LiLo-VLA: Compositional Long-Horizon Manipulation via Linked Object-Centric Policies

Yue Yang[1,†], Shuo Cheng[2], Yu Fang[1], Homanga Bharadhwaj[3],
Mingyu Ding[1], Gedas Bertasius[1], Daniel Szafir[1]

[1]**University of North Carolina at Chapel Hill** [2]**Georgia Institute of Technology** [3]**Carnegie Mellon University**

[†]Corresponding author: yygx@cs.unc.edu

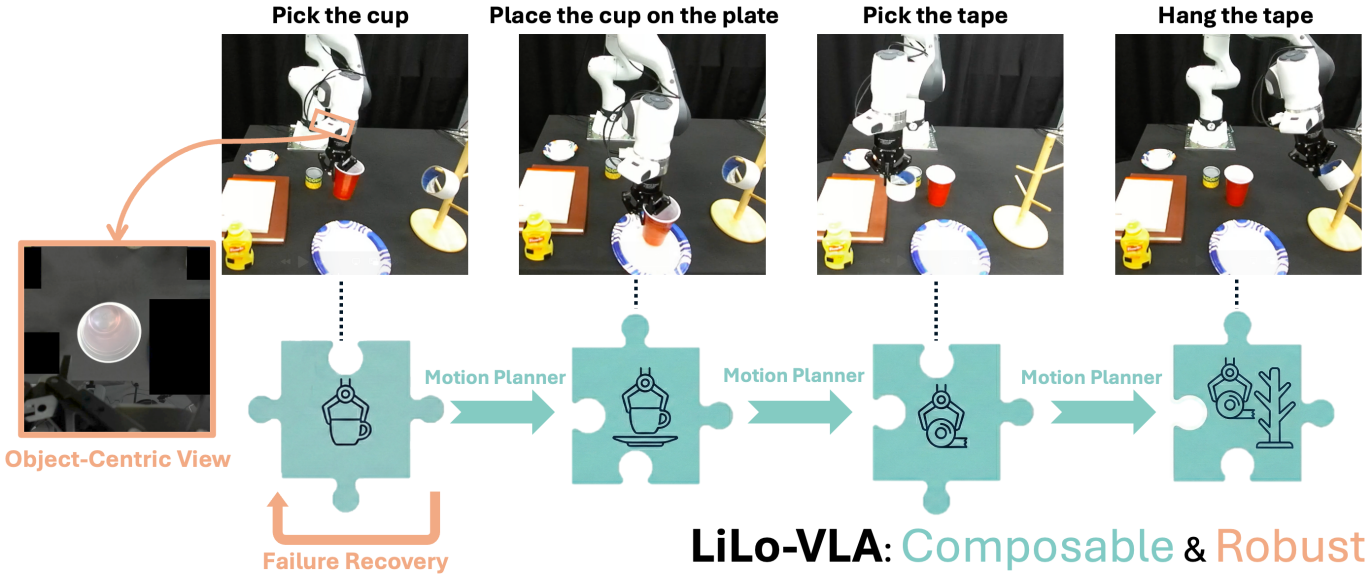

Fig. 1. **LiLo-VLA enables composable and robust manipulation.** LiLo-VLA solves long-horizon tasks by sequentially executing object-centric skill policies connected by robust motion planning. This enables zero-shot compositional generalization and robustness against cascading failures.

***Abstract*—General-purpose robots must master long-horizon manipulation, defined as tasks involving multiple kinematic structure changes (e.g., attaching or detaching objects) in unstructured environments. While Vision-Language-Action (VLA) models offer the potential to master diverse atomic skills, they struggle with the combinatorial complexity of sequencing them and are prone to cascading failures due to environmental sensitivity. To address these challenges, we propose LiLo-VLA (Linked Local VLA), a modular framework capable of zero-shot generalization to novel long-horizon tasks without ever being trained on them. Our approach decouples transport from interaction: a Reaching Module handles global motion, while an Interaction Module employs an object-centric VLA to process isolated objects of interest, ensuring robustness against irrelevant visual features and invariance to spatial configurations. Crucially, this modularity facilitates robust failure recovery through dynamic replanning and skill reuse, effectively mitigating the cascading errors common in end-to-end approaches. We introduce a 21-task simulation benchmark consisting of two challenging suites: LIBERO-Long++ and Ultra-Long. In these simulations, LiLo-VLA achieves a 69% average success rate, outperforming Pi0.5 by 41% and OpenVLA-OFT by 67%. Furthermore, real-world evaluations across 8 long-horizon tasks demonstrate an average success rate of 85%. Project page: https://yy-gx.github.io/LiLo-VLA/.**

I. Introduction

A grand goal of general-purpose robotics is to enable complex long-horizon manipulation, such as activities involving multiple skills (e.g., picking, placing, and pouring) to achieve high-level goals like cooking or cleaning. While recent generalist policies have achieved impressive results on short-horizon, single-stage manipulation [6, 7, 42, 36], extending these capabilities to multi-stage activities remains a grand challenge. Vision-Language-Action (VLA) models have emerged as a promising solution to bridge this gap, utilizing Internet-scale pre-training to enable diverse skill execution and semantic reasoning [21, 22, 3, 18]. However, applying current VLA paradigms to long-horizon tasks reveals two fundamental limitations. First, existing methods lack compositional generalization: they struggle to adapt to novel task sequences that were not explicitly seen during training, as they fail to flexibly recombine learned atomic skills without extensive task-specific demonstrations [32, 31]. Second, long-horizon execution is prone to cascading failures: since VLA policies easily overfit to visual features or specific spatial configurations, they become brittle to minor variations, where a single failure at any

stage jeopardizes the entire task sequence [41, 40].

To mitigate the data requirements for compositional generalization, recent research has explored learning modular skills. Approaches like PlanSeqLearn and local policy learning decouple tasks into atomic behaviors, employing motion planners for reaching and learned policies for interaction [8, 9]. However, these methods struggle with skill chaining, often relying on labor-intensive reward engineering or ad-hoc local goals for motion planning integration that hinder scalability. Alternatively, unified approaches like Long-VLA train a single end-to-end model to handle both transport and interaction phases via input masking [10]. Yet, these methods remain data-inefficient as they couple global transport with interaction, forcing the network to learn geometric motion planning through unstructured data. To address the second challenge of cascading failures, VLM-based recovery methods leverage the reasoning capabilities of foundation models to detect errors and trigger replanning [16, 28, 14]. However, these approaches primarily focus on task-level planning and rely on the critical assumption that underlying skills are robust. Simply reinvoking a skill policy that is sensitive to irrelevant visual features is ineffective, as it often leads to repeated failures. Moreover, execution failures frequently modify the spatial configuration of the scene, pushing the system outside the policy's initial state distribution.

