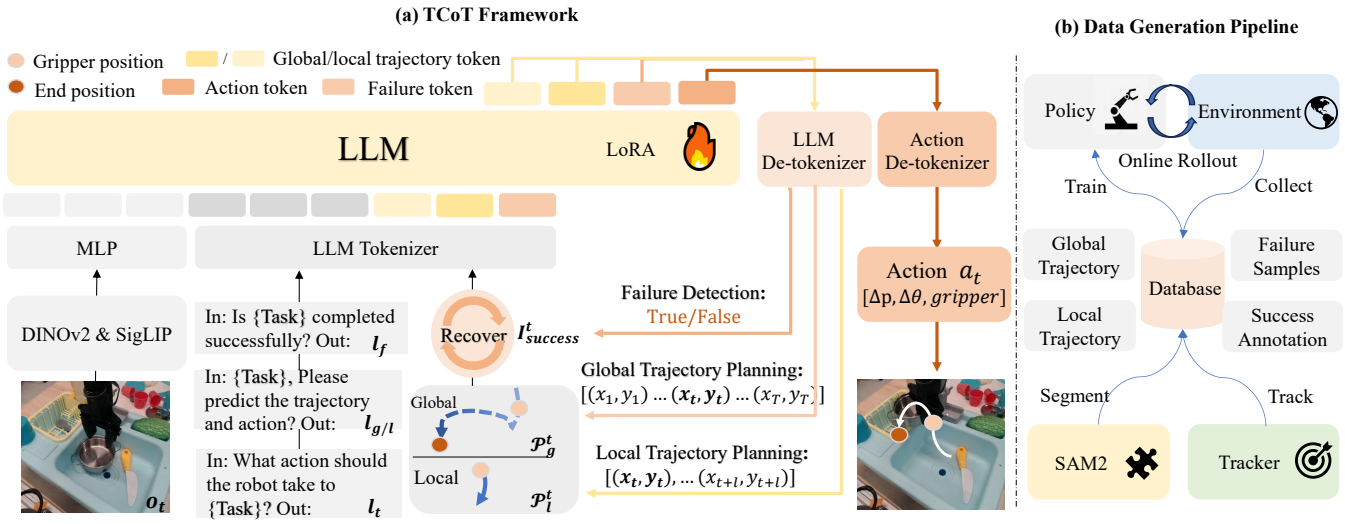

Figure 2: (a) The overall framework of TCoT (Sec.). It is composed of three key parts: a pre-trained VLA backbone, the trajectory-based reasoning (Sec.), and the failure detection and recovery module (Sec.). This integrated design empower TCoT to achieve precise and robust manipulation performance, driven by (b) the data generation and annotation pipeline.

videos often lack geometric and motion consistency, which can undermine the performance of the control policy.

Our approach enhances VLA models by enabling them to perform CoT reasoning at a trajectory level, establishing an intermediate layer between the high-level tasks and the low-level actions. By generating 2D trajectories for the end-effector before action execution, our method ensures more accurate and contextually informed manipulation, improving the model's overall performance.

Failure Detection and Recovery

Failure detection and recovery are essential for robust robot execution. Existing approaches (Luo 2024; Guo et al. 2023; Liu et al. 2024b; Zhou et al. 2025) typically rely on separate large vision-language models (VLMs) as external failure detectors, often using visual question answering or constraint-based reasoning to identify and explain errors. While these methods enable post-hoc self-reflection (Liu, Bahety, and Song 2023; Liu et al. 2024b), they introduce additional computational overhead. More critically, failure detection and recovery are decoupled, detection may occur externally, while recovery still depends on separate control mechanisms, limiting response coherence and efficiency. In contrast, our proposed GLSR algorithm integrates failure detection and recovery within a unified VLA model. When failures arise, the model explicitly switching strategies—from global to local planning—enabling re-execution of the failed task segment. This design improves task success rates and enhances adaptability in dynamic environments.

Method

The Proposed TCoT Framework

As illustrated in Fig. 2, our TCoT framework is built upon a pre-trained VLA model as its backbone. The primary objective of the VLA model is to generate robot actions for a variety of tasks by integrating visual observations and task instructions. Formally, at each time step t, given a visual observation o_t and a language instruction l_t, the VLA model π predicts the corresponding action a_t for robot execution: $a_t = \pi(o_t, l_t)$. In the context of robotic manipulation tasks, the action a_t is defined as the delta movement of the end-effector, represented as a 7-degree-of-freedom vector: $[\Delta p, \Delta\theta, gripper]$, where $\Delta p = [\Delta p_x, \Delta p_y, \Delta p_z]$ denotes the relative translation and $\Delta\theta = [\Delta\theta_x, \Delta\theta_y, \Delta\theta_z]$ represents the relative rotation. The gripper indicates the open/-closed state of the gripper.

Specifically, TCoT is a general framework, and we instantiate our TCoT algorithm using the OpenVLA (Kim et al. 2024) as the VLA backbone, which is trained using a standard next-token prediction objective, minimizing the cross-entropy loss on the predicted action tokens:

$$\mathcal{L}_a = -\sum_{S\in\mathcal{D}}\sum_{t}^{T} log\ p(a_t|o_t, l_t; \pi) \quad (1)$$

where $\mathcal{S} = \{o_1, a_1, \dots, o_T, a_T, l\}$ represent an expert demonstration trajectory, and $\mathcal{D} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_N\}$ denote the training dataset, comprising N such demonstrations.