To overcome these limitations, we introduce **LiLo-VLA** (**Li**nked **Lo**cal **VLA**), a framework that synergizes the strengths of classical motion planning and VLA policies to master complex long-horizon manipulation. We adopt a modular control strategy that decouples execution into two distinct phases. A Reaching Module employs classical motion planners to navigate a robot's end-effector to a target vicinity, where control is handed over to an Interaction Module, an object-centric VLA dedicated to fine-grained atomic manipulation. Through this modular design and the seamless integration of learned policies with classical planners, **LiLo-VLA** offers three decisive advantages. First, it achieves compositional generalization. By reusing object-centric atomic skills, the system can zero-shot generalize to novel long-horizon tasks without requiring any task-specific demonstration data. Second, it ensures robustness to environmental variations. By decomposing the scene and confining VLA observations strictly to the object of interest, the policy remains invariant to global workspace layouts and background clutter. Third, it enables inherent failure recovery. Rather than blindly retrying a failed skill, our system utilizes the Reaching Module to dynamically replan and reset the workspace, allowing for effective re-invocation of VLA skills. To comprehensively evaluate **LiLo-VLA**, we curate a challenging benchmark and provide the corresponding dataset, consisting of 21 tasks across two distinct suites. Crucially, both suites enforce randomized skill sequencing to rigorously stress-test compositional generalization. The first enhances LIBERO-Long [26] by introducing complex visual clutter to verify robustness against distractors. The second features custom ultra-long tasks chaining up to 16 atomic skills. This significantly exceeds the horizon length of prior benchmarks such as LIBERO-Long, which are typically limited to sequences of 3 to 4 skills, thereby challenging scalability. Our contributions in this work are four-fold:

- **LiLo-VLA Framework:** A modular framework for long-horizon tasks that enables zero-shot compositional generalization and achieves robustness to visual clutter and execution failures.
- **Evaluation Benchmarks:** A 21-task benchmark across two suites, "LIBERO-Long++" and "Ultra-Long", both of which evaluate compositional generalization while respectively focusing on visual robustness and extreme temporal scalability reaching up to 16 sequential skills.
- **Simulation Performance:** Extensive experiments show **LiLo-VLA** achieves a 69% average success rate, significantly outperforming Pi0.5 (28%) and OpenVLA-OFT (2%).
- **Real-World Validation:** Deployment on 8 long-horizon tasks (up to 8 skills) with complex backgrounds and varied sequence, achieving an 85% average success rate.

II. Related Work

A. Vision-Language-Action Models

Recent robot learning has been driven by VLA models that unify perception and control via Transformer architectures. Pioneering works like RT-2 [44] and Octo [37] demonstrated the efficacy of large-scale pretraining. Building on this, OpenVLA [21] utilized LLM backbones to enhance performance, while OpenVLA-OFT [22] improved control precision. Applications have expanded to bimanual control with RDT-1b [27] and humanoid embodiments via Project GR00t [2]. Recently, Pi0 [3] and Pi0.5 [18] leveraged flow matching and extensive real-world pretraining to capture complex action distributions. Despite improving atomic skill proficiency, these methods often overfit to visual signals and lack inherent compositional generalization.

B. Long-Horizon Manipulation and Skill Chaining

Solving long-horizon tasks requires effectively sequencing atomic behaviors to achieve a high-level goal. Classical Task and Motion Planning (TAMP) addresses this by combining symbolic search with geometric feasibility checks and engineered low-level controllers [19, 12, 34]. However, TAMP typically assumes accurate state estimation and known object models, which limits its robustness and applicability in unstructured, real-world environments. More recently hierarchical learning approaches have integrated learned models for generating low-level motions [5, 4, 30, 29]. To reduce reliance on manually defined planning domains, researchers have explored using large language models to decompose abstract task instructions into intermediate, executable subgoals. Frameworks, such as Code as Policy [23], ProgPrompt [35], SayCan [1] and VoxPoser [17], demonstrate strong semantic planning capabilities but often assume robust low-level execution. Most recently, LEAGUE [4] and Planseq-learn [8] pioneered the integration of motion planning

with RL trained policies for long-horizon tasks, while LongVLA [10] introduced the first VLA architecture explicitly designed for long-horizon manipulation. However, the former relies on ad-hoc integration strategies that hinder scalability, while the latter is data-inefficient by coupling global transport with local interaction.

III. Methodology

We describe **LiLo-VLA** in four stages: In Section III-A, we first formalize the problem and outline the system architecture. We then detail the Reaching Module, which handles global transport via robust motion planning, in Section III-B. Subsequently, Section III-C introduces our Interaction Module, featuring the Object-Centric VLA designed for local robustness. Finally, Section III-D describes the closed-loop execution pipeline and our hierarchical failure recovery mechanism.

A. Overview of *LiLo-VLA*

Formally, we define a long-horizon manipulation task as a sequence of symbolic actions $\mathcal{T} = \{a_1, a_2, \ldots, a_N\}$. Each action a_i is formulated as a parameterized predicate $\alpha_i(o_i, \rho_i)$, where α_i denotes the primitive skill category (e.g., `Pick`, `Place`), o_i represents the reference object that establishes the local coordinate frame for execution (e.g., the target object for a Pick action, or the receptacle for a Place action), and ρ_i encapsulates auxiliary parameters such as constraints. Although such high-level task skeleton can be generated automatically—either by symbolic planners [33, 12, 24] or by foundation models (LLMs/VLMs) [25, 11]—our focus in this work is on generating low-level robot motions that faithfully realize a given task plan $\mathcal{T}$ and achieve the desired goal under realistic geometric and dynamical constraints.