To enable the VLA model with trajectory planning capabilities, we designed an automated trajectory generation pipeline to produce global and local 2D trajectories from existing demonstration data. Subsequently, we fine-tuned the model using prompt templates $l_{g/l}$. This allows the model to perform trajectory planning: $\mathcal{P}_{g/l} = \pi_1(o_1, l_{g/l})$, breaking down complex tasks into sequential waypoints, which are used as reasoning prompts for more accurate and contextually informed action generation: $a_t = \pi_1(o_t, l_{g/l} \mid \mathcal{P}_{g/l})$.

To enable failure detection and recovery, we design a GLSR method. During the model rollout process, we collect

data and corresponding labels from both successful and unsuccessful scenarios. Through instruction tuning with prompt template l_f, the model learns to recognize failure from expected task progress: $I_{\text{success}} = \pi_1(o_t, l_f)$. When failures are detected during actual inference, the model automatically switches to a more appropriate planning strategy, thereby recovering from them. Our TCoT framework integrates global-local trajectory planning, action generation, as well as failure detection and recovery into a unified VLA model through a two-stage instruction tuning, allowing it to handle diverse prompts and generate varied responses.

Trajectory-based CoT Reasoning

In this section, we explore the details of the data generation pipeline and the training for hierarchy trajectory-based chain-of-thought reasoning before generating action.

Data Generation Pipeline. We propose an automated trajectory generation pipeline to produce global and local 2D trajectories from existing demonstrations. In this paper, trajectory refers to the planned path that a robot's end-effector follows in the image coordinate system over time to accomplish a specific task. We adopt the off-the-shelf tracking model proposed in (Karaev et al. 2024) to track the gripper's trajectory in the demonstration video. Formally, given the sequence of images $\{o_1, \ldots, o_T\}$ from S and the gripper's initial position $p_1 = (x, y)$ in the image coordinate system, the tracker predicts the 2D position of the target point in subsequent frames $p_t, t \in [2, T]$. To obtain the initial gripper position p_1, we utilize an open-vocabulary detector Φ_{d} (Minderer, Gritsenko, and Houlsby 2024) to predict the bounding box of gripper $b_1 = \Phi_{\text{d}}(o_1, v)$, where v is the text description of the gripper. The detection results are then used as input to a segmentation model Φ_{s} (Ravi et al. 2024), which generates a segmentation mask. The gripper's initial position is then computed as the centroid of the mask:

$$p_1 = \text{Centroid}(M_1), \quad M_1 = \Phi_{\text{s}}(o_1, b_1) \tag{2}$$

We employ this detection-then-segment method only for the first frame, as the mask may be inaccurate in subsequent frames. Additionally, in real-world datasets, the gripper may become occluded or rotate, causing the initially set tracking points to disappear, leading to trajectory interruptions. To address this, we use the promptable video segmentation model SAM2 (Ravi et al. 2024) to infer the gripper's position when the tracker signals a missing point. This model takes both positive and negative points as prompts for fine-grained segmentation while considering the temporal state s. The gripper's position at time t is then given by:

$$p_t = \text{Centroid}(\Phi_{\text{s}}(o_t, p_p, p_n, s_{t-1})), \quad t \in [2, T] \tag{3}$$

where p_p consists of positive points, including p_1 and other sampled points inside the mask M_1, while p_n denotes the negative points sampled around p_1 but outside M_1. Finally, we obtain the trajectory $\mathcal{P}' = \{p_1, p_2, \ldots, p_T\}$. To ensure the continuity and consistency of the tracked points, we filter out outliers in $\mathcal{P}'$ with the RANSAC (Fischler and Bolles 1981) algorithm. Specifically, we extract the 3D position of the gripper from the robot state to construct the trajectory in 3D space, denoted as $X \in \mathbb{R}^{N\times 3}$. Then a mapping from 3D trajectories $\mathcal{X}$ to 2D trajectories $\mathcal{P}'$ is constructed, thereby filtering out the noise points to get the smooth trajectory $\mathcal{P}$.

Hierarchical Trajectory Planning. To fully leverage the trajectory $\mathcal{P}$, TCoT employs a hierarchical planning strategy: 1) global trajectory for task-oriented, long-horizon planning, and 2) local trajectory for fine-grained, short-horizon planning. The global trajectory decomposes complex tasks into a series of waypoints, providing a comprehensive trajectory plan as a reference for the manipulation task. We first determine two scenario-aware hyperparameters: maximum trajectory length L_g and temporal sampling interval n_g. At each timestep t, the global trajectory $\mathcal{P}_g^t$ is constructed through symmetric windowed sampling:

$$\mathcal{P}_g^t = \left\{p_{t-n_g\lfloor L_g/2\rfloor}, \ \ldots, \ p_t, \ \ldots, \ p_{t+n_g\lfloor L_g/2\rfloor}\right\} \tag{4}$$