As illustrated in Figure 2, **LiLo-VLA** adopts a modular architecture to ground and execute these symbolic actions sequentially. For each action a_i, the system executes the corresponding atomic skill through a two-phase process. First, the Reaching Module (Figure 2 Top-Left) transports the end-effector from its current configuration to a robust approach pose defined relative to o_i via motion planning. Subsequently, the Interaction Module (Figure 2 Top-Right) is activated to perform the contact-rich manipulation using a learned policy centered on o_i with visual masking. As depicted in the bottom pipeline of Figure 2, this design philosophy effectively decouples the problem space. It utilizes motion planning for collision-free transport and leverages the Object-Centric VLA for local interaction, ensuring that the expensive learning capacity is focused exclusively on mastering the diverse dynamics of atomic skills while enabling autonomous failure recovery.

B. The Reaching Module: Global Transport

The Reaching Module serves as the bridge between distinct atomic skills, addressing the challenge of navigating the end-effector from the termination state of the previous skill to the initial state of the next. By leveraging a motion planner, this module ensures collision avoidance over long distances and guides the robot to a precise approach pose suitable for the downstream object-centric VLA. This approach pose also serves as a deterministic target for the motion planner during deployment.

1) Relative Goal Generation with Perturbation: The core function of this module is to determine the robust *Approach Pose*, $T_{approach} \in SE(3)$, which serves as the deterministic target for the motion planner. We derive this pose via a relative transformation from the reference object's frame: $T_{approach} = T_{o_i} \cdot T_{offset}(\alpha_i)$. Here, the offset transformation $T_{offset}(\alpha_i)$ is constructed to enforce a fixed face-down orientation relative to the object surface for all skills, while applying a skill-specific translation vector defined by α_i (e.g., a vertical clearance h_{pick} for picking tasks).

To ensure the downstream VLA policy is resilient to noise stemming from motion planning and perception, we introduce a perturbation strategy. During demonstration generation, rather than always initializing from the canonical approach pose $T_{approach}$, we randomize the start pose of each training trajectory by sampling

$$T_{init} = T_{approach}\, \Delta T, \qquad \Delta T = \exp(\hat{\xi}),$$

where $\xi \sim \mathcal{N}(0, \Sigma)$ defines a zero-mean perturbation in $\mathrm{SE}(3)$, injecting noise in both translation and rotation. The robot is transported to this perturbed state T_{init} to initiate each data collection episode. Consequently, during the deployment phase, while the Reaching Module targets the canonical $T_{approach}$, the VLA policy acquires strong local generalization capabilities by training on the dispersed distribution of T_{init}. This allows the Interaction Module to effectively compensate for deviations caused by pose estimation errors and imperfect motion planning convergence.

2) Collision-Free Motion Planning: Given the determined target pose, $T_{approach}$, during deployment or T_{init} during demonstration generation, we employ MPLib [13] to generate a collision-free trajectory. This planner integrates the environmental point cloud with the robot's kinematic chain to solve for a feasible path, ensuring safe global transport to the interaction start state without requiring any learning.

C. The Interaction Module: Object-Centric VLA

The reaching module brings the robot end-effector into proximity with the object. We model robot–object interaction by learning an end-to-end visuomotor policy π_θ that reactively predicts end-effector actions from the current observations. To ensure robustness against the spatial variability inherent in long-horizon tasks, we introduce a strictly object-centric design.

1) Object-Centric Observation Space: To ensure a strictly object-centric representation, our observation space relies exclusively on egocentric visual inputs captured by the wrist-mounted camera. We deliberately exclude static third-person views to mitigate the observation space shift (OSS) problem common in long-horizon manipulation [41], defined as the performance degradation caused by task-irrelevant visual changes across different stages of a long-horizon task. Since

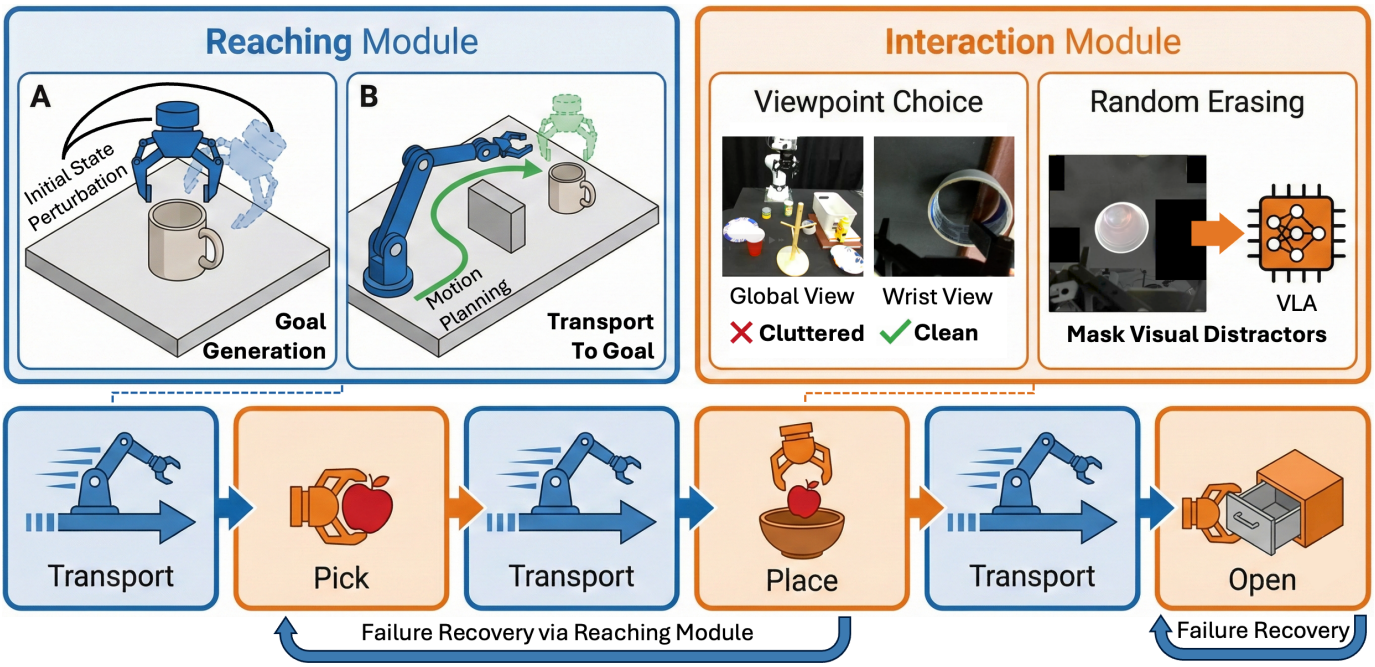

Fig. 2. **Architecture of LiLo-VLA.** Our framework decouples manipulation into two distinct phases. **(Top Left) The Reaching Module** handles global transport via collision-free motion planning. It employs initial state perturbation during training to ensure the policy to be robust to pose errors during deployment. **(Top Right) The Interaction Module** executes atomic skills via an object-centric VLA, strictly utilizing wrist-view observations and visual masking to eliminate environmental distractors. **(Bottom)** The system sequentially chains these modules, enabling closed-loop failure recovery where each skill's execution errors trigger a fallback to the Reaching Module for state resetting.