The future waypoints $\{p_i | i > t\}$ ensure smooth, goal-directed task planning, while the past waypoints $\{p_i | i < t\}$ preserve the historical context information, which helps the model to access task progress effectively. Also, using trajectories as the intermediate layer between high-level task understanding and low-level execution leads to much better data sharing between different tasks, thus improving generalization in multi-task datasets. While global planning establishes task context, predicting full trajectory at every timestep is computationally prohibitive. When changes occur during task execution, the global trajectory cannot be updated in real time, therefore, we introduce a complementary local trajectory mechanism with higher updating frequency focused on dynamic task planning and precise manipulation ability. The local trajectory $\mathcal{P}_l^t$ includes the planning for a short period starting from the current position, featuring a denser distribution of waypoints:

$$\mathcal{P}_l^t = \{p_t, \ p_{t+n_l}, \ \ldots, \ p_{t+n_l L_l}\} \tag{5}$$

where L_l and n_l denote the local trajectory length and sampling interval ($n_l < n_g$). The shorter horizon enables the local trajectory to update at a higher frequency, facilitating more dynamic task execution. The denser distribution of waypoints provides finer-grained spatial guidance for low-level motion, allowing for more precise and delicate manipulation tasks. This hierarchy planning achieves a balance between computational efficiency and motion precision.

Trajectory-based Chain-of-Thought Training. To enhance the VLA model with trajectory-based reasoning capabilities, we introduce the first stage instruction tuning that integrates hierarchical trajectory planning and action generation. After extracting both global and local trajectories $\mathcal{T} = \{\mathcal{P}_g, \mathcal{P}_l\}$ from multi-task demonstrations, we reformulate the training objective to explicitly condition action prediction on planned trajectories based on the following prompt template $l_{g/l}$:

> % Trajectory-based Reasoning Prompt Templates
> **Instruction**: "Predict the gripper's global/local trajectory and the action to complete the {TASK}."
> **Answer**: "Trajectory: $\mathcal{P}_g$/$\mathcal{P}_l$ Action: $\mathcal{A}$".

To optimize the parameters of model π_1 efficiently, we apply LoRA (Hu et al. 2022) for fine-tuning, introducing an

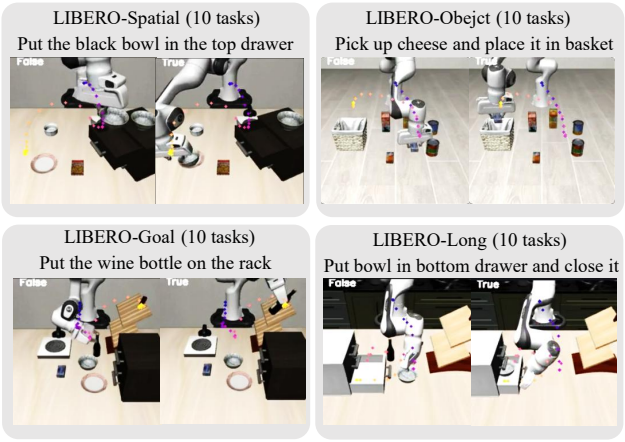

Figure 3: Visualization of the four LIBERO task suites. Each tasks suites contain 10 tasks and 500 demonstrations.

additional objective that jointly predicts both trajectory and action tokens:

$$\mathcal{L}_p = -\sum_{S\in\mathcal{D}}\sum_{t}^{T} log\ p(\mathcal{P}_i^t, a_t|o_t, l_t; \pi_1),\ i \in \{l, g\} \quad (6)$$

This contrasts with conventional VLA approaches (Zitkovich et al. 2023), which directly map observations to actions a_t without explicit motion planning as an intermediate step. In contrast, our formulation introduces structured trajectory reasoning, enabling both global and local inference modes.

Algorithm 1: Failure Recovery with Global-Local Switching

Require: Policy π_1, task instructions l, visual input o, time step for failure determination t_f, local planning frequency t_l, global-local planning prompt l_g, l_l, failure detection prompt l_f.
Ensure: Action a_t

$a_1, \mathcal{P}_g^1 \leftarrow \pi_1(o_1, l_g)$ ▷ Generate global traj.
for $t = 2$ **to** t_f **do**
  $a_t \leftarrow \pi_1(o_t, l_g \mid \mathcal{P}_g^t)$ ▷ Follow frozen global traj.
end for
for $t > t_f$ **do**
  $I_{\text{success}} \leftarrow \pi_1(o_t, l_f)$ ▷ Failure detection
  if ($t \mod t_l = 0$ and not I_{success}) **then**
    $a_t, \mathcal{P}_l^t \leftarrow \pi_1(o_t, l_l)$ ▷ Replan with local traj.
  else
    $a_t \leftarrow \pi_1(o_t, l \mid \mathcal{P}_l^t)$ ▷ Follow local traj.
  end if
end for

Global-Local Switching Recovery

Failures are an inherent challenge in robot-environment interactions, making failure detection and recovery essential for task completion. We propose to integrate failure detection and recovery directly into the VLA model.