the robot base or global position may vary between skill executions, relying on fixed global cameras can introduce inconsistent visual features. In contrast, the wrist view maintains a consistent perspective relative to the workspace during interaction. This observation locality helps the robot focus on task-relevant objects, reducing the negative impact of visual distractors. We provide empirical validation for this design choice in Section IV-C2, demonstrating that a wrist-only policy outperforms multi-view configurations in atomic skill success rates.

2) Visual Clutter Augmentation: VLA policies frequently overfit to environmental distractors, such as objects relevant to other skills [40]. To eliminate this interference during deployment, we apply a heuristic masking strategy that covers non-target objects with black rectangles derived from their bounding boxes. However, these artificial occlusions constitute a significant visual domain shift, potentially leading to out-of-distribution (OOD) failures for a standard policy.

To ensure robustness against these artifacts, we introduce a mask-aware data augmentation strategy during training, inspired by [43]. We utilize segmentation to partition each training frame into foreground (the gripper, the reference object, and any grasped object) and background (all other pixels). We then apply a random erasing augmentation that overlays black rectangles exclusively onto the background regions. The number of rectangles and their total area ratio are sampled uniformly from a predefined range. This training procedure effectively simulates the visual artifacts introduced by our deployment masking, ensuring that the masked inference observations remain within the learned distribution of the policy.

Algorithm 1 Compositional Inference with Closed-Loop Recovery

Require: Skill sequence $\mathcal{T} = \{a_1, a_2, \ldots, a_N\}$
Require: Modules: Reaching $\mathcal{M}_{\text{reach}}$, Interaction $\mathcal{M}_{\text{int}}$
Initialize skill index $i \leftarrow 1$
while $i \leq N$ **do**
  $p_{\text{target}} \leftarrow \text{EstimateObjectPose}(a_i)$
  $\mathcal{M}_{\text{reach}}.\text{MoveToAbove}(p_{\text{target}})$
  $\mathcal{M}_{\text{int}}.\text{ExecuteSkill}(a_i)$
  $success \leftarrow \text{VerifyCondition}(a_i)$
  if success **then**
    $i \leftarrow i + 1$
  else
    if a_i involves holding object **then**
      $i \leftarrow \text{LastPickIndex}(\mathcal{T}, i)$
    else
      continue
    end if
  end if
end while

D. Compositional Execution and Failure Recovery

The complete inference pipeline integrates the proposed modules into a cohesive control loop, as summarized in Algorithm 1.

1) Sequential Execution Pipeline: For each atomic skill a_i in the plan, the system orchestrates the execution through a standardized pipeline (Alg. 1, Lines 3-6). First, the system estimates the 6D pose of the target object relevant to the current skill. This pose is ingested by the Reaching Module ($\mathcal{M}_{\text{reach}}$), which plans and executes a collision-free trajectory to navigate the end-effector to the approach pose, denoted as T_{approach}. This alignment ensures a consistent geometric initialization for the subsequent manipulation. Subsequently, the Interaction Module ($\mathcal{M}_{\text{int}}$) takes control to perform the fine-grained manipulation task using the VLA policy. Finally, the execution concludes with a geometric verification function $\mathcal{V}(a_i)$, which evaluates the spatial configuration of the object of interest to determine if the skill's effect has been satisfied.

2) Closed-Loop Recovery Mechanism: The modular architecture enables robust recovery from execution failures by adapting the control flow based on the semantic state of the robot (Alg. 1, Lines 8-12). When a failure is detected by $\mathcal{V}(a_i)$ for a skill that does not involve holding an object (e.g., a *Pick* attempt), the system triggers a local retry. By maintaining the current skill index i and re-initiating the loop, the system forces a re-evaluation of the spatial state. Specifically, this updates the object pose and resets the end-effector to T_{approach}, correcting any transient errors caused by the failed interaction and ensuring the policy is conditioned on the latest observation. In contrast, if a failure occurs during a skill that requires object retention (e.g., a *Place* operation), we conservatively assume that the object has been dropped or lost during transport. Operating under this assumption, a local retry is deemed risky. Therefore, the system backtracks the index i to the most recent *Pick* skill, ensuring the robot re-acquires the object before attempting the transport task again.

IV. Simulation Experiments

A. Experimental Setup

1) Benchmarks: To evaluate zero-shot compositional generalization and robustness to cascading failures, we curate a 21-task benchmark across two distinct suites within the LIBERO environment [26], derived from 9 core scenarios.

Suite 1: LIBERO-Long++. We select 6 tasks from LIBERO-Long that are amenable to sequence re-ordering and augment them with randomized visual distractors (e.g., mugs, cans). As shown in Fig. 3, this setup increases visual complexity to test robustness, forcing the policy to filter background clutter and attend strictly to task-relevant objects.

Suite 2: Ultra-Long. We design 3 tasks with increasing complexity: Kitchen Organization (9 steps), Cooking Preparation (10 steps), and Living Room Organization (16 steps). As illustrated in Fig. 3, the 16-step task presents a extreme challenge in workspace saturation; here, the high density of objects and receptacles pushes the kinematic reachability limits of a fixed-base tabletop robot. This suite systematically evaluates the system's ability to maintain coherent execution and temporal scalability over extended horizons.