Failure Detection. Conventional VLA architectures lack built-in failure detection mechanisms, which constrain their practical deployment. Thus, we deploy the policy π_1 in the simulated or real-world environment and let the VLA model iteratively interact with the environment, collecting an online dataset that contains both successful and failed samples: $\mathcal{S}_{online} = \{o_1, a_1, \ldots, o_T, a_T, l\}$. The dataset is accompanied by a binary success indicator $I_{\text{success}} \in \{\text{True}, \text{False}\}$. The criteria for determining success vary across environments: in simulation-based evaluations, predefined physical simulator metrics serve as the basis, whereas real-world implementations rely on operator-mediated validation. Specifically, tasks that exceed temporal constraints or exhibit operational failures are manually terminated and annotated. To enable direct failure state prediction within the VLA architecture, we train the model with the following template l_f:

Failure Detection Prompt Template
Instruction: "Determine whether the {TASK} has been successfully completed based on visual evidence."
Answer: "$\mathcal{I}_{success}$"

Then, we propose the following failure detection objective to predict the failure token:

$$\mathcal{L}_f = -\sum_{S\in\mathcal{D}_f}\sum_{t=t_f}^{T} \log p(I_{\text{success}}^t|o_t, l_t; \pi_1) \quad (7)$$

where $\mathcal{D}_f = \{S_{online}^i\}_{i=1}^N$ represents the dataset collected online and t_f is the time step for failure determination. Based on these datasets $\mathcal{D}, \mathcal{D}_{online}$ and annotations $\mathcal{P}_{g/l}, \mathcal{I}_{success}$, we adopt LoRA (Hu et al. 2022) in the second stage instruction finetuning, integrating multiple learning objectives:

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}_p + \mathcal{L}_f \quad (8)$$

Therefore, TCoT is capable of trajectory planning, failure detection, and action generation simultaneously through different prompts. We optimize performance by adjusting the proportion of the three types of data rather than modifying the loss weights (refer to supplementary materials).

Failure Recovery. Policy π_1 integrates action generation, trajectory planning, and failure detection into a unified framework. These capabilities can be invoked dynamically through different prompts during inference, enabling automated failure recovery. We detail the GLSR in Algorithm 1. Given the task instructions l and the visual input o, π_1 first predicts the global trajectory and action for the initial timestep. To enhance inference efficiency, subsequent actions are generated while conditioning on the frozen global trajectory. When t reaches t_f, the model will actively detect I_{success}^t to determine task completion status. If the task is deemed unsuccessful, the control strategy transitions to action generation based on local trajectory planning. Compared to the implicit retry behavior of standard VLA models—often repeatedly fail in the same way—GLSR provides explicit and structured recovery by detecting failures and switching to an alternative planning. This enables TCoT to explore different solution paths, improving robustness and adaptability in complex tasks.

Experiments

Datasets. We perform simulated evaluations on the LIBERO benchmark (Liu et al. 2023), which includes four task suites

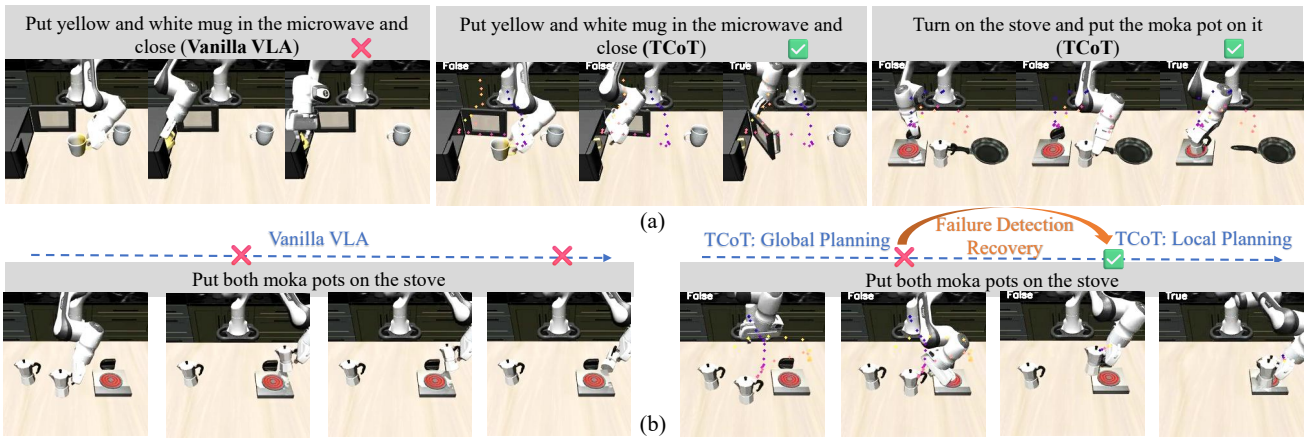

Figure 4: Qualitative Results of TCoT vs. Vanilla VLA on LIBERO (a) illustrates the global trajectory planning (colored) capability of TCoT, enabling precise and task-oriented execution. (b) highlights the failure detection and recovery mechanism.