Evaluation Protocol. To assess true compositionality rather than trajectory memorization, we evaluate multiple skill permutations for each scenario: 2 permutations for each of the 6 "Suite 1" scenarios and 3 for the 3 "Suite 2" scenarios (21 tasks in total). This protocol rigorously tests the system's ability to ground novel skill execution orders in zero-shot way.

2) Baselines: We compare **LiLo-VLA** against two state-of-the-art generalist policies: **Pi 0.5** [18], a flow-matching VLA, and **OpenVLA-OFT** [22], a highly optimized version of OpenVLA [21].

Setup for Suite 1. For standard sequences, baselines are LoRA fine-tuned [15] on the original continuous long-horizon demonstrations to learn skill transitions in a fixed order. For variant sequences with novel orders, we evaluate zero-shot performance by prompting the models with re-ordered language instructions.

Setup for Suite 2. Since collecting continuous demonstrations for every possible sequence permutation is combinatorially intractable, we train baselines on the aggregate dataset of constituent atomic skills to evaluate their inherent compositionality. During inference, we employ language chaining, sequentially prompting the model with sub-task descriptions to execute skills one-by-one. This evaluates whether VLAs can compose learned behaviors into novel long-horizon chains without requiring task-specific trajectory data.

3) Evaluation Metrics: We evaluate each task configuration over 10 trials and report performance using two standard metrics. **Success Rate (SR)** measures the percentage of episodes that reach the final goal state. Crucially, we enforce a strict standard: we count an episode as a success only if the robot executes the entire sequence of skills in the exact required order. **Average Progress (AP)** calculates the ratio of completed skills starting from the first step. We count only the continuous sequence of correct actions; if a skill is performed out of order, we stop counting immediately.

4) Implementation Details: To isolate manipulation capability from perception noise, we utilize simulator-provided ground-truth poses and segmentation masks. This oracle assumption ensures reported failures are attributable solely to the manipulation policy or motion planning rather than upstream estimation errors.

Data Generation. We segment LIBERO-90 demonstrations into atomic skills and augment them via MPLib to bridge the planner-policy domain gap. By generating perturbed initial states around the approach pose and planning trajectories back to the original demonstrations, we create a dataset of skills initialized from diverse starting configurations.

Policy Architecture. We adopt OpenVLA-OFT [22] as our backbone, retaining original hyperparameters for controlled comparison. To enforce object-centricity, we utilize the wrist camera view and apply random erasing to background pixels, mitigating visual overfitting to distractors.

Deployment. During inference, the system executes the symbolic plan sequentially. For each skill, the Reaching Module plans a collision-free path to an approach pose defined by a fixed offset; the Interaction Module then takes control using language instructions. We implement the failure detector $\mathcal{V}(S_i)$ via geometric heuristics to trigger closed-loop recovery.

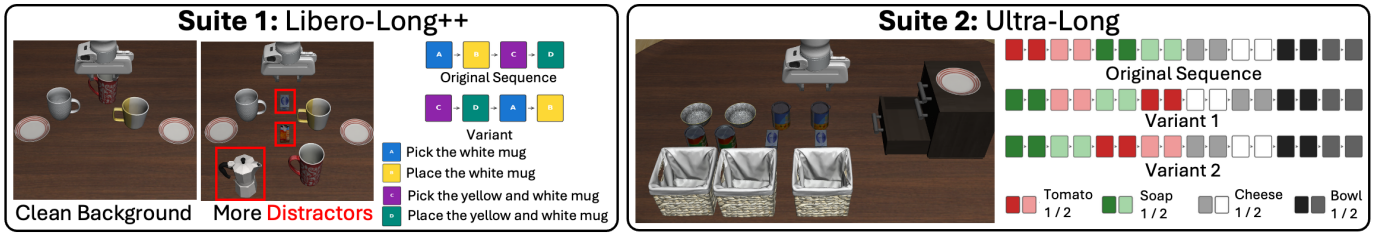

Fig. 3. **Overview of Evaluation Benchmarks.** We introduce two suites to evaluate long-horizon manipulation: Suite 1 (LIBERO-Long++) focuses on visual robustness by introducing more complex backgrounds with multiple distractors (highlighted in red), while Suite 2 (Ultra-Long) tests temporal scalability with task sequences extending up to 16 steps. Both suites incorporate multiple variant configurations with permuted skill orders to rigorously assess zero-shot compositional generalization.

B. Main Results: Zero-Shot Compositionality and Scalability

The quantitative results in Table I demonstrate that **LiLo-VLA** significantly outperforms state-of-the-art baselines across all metrics achieving an average success rate of 69% compared to 28% for Pi0.5 [18] and 2% for OpenVLA-OFT [22]. This substantial performance gap highlights the limitations of monolithic policies in handling the complexity of long-horizon manipulation. In the LIBERO-Long++ suite the results reveal a critical weakness in current VLAs where Pi0.5 achieves a high success rate on Original sequences (83%) which is statistically comparable to **LiLo-VLA** (78%, $p > 0.05$) yet collapses to 0% on Variant sequences. We observe that Pi0.5 frequently ignores the altered language instructions in these variant tasks and persists in executing the sequence order seen during training which confirms that the model overfits to the demonstrated trajectories rather than grounding the current language command. In contrast **LiLo-VLA** maintains robust performance (85%) by effectively isolating atomic skills. Furthermore in the ultra-long tasks of Suite 2 both baselines fail completely with a 0% success rate while **LiLo-VLA** maintains a 44% success rate. We attribute this failure to the coupling of global transport and local interaction in baseline models. Long-horizon tasks inherently involve evolving geometric configurations and spatial distribution shifts. Consequently baselines struggle to generalize to these changing layouts without intractable amounts of transition data. By explicitly decoupling global transport via a motion planner **LiLo-VLA** renders the interaction policy invariant to these global spatial shifts and effectively addresses the scalability bottleneck.

*C. Ablation Studies: Dissecting the **LiLo-VLA** Framework*

1) Necessity and Robustness of the Reaching Module: We first investigate the fundamental necessity of decoupling transport from interaction by removing the Reaching Module as shown in Table I. To address potential concerns regarding fair comparison and strictly isolate the impact of the modular architecture, we provided this ablation baseline with the same privileged ground-truth object poses used by our full method. Despite having access to this oracle geometric information, the policy fails completely with a 0% success rate across all tasks. This failure confirms that standard VLA architectures cannot implicitly learn long-horizon transport dynamics from atomic skill demonstrations alone even when the target location is known. Consequently, an explicit motion planner is not merely an auxiliary component but a structural prerequisite for bridging spatially distributed skills.

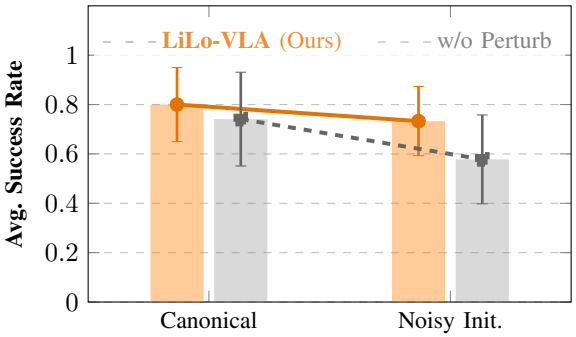

Fig. 4. **Impact of State Perturbation.** Average success rates across 27 unique skills demonstrate that our interaction policy remains robust to initial pose noise due to state perturbation, whereas the unperturbed policy degrades significantly.

Beyond necessity, we analyze the robustness of the interface between the motion planner and the policy. In real-world deployment, the system must contend with inevitable inaccuracies stemming from perception noise and imperfect motion planning convergence. To evaluate resilience to these factors, we conduct a comprehensive evaluation aggregating all 36 atomic skills from both Suite 1 and Suite 2, introducing execution noise to the initial poses of evaluation episodes as shown in Fig. 4. A baseline policy trained solely on canonical trajectory endpoints performs adequately under perfect initialization but degrades significantly under this noisy execution setting. In contrast, our full model which incorporates initial state perturbation during training maintains robust performance with no statistically significant drop in success rate ($p > 0.05$). This stability confirms that our data generation strategy effectively bridges the domain gap between the precision of the motion planner and the local generalization required by the policy.

2) Visual Robustness via Object-Centric Design: Next, we validate the design choices of our interaction module, focusing

TABLE I

EVALUATION ON LIBERO-LONG++ AND ULTRA-LONG. WE REPORT SUCCESS RATE (SR, %) AND AVERAGE PROGRESS (AP). **SUITE 1** TESTS ROBUSTNESS TO VISUAL CLUTTER. **SUITE 2** TESTS SCALABILITY (UP TO 16 SKILLS). GRAY COLUMNS INDICATE AVERAGE PERFORMANCE.

Method	Suite 1: Visual Clutter (LIBERO-Long++)						Suite 2: Scalability (Ultra-Long)								Overall	
	Original		*Variant*		*Avg.*		*Original*		*Variant 1*		*Variant 2*		*Avg.*		**Avg.**	
	SR	**AP**	**SR**	**AP**	**SR**	**AP**	**SR**	**AP**	**SR**	**AP**	**SR**	**AP**	**SR**	**AP**	**SR**	**AP**
Baselines																
Pi0.5	**83%**	**93%**	0%	0%	42%	46%	0%	1%	0%	0%	0%	0%	0%	0.3%	28%	31%
OpenVLA-OFT	7%	11%	0%	0%	3%	5%	0%	0%	0%	0%	0%	0%	0%	0%	2%	4%
Ablations																
w/o Reaching	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
w/o Masking	67%	80%	77%	87%	72%	83%	0%	16%	0%	32%	0%	10%	0%	20%	48%	64%
w/o Recovery	2%	25%	23%	59%	13%	42%	0%	16%	0%	16%	0%	21%	0%	18%	8%	33%
LiLo-VLA (ours)	78%	88%	**85%**	**89%**	**82%**	**89%**	**53%**	**79%**	**37%**	**84%**	**43%**	**74%**	**44%**	**79%**	**69%**	**86%**

on viewpoint selection and visual augmentation. To evaluate resilience to visual disturbances, we conduct experiments on the BOSS-C1 benchmark [41], which stress-tests policies against Observation Space Shift, defined as the phenomenon where changes in task-irrelevant visual predicates within the observation space disrupt the performance of a learned policy. As shown in Fig. 5, the "Wrist Only" configuration achieves the highest absolute success rate on original tasks (0.88). We attribute this to the wrist-mounted perspective, where the target object occupies a larger portion of the field of view, facilitating feature extraction compared to global views. Crucially, under OSS conditions, the "Wrist Only" configuration also exhibits the lowest Ratio Performance Delta (15.9%) compared to "Both Views" (18.8%) and "3rd Person" (29.8%). This confirms that the wrist view is inherently more robust to background clutter and task-irrelevant changes in the environment.

Complementing the viewpoint choice, the "w/o Masking" row in Table I highlights the necessity of our visual augmentation. Removing the random erasing strategy leads to a sharp decline in the overall success rate from 69% to 48%. This significant drop indicates that even with a wrist camera, the policy's receptive field inevitably includes environmental distractors. Explicitly masking these irrelevant regions during training is therefore essential to strictly enforce object-centricity and prevent the policy from overfitting to spurious visual features.

3) Efficacy of Closed-Loop Recovery: Finally, we examine the role of our closed-loop recovery mechanism. The "w/o Recovery" row in Table I reveals that removing this module leads to a catastrophic performance drop, with the overall average success rate plummeting from 69% to 8%. This failure is particularly absolute in the ultra-long tasks of Suite 2, where the success rate collapses to 0% without recovery compared to 44% with the full system. This dramatic difference demonstrates that closed-loop recovery is not merely an optimization but a structural prerequisite for scaling to extended horizons, where open-loop execution inevitably fails due to cumulative errors.

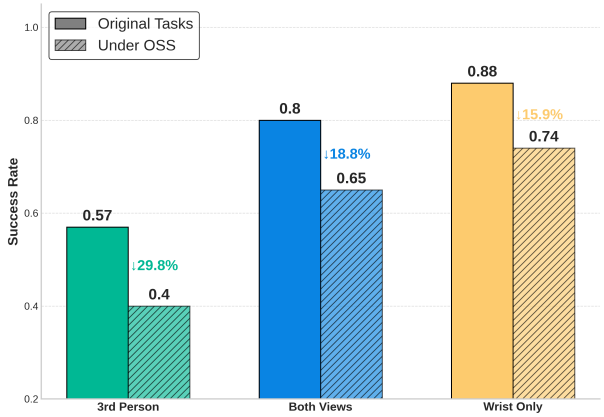

Fig. 5. **Comparison across different camera configurations.** Our wrist-only design achieves the highest success rate with the minimal performance drop under OSS.