Model	Publication	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Avg. SR ↑	Rank ↓
ECoT	CoRL2024	$0.840^{\pm.011}$	$0.870^{\pm.012}$	$0.785^{\pm.015}$	$0.550^{\pm.008}$	$0.761^{\pm.012}$	7
OpenVLA	CoRL2024	$0.847^{\pm.009}$	$0.884^{\pm.008}$	$0.792^{\pm.010}$	$0.537^{\pm.013}$	$0.765^{\pm.006}$	5
OpenVLA-Multi	CoRL2024	$0.840^{\pm.005}$	$0.860^{\pm.007}$	$0.700^{\pm.012}$	$0.440^{\pm.011}$	$0.710^{\pm.006}$	6
GRAPE	ICLR2025	$0.876^{\pm.004}$	$0.912^{\pm.007}$	$0.822^{\pm.009}$	$0.558^{\pm.011}$	$0.792^{\pm.008}$	4
CoT-VLA	CVPR2025	$0.875^{\pm.006}$	$0.916^{\pm.014}$	$\mathbf{0.876}^{\pm.005}$	$0.690^{\pm.006}$	$0.811^{\pm.008}$	3
TCoT	-	$\mathbf{0.910}^{\pm.002}$	$\mathbf{0.948}^{\pm.008}$	$0.788^{\pm.010}$	$0.668^{\pm.012}$	$0.829^{\pm.007}$	2
TCoT-Multi	-	$0.900^{\pm.004}$	$0.870^{\pm.009}$	$0.850^{\pm.007}$	$\mathbf{0.710}^{\pm.011}$	$\mathbf{0.833}^{\pm.005}$	**1**

Table 1: Quantitative comparison with SOTA methods with discrete action modeling on LIBERO benchmark. 'SR' indicates the success rates. ± represents the standard error. 'Multi' represents cross-task settings.

Method	Publication	Spatial	Object	Goal	Long	Avg.SR
π_0	Arxiv2024	96.8	98.8	**95.8**	85.2	94.2
UniVLA	RSS2025	96.5	96.8	95.6	92.0	95.2
OpenVLA-OFT	Arxiv2025	96.9	98.1	95.5	91.1	95.4
TCoT-OFT	-	**97.5**	**98.8**	95.0	**93.5**	**96.2**

Table 2: Comparison with SOTA methods with continuous action modeling on LIBERO benchmark.

designed for robotic manipulation. These suites (see in Fig. 3) each consist of 10 tasks, with 50 demonstrations per task collected via human teleoperation for evaluating the ability to transfer knowledge related to spatial relationships, object manipulation, task-specific goals, and long-horizon execution. For real-world evaluations, we utilize the AIRBOT Arm. As shown in Fig. 5, we designed 7 tasks involving tool use, fine manipulation, cluttered objects, and multiple steps.

Simulated Evaluation

Quantitative Results. As shown in Tab. 1, we evaluate TCoT against several state-of-the-art approaches with discrete action modeling, including OpenVLA, ECoT, GRAPE and CoT-VLA (Kim et al. 2024; Zawalski et al. 2024; Zhang et al.; Zhao et al. 2025). In the single-task setting, TCoT achieves the highest average success rate (SR) across all models, outperforming SOTA method by 1.8%. Notably, TCoT shows significant improvement cmpared to baseline on more challenging LIBERO-Long task suites, highlighting the effectiveness of TCoT in spatial reasoning and long-horizon planning. We also evaluate TCoT under multi-task setting, which trains a generalist model across all four tasks. The results indicate that baseline OpenVLA experiences a performance decline in multi-task settings, whereas TCoT achieves further improvements in more complex and long-horizon tasks, with its average performance surpassing that of single-task training (83.3% vs. 82.9%). This demonstrates that our approach effectively facilitates knowledge sharing between tasks, primarily due to its use of trajectories as an intermediate representation between instructions and actions. When two tasks share similar trajectory segments, the model can learn the abstraction of them from the overlapping trajectories, thereby enhancing both data utilization and generalization.

In Tab 2, we instantiate the TCoT framework using OpenVLA-OFT (Kim, Finn, and Liang 2025) and compare it with VLA models based on continuous action modeling (Black et al. 2024a; Wang et al. 2025). The results demonstrate that our approach achieves consistent improvements over the baseline, particularly outperforming state-of-the-art methods on long-horizon tasks.

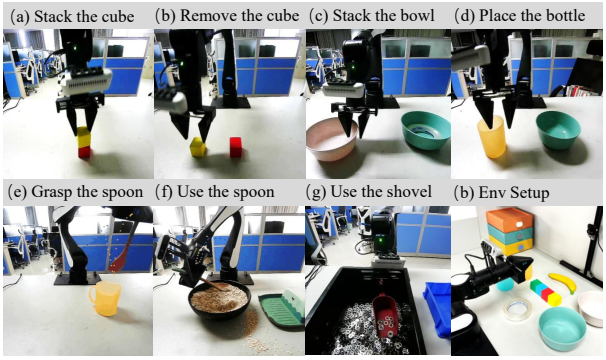

Figure 5: (a-g) Visualization of the 7 real-world tasks.

Qualitative Results. As illustrated in Fig. 4, (a) in the task 'Put the mug in the microwave and close,' TCoT generates a global planning trajectory at the initial step and uses it as a chain-of-thought prompt to predict subsequent actions. This intermediate task planning enables precise, goal-oriented execution, while the vanilla VLA (Kim et al. 2024) model becomes stuck at the microwave door. (b) In the task 'Put both moka pots on the stove,' both TCoT and vanilla VLA initially fail. However, our method identifies the failure, switches strategies, and recovers effectively, whereas the vanilla VLA model fails to respond and gradually adopts abnormal poses.