V. REAL ROBOT EXPERIMENTS

A. Real-World System Implementation

We implement **LiLo-VLA** on a Franka Emika Panda robot equipped with a Robotiq 2F-85 parallel gripper. Following the standardized hardware protocol of DROID [20] we utilize a dual-camera setup where a static third-person ZED 2 stereo camera provides global context for the Reaching Module and a wrist-mounted ZED Mini camera captures egocentric data for the Interaction Module.

To transition from the oracle perception used in simulation to the unstructured real world we deploy a modular perception stack. We utilize YOLOE [38] for open-vocabulary object detection and segmentation coupled with FoundationPose [39] to estimate the 6D pose of target objects. This geometric information is ingested by the Reaching Module to generate collision-free motion plans to the target vicinity.

Complementing our simulation experiments which utilized OpenVLA-OFT [22], we deploy **LiLo-VLA** with the Pi0.5 [18] backbone in the real world. This architectural switch empirically validates the model-agnostic nature of our framework demonstrating that the modular benefits of **LiLo-VLA** can enhance diverse VLA architectures without modification.

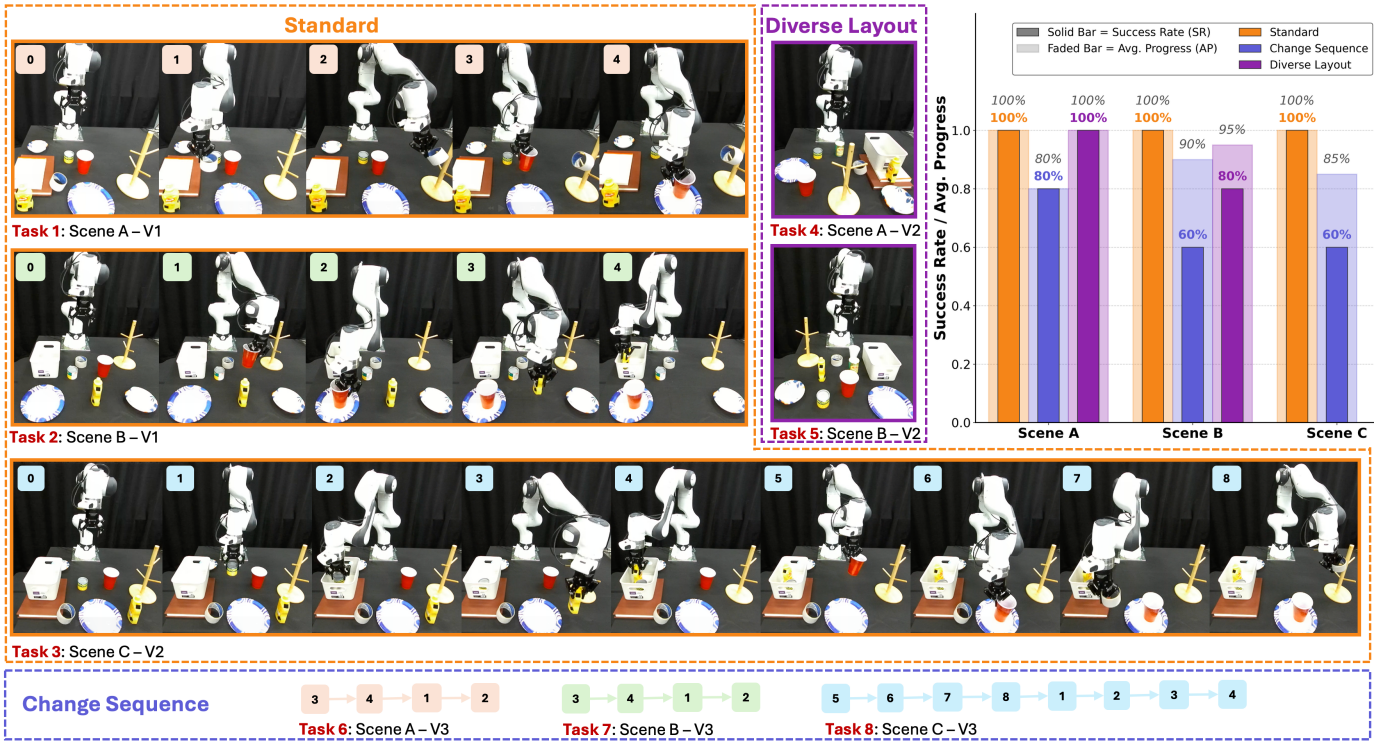

Fig. 6. **Real-World Experimental Evaluation.** We evaluate LiLo-VLA on 8 tasks ranging from 4 to 8 steps. To test generalization, we introduce: "Standard" configurations (Task 1-3), "Diverse Layouts" with more complex background and large layout variation (Task 4-5), and "Change Sequence" with permuted skill orders. The quantitative results (top right) demonstrate that LiLo-VLA maintains robust performance across all variations.

To train this policy we collect a minimal set of teleoperated demonstrations covering only atomic skills without any long-horizon sequence data. Finally due to the severe occlusions and visual clutter inherent in our evaluation tasks, we employ a human-in-the-loop protocol for success verification of each skill.

B. Evaluation Tasks: Stress-Testing Generalization

To evaluate the zero-shot generalization capabilities of **LiLo-VLA** in the physical world, we design three long-horizon tasks of increasing complexity as illustrated in Fig. 6. Task 1 and Task 2 each consist of 4 atomic skills, while Task 3 represents an ultra-long sequence chaining 8 atomic skills to push the limits of temporal scalability.

Beyond the Standard evaluation where tasks are executed in canonical configurations, we introduce two variant protocols to stress-test specific generalization axes. The first protocol, *Diverse Layout* (Variant 1), is applied to Tasks 1 and 2. Here we drastically alter the initial workspace layout and introduce previously unseen visual distractors. This setting evaluates the robustness of the Object-Centric VLA against visual clutter. The second protocol, *Change Sequence* (Variant 2), is applied across all three tasks. We permute the execution order of the atomic skills forcing the system to generate novel long-horizon behaviors that were never seen during data collection. This protocol strictly tests compositional generalization distinguishing true skill understanding from trajectory memorization.