Real-World Evaluation

Real-world evaluation are presented in Table 3. OpenVLA (Kim et al. 2024) suffers from repetitive motions and often becomes confused about its progress. This issue is particularly pronounced in scenarios with repetitive trajectories, which can be attributed to a lack of global planning or historical context. In contrast, TCoT leverages predicted trajectories as prompts, effectively incorporating both past and future movement information. This approach significantly enhances spatial-temporal reasoning, improving manipulation task performance. Overall, TCoT achieves a 28% improvement compared to the baseline. Here, we also observed that TCoT effectively facilitates knowledge sharing across different tasks: the results show that multi-task training only improves OpenVLA by 4%. In comparison, TCoT achieves a 8% improvement, nearly doubling the gains observed in the other methods, highlighting the advantages of proposed trajectory-based reasoning. In the tool using tasks, model is required to transfer cereal or gaskets to a target location, involving complex tool use and multi-step, long-horizon operations. The results show that TCoT benefits from trajectory-level planning, achieving greater improvements over baseline.

Ablation Study

As shown in Table 4, we conduct an ablation study to evaluate the impact of different components on the success rates in the LIBERO benchmark. Compared to the baseline, using either global or local trajectory planning alone results in performance improvement of 2.1% and 1.1% respectively. However, enabling failure detection and allowing the robot to recover through GLSR, the average success rate increases significantly by 6.4%. This result underscores the effectiveness of our TCoT framework and highlights the pivotal role of failure detection and recovery in enhancing manipulation task performance. Unlike implicit retry behaviors often exhibited by VLA models, our explicit failure recovery mechanism enables structured re-planning at the trajectory level, allowing the system to more reliably correct diverse failure modes and improve robustness in long-horizon tasks. Refer to supplementary for more comprehensive ablations.

Setting	Method	Pick		Stack		Use			Avg.
		cube	cup	spoon	bowl	cube	spoon	shovel	
Single	Octo	0.20	0.10	0.40	0.35	0.00	0.55	0.25	0.26
	OpenVLA	0.50	0.25	0.45	0.40	0.20	0.60	0.35	0.39
	TCoT	**0.70**	**0.50**	**0.75**	**0.90**	**0.35**	**0.85**	**0.60**	**0.67**
Multi	Octo	0.25	0.10	0.35	0.40	0.05	0.45	0.30	0.27
	OpenVLA	0.55	0.25	0.45	0.45	0.30	0.60	0.40	0.43
	TCoT	**0.80**	**0.65**	**0.85**	**0.90**	**0.50**	**0.85**	**0.70**	**0.75**

Table 3: Real-world evaluation on AIRBOT.

Method	Global Traj.	Local Traj.	GLSR	Average SR
OpenVLA	-	-	-	$0.765^{\pm.006}$
TCoT	✓	-	-	$0.786^{\pm.005}$
	-	✓	-	$0.773^{\pm.009}$
	✓	✓	✓	$\mathbf{0.829}^{\pm.007}$

Table 4: Ablation study on the effects of different components on the average success rate on the LIBERO benchmark.

Conclusion

In this paper, we present TCoT, a novel framework that enhances vision-language-action (VLA) models for robotic manipulation by integrating hierarchical trajectory planning with failure detection and recovery. TCoT addresses two key limitations of existing VLA models: the absence of intermediate task planning and the lack of explicit failure recovery. By generating both global and local trajectories, TCoT enables trajectory-level chain-of-thought reasoning, while the proposed GLSR algorithm allows adaptive strategy switching during execution. Extensive experiments on the LIBERO benchmark and real-world tasks show that TCoT significantly outperforms state-of-the-art methods, demonstrating strong generalization across diverse environments. In addition, TCoT facilitates knowledge sharing across tasks, a critical capability for building general-purpose robotic policies.

Acknowledgments

This work is supported by the research fund under Grant No. 20242910035 from the Tsinghua University-Jiangsu CRRC Digital Technology Co.,Ltd. Joint Research Center for Data Driven Intelligence of Industry.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Belkhale, S.; Ding, T.; Xiao, T.; Sermanet, P.; Vuong, Q.; Tompson, J.; Chebotar, Y.; Dwibedi, D.; and Sadigh, D. 2024. Rt-h: Action hierarchies using language. In *Proceedings of Robotics: Science and Systems*.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024a. pi0 : A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*.

Black, K.; Nakamoto, M.; Atreya, P.; Walke, H. R.; Finn, C.; Kumar, A.; and Levine, S. 2024b. Zero-Shot Robotic Manipulation with Pre-Trained Image-Editing Diffusion Models. In *International Conference on Learning Representations*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2023a. RT-1: Robotics Transformer for Real-World Control at Scale. In *Robotics: Science and Systems*.

Brohan, A.; Chebotar, Y.; Finn, C.; Hausman, K.; Herzog, A.; Ho, D.; Ibarz, J.; Irpan, A.; Jang, E.; Julian, R.; et al. 2023b. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, 287–318.

Bu, Q.; Li, H.; Chen, L.; Cai, J.; Zeng, J.; Cui, H.; Yao, M.; and Qiao, Y. 2024a. Towards Synergistic, Generalized, and Efficient Dual-System for Robotic Manipulation. *arXiv preprint arXiv:2410.08001*.

Bu, Q.; Zeng, J.; Chen, L.; Yang, Y.; Zhou, G.; Yan, J.; Luo, P.; Cui, H.; Ma, Y.; and Li, H. 2024b. Closed-Loop Visuomotor Control with Generative Expectation for Robotic Manipulation. In *Advances in Neural Information Processing Systems*.

Chi, C.; Xu, Z.; Feng, S.; Cousineau, E.; Du, Y.; Burchfiel, B.; Tedrake, R.; and Song, S. 2024. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *The International Journal of Robotics Research*.

Dai, Y.; Lee, J.; Fazeli, N.; and Chai, J. 2024. RACER: Rich Language-Guided Failure Recovery Policies for Imitation Learning. In *IEEE International Conference on Robotics and Automation*, 6892–6903.

Fischler, M. A.; and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6): 381–395.

Fu, Z.; Zhao, T. Z.; and Finn, C. 2024. Mobile ALOHA: Learning Bimanual Mobile Manipulation using Low-Cost Whole-Body Teleoperation. In *Conference on Robot Learning*.

Gu, J.; Kirmani, S.; Wohlhart, P.; Lu, Y.; Arenas, M. G.; Rao, K.; Yu, W.; Fu, C.; Gopalakrishnan, K.; Xu, Z.; et al. 2024. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *International Conference on Learning Representations*.

Guo, Y.; Wang, Y.-J.; Zha, L.; and Chen, J. 2023. Doremi: Grounding language model by detecting and recovering from plan-execution misalignment. *arXiv preprint arXiv:2307.00329*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations 2022, Virtual Event, April 25-29, 2022*.

Huang, H.; Lin, F.; Hu, Y.; Wang, S.; and Gao, Y. 2024a. Copa: General robotic manipulation through spatial constraints of parts with foundation models. *arXiv preprint arXiv:2403.08248*.

Huang, W.; Wang, C.; Li, Y.; Zhang, R.; and Fei-Fei, L. 2024b. ReKep: Spatio-Temporal Reasoning of Relational Keypoint Constraints for Robotic Manipulation. In *Conference on Robot Learning*.

Huang, W.; Wang, C.; Zhang, R.; Li, Y.; Wu, J.; and Fei-Fei, L. 2023a. VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models. In *Conference on Robot Learning*, 540–562.

Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; et al. 2023b. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *Conference on Robot Learning*, 1769–1782.

Karaev, N.; Rocco, I.; Graham, B.; Neverova, N.; Vedaldi, A.; and Rupprecht, C. 2024. CoTracker: It is Better to Track Together. In *European Conference on Computer Vision*.

Khazatsky, A.; Pertsch, K.; Nair, S.; Balakrishna, A.; Dasari, S.; Karamcheti, S.; Nasiriany, S.; Srirama, M. K.; Chen, L. Y.; Ellis, K.; et al. 2024. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*.

Kim, M. J.; Finn, C.; and Liang, P. 2025. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E. P.; Sanketi, P. R.; Vuong, Q.; Kollar, T.; Burchfiel, B.; Tedrake, R.; Sadigh, D.; Levine, S.; Liang, P.; and Finn, C. 2024. OpenVLA: An Open-Source Vision-Language-Action Model. In *Conference on Robot Learning*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 19730–19742.

Li, X.; Liu, M.; Zhang, H.; Yu, C.; Xu, J.; Wu, H.; Cheang, C.; Jing, Y.; Zhang, W.; Liu, H.; Li, H.; and Kong, T. 2024a. Vision-Language Foundation Models as Effective Robot Imitators. In *International Conference on Learning Representations*.

Li, X.; Zhang, M.; Geng, Y.; Geng, H.; Long, Y.; Shen, Y.; Zhang, R.; Liu, J.; and Dong, H. 2024b. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 18061–18070.

Liang, J.; Huang, W.; Xia, F.; Xu, P.; Hausman, K.; Ichter, B.; Florence, P.; and Zeng, A. 2023. code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation*, 9493–9500. IEEE.

Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems*, volume 36, 44776–44791.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024a. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36.

Liu, J.; Li, C.; Wang, G.; Lee, L.; Zhou, K.; Chen, S.; Xiong, C.; Ge, J.; Zhang, R.; and Zhang, S. 2024b. Self-Corrected Multimodal Large Language Model for End-to-End Robot Manipulation. *arXiv preprint arXiv:2405.17418*.

Liu, S.; Wu, L.; Li, B.; Tan, H.; Chen, H.; Wang, Z.; Xu, K.; Su, H.; and Zhu, J. 2025. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation. In *The Thirteenth International Conference on Learning Representations*.

Liu, Z.; Bahety, A.; and Song, S. 2023. REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction. In *Conference on Robot Learning*, 3468–3484.

Luo, F. 2024. Vision-Language Models for Robot Success Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 23750–23752.

Ma, X.; Patidar, S.; Haughton, I.; and James, S. 2024. Hierarchical Diffusion Policy for Kinematics-Aware Multi-Task Robotic Manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 18081–18090.

Minderer, M.; Gritsenko, A.; and Houlsby, N. 2024. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36.