C. Results Analysis

We report the quantitative results in Fig. 6 based on 5 evaluation trials per configuration. Under Standard conditions **LiLo-VLA** achieves a 100% success rate across all three tasks demonstrating effective sim-to-real transfer. When introduced to the "Diverse Layout" protocol the system exhibits strong resilience to spatial shifts and visual distractors maintaining a 100% success rate on Task 1 and 80% on Task 2. This stability validates the robustness of our object-centric Interaction Module against environmental noise. Furthermore under the "Change Sequence" protocol performance remains robust even when atomic skills are permuted into different orders. In the 8-step ultra-long Task 3 the system maintains a 60% success rate with 85% average progress. Collectively these results confirm that **LiLo-VLA** enables robust real-world manipulation capable of handling severe visual clutter and flexible skill composition.

VI. Conclusion and Limitations

In this work, we introduced **LiLo-VLA**, a modular framework that integrates a Reaching Module for global transport with an Interaction Module for local manipulation. By incorporating a closed-loop recovery mechanism our system effectively addresses observation space shifts and achieves robust performance in both simulation benchmarks and real-world deployments.

Despite these advancements, our reliance on external perception models introduces limitations where accurate detection

of transparent or severely occluded objects remains challenging. Additionally, the overall system performance is bounded by the atomic proficiency of the underlying VLA backbone. Future work will investigate active perception strategies that autonomously navigate to the most favorable viewpoints for execution, thereby mitigating visual occlusions and maximizing the performance of the VLA policy.

REFERENCES

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

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

[3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π0: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*.

[4] Shuo Cheng and Danfei Xu. League: Guided skill learning and abstraction for long-horizon manipulation. *IEEE Robotics and Automation Letters*, 8(10):6451–6458, 2023.

[5] Shuo Cheng, Caelan Reed Garrett, Ajay Mandlekar, and Danfei Xu. NOD-TAMP: Generalizable long-horizon planning with neural object descriptors. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=rThtgkXuvZ.

[6] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[7] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[8] Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, and Ruslan Salakhutdinov. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. *arXiv preprint arXiv:2405.01534*, 2024.

[9] Murtaza Dalal, Min Liu, Walter Talbott, Chen Chen, Deepak Pathak, Jian Zhang, and Ruslan Salakhutdinov. Local policies enable zero-shot long-horizon manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13875–13882. IEEE, 2025.

[10] Yiguo Fan, Pengxiang Ding, Shuanghao Bai, Xinyang Tong, Yuyang Zhu, Hongchao Lu, Fengqi Dai, Wei Zhao, Yang Liu, Siteng Huang, et al. Long-vla: Unleashing long-horizon capability of vision language action model for robot manipulation. *arXiv preprint arXiv:2508.19958*, 2025.

[11] Yunhai Feng, Jiaming Han, Zhuoran Yang, Xiangyu Yue, Sergey Levine, and Jianlan Luo. Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation. *arXiv preprint arXiv:2502.16707*, 2025.

[12] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4(1):265–293, 2021.

[13] RK Guo, Xinsong Lin, Minghua Liu, Jiayuan Gu, and Hao Su. Mplib: a lightweight motion planning library.

[14] Yanjiang Guo, Yen-Jen Wang, Lihan Zha, and Jianyu Chen. Doremi: Grounding language model by detecting and recovering from plan-execution misalignment. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12124–12131. IEEE, 2024.

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

[16] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[17] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[18] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. π0. 5: a vision-language-action model with open-world generalization, 2025. *URL https://arxiv.org/abs/2504.16054*, 1(2):3.

[19] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *2011 IEEE international conference on robotics and automation*, pages 1470–1477. IEEE, 2011.

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

[21] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla:

An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

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

[23] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

[24] Jacky Liang, Mohit Sharma, Alex LaGrassa, Shivam Vats, Saumya Saxena, and Oliver Kroemer. Search-based task planning with learned skill effect models for lifelong robotic manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6351–6357. IEEE, 2022.

[25] Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47(8):1345–1365, 2023.

[26] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[27] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[28] Zeyi Liu, Arpit Bahety, and Shuran Song. Reflect: Summarizing robot experiences for failure explanation and correction. *arXiv preprint arXiv:2306.15724*, 2023.

[29] Ajay Mandlekar, Caelan Reed Garrett, Danfei Xu, and Dieter Fox. Human-in-the-loop task and motion planning for imitation learning. In *Conference on Robot Learning*, pages 3030–3060. PMLR, 2023.

[30] Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, pages 2905–2925. PMLR, 2023.

[31] Nived Rajaraman, Lin Yang, Jiantao Jiao, and Kannan Ramchandran. Toward the fundamental limits of imitation learning. *Advances in Neural Information Processing Systems*, 33:2914–2924, 2020.

[32] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[33] Tom Silver, Rohan Chitnis, Joshua Tenenbaum, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Learning symbolic operators for task and motion planning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3182–3189. IEEE, 2021.

[34] Tom Silver, Ashay Athalye, Joshua B Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Learning neuro-symbolic skills for bilevel planning. *arXiv preprint arXiv:2206.10680*, 2022.

[35] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

[36] Generalist AI Team. Gen-0: Embodied foundation models that scale with physical interaction. *Generalist AI Blog*, 2025. https://generalistai.com/blog/nov-04-2025-GEN-0.

[37] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[38] Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anything, 2025. URL https://arxiv.org/abs/2503.07465.

[39] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[40] Kechun Xu, Zhenjie Zhu, Anzhe Chen, Shuqi Zhao, Qing Huang, Yifei Yang, Haojian Lu, Rong Xiong, Masayoshi Tomizuka, and Yue Wang. Seeing to act, prompting to specify: A bayesian factorization of vision language action policy. *arXiv preprint arXiv:2512.11218*, 2025.

[41] Yue Yang, Linfeng Zhao, Mingyu Ding, Gedas Bertasius, and Daniel Szafir. Boss: Benchmark for observation space shift in long-horizon task. *IEEE Robotics and Automation Letters*, 2025.

[42] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[43] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[44] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.