Ni, F.; Hao, J.; Wu, S.; Kou, L.; Liu, J.; Zheng, Y.; Wang, B.; and Zhuang, Y. 2024a. Generate Subgoal Images before Act: Unlocking the Chain-of-Thought Reasoning in Diffusion Model for Robot Manipulation with Multimodal Prompts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 13991–14000.

Ni, F.; Jianye, H.; Wu, S.; Kou, L.; Yuan, Y.; Dong, Z.; Liu, J.; Li, M.; Zhuang, Y.; and ZHENG, Y. 2024b. PERIA: Perceive, Reason, Imagine, Act via Holistic Language and Vision Planning for Manipulation. In *Advances in Neural Information Processing Systems*.

Octo Model Team; Ghosh, D.; Walke, H.; Pertsch, K.; Black, K.; Mees, O.; Dasari, S.; Hejna, J.; Xu, C.; Luo, J.; Kreiman, T.; Tan, Y.; Chen, L. Y.; Sanketi, P.; Vuong, Q.; Xiao, T.; Sadigh, D.; Finn, C.; and Levine, S. 2024. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*. Delft, Netherlands.

O'Neill, A.; Rehman, A.; Maddukuri, A.; Gupta, A.; Padalkar, A.; Lee, A.; Pooley, A.; Gupta, A.; Mandlekar, A.; Jain, A.; Tung, A.; et al. 2024. Open X-Embodiment: Robotic Learning Datasets and RT-X Models : Open X-Embodiment Collaboration. In *IEEE International Conference on Robotics and Automation*, 6892–6903.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Ryu, H.; Kim, J.; An, H.; Chang, J.; Seo, J.; Kim, T.; Kim, Y.; Hwang, C.; Choi, J.; and Horowitz, R. 2024. Diffusion-EDFs: Bi-equivariant Denoising Generative Modeling on SE(3) for Visual Robotic Manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 18007–18018.

Sermanet, P.; Ding, T.; Zhao, J.; Xia, F.; Dwibedi, D.; Gopalakrishnan, K.; Chan, C.; Dulac-Arnold, G.; Maddineni, S.; Joshi, N. J.; et al. 2024. Robovqa: Multimodal long-horizon reasoning for robotics. In *IEEE International Conference on Robotics and Automation*, 645–652.

Song, C. H.; Wu, J.; Washington, C.; Sadler, B. M.; Chao, W.-L.; and Su, Y. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE International Conference on Computer Vision*, 2998–3009.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Walke, H. R.; Black, K.; Zhao, T. Z.; Vuong, Q.; Zheng, C.; Hansen-Estruch, P.; He, A. W.; Myers, V.; Kim, M. J.; Du, M.; et al. 2023. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, 1723–1736.

Wang, Y.; Li, X.; Wang, W.; Zhang, J.; Li, Y.; Chen, Y.; Wang, X.; and Zhang, Z. 2025. Unified Vision-Language-Action Model. *arXiv preprint arXiv:2506.19850*.

Wen, C.; Lin, X.; So, J.; Chen, K.; Dou, Q.; Gao, Y.; and Abbeel, P. 2024. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems*.

Wen, J.; Zhu, Y.; Li, J.; Zhu, M.; Wu, K.; Xu, Z.; Liu, N.; Cheng, R.; Shen, C.; Peng, Y.; et al. 2025. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. In *IEEE Robotics and Automation Letters (RA-L)*.

Zawalski, M.; Chen, W.; Pertsch, K.; Mees, O.; Finn, C.; and Levine, S. 2024. Robotic Control via Embodied Chain-of-Thought Reasoning. In *Conference on Robot Learning*.

Zhang, K.; Ren, P.; Lin, B.; Lin, J.; Ma, S.; Xu, H.; and Liang, X. 2024. PIVOT-R: Primitive-Driven Waypoint-Aware World Model for Robotic Manipulation. In *Advances in Neural Information Processing Systems*.

Zhang, Z.; Zheng, K.; Chen, Z.; Jang, J.; Li, Y.; Han, S.; Wang, C.; Ding, M.; Fox, D.; and Yao, H. ???? GRAPE: Generalizing Robot Policy via Preference Alignment. In *ICRA 2025 Workshop on Foundation Models and Neuro-Symbolic AI for Robotics*.

Zhao, Q.; Lu, Y.; Kim, M. J.; Fu, Z.; Zhang, Z.; Wu, Y.; Li, Z.; Ma, Q.; Han, S.; Finn, C.; et al. 2025. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Zheng, R.; Liang, Y.; Huang, S.; Gao, J.; Daumé III, H.; Kolobov, A.; Huang, F.; and Yang, J. 2024. TraceVLA: Visual Trace Prompting Enhances Spatial-Temporal Awareness for Generalist Robotic Policies. *arXiv preprint arXiv:2412.10345*.

Zhou, E.; Su, Q.; Chi, C.; Zhang, Z.; Wang, Z.; Huang, T.; Sheng, L.; and Wang, H. 2025. Code-as-Monitor: Constraint-aware Visual Programming for Reactive and Proactive Robotic Failure Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zitkovich, B.; Yu, T.; Xu, S.; Xu, P.; Xiao, T.; Xia, F.; Wu, J.; Wohlhart, P.; Welker, S.; Wahid, A.; et al. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Conference on Robot Learning*, 2165–2183.