EvoWorld: A World-Model-Centric Framework for Continuous Self-Evolution of Modular Embodied Skills

Boshi Zhang[*1], Sen Cui[*,‡1], BaoHuaYin[3], Youyi Kou[4], Junyu Wu[5], Zuo Pu[1], Tao Xue[1], Zhikang Chen[2], Shanshan Wei[1], Min Zhang[6], Miao Liu[1], Changshui Zhang[†1], and Zhang Tao[†1]

[1]Tsinghua University, Beijing, China [2]University of Oxford, Oxford, UK
[3]University of Sussex, Brighton, UK [4]Beijing Jiaotong University, Beijing, China
[5]Xiamen University, Xiamen, China [6]East China Normal University, Shanghai, China
zbs22@mails.tsinghua.edu.cn; cuis@mail.tsinghua.edu.cn
zcs@mail.tsinghua.edu.cn; taozhang@tsinghua.edu.cn
[*]Equal contribution. [‡]Project leader. [†]Corresponding authors.

Abstract. Current progress in physical intelligence largely relies on scaling monolithic Vision-Language-Action (VLA) models, yet real-world policy data remain fragmented across scenes and tasks. This mismatch limits transfer, exacerbates catastrophic forgetting, and impedes continual improvement. A modular design that shares dynamics while specializing skills is therefore a promising paradigm.

We introduce EvoWorld (EvoW), a world-model-centric framework for skill orchestration and iterative self-evolution. In EvoW, VLAs form an expandable library of pluggable experts. A high-level router selects experts conditioned on scene and task, while an action-conditioned video world model serves as the cognitive core for rollout-based planning. The world model provides counterfactual rollouts to score candidate experts, while selected experts execute in the grounded scene to generate trajectories for verification. A vision-language evaluator delivers semantic scoring and diagnostic tags, enabling targeted updates to the world model, router memory, or specific experts rather than global retraining. This closes an automated loop that jointly improves grounding, routing, and skill refinement without manual task engineering. Experiments show that EvoW enables automated task-to-policy synthesis with competitive success rates in our evaluated settings and iterative improvement trends over iterations, while producing valid and diverse trajectories that support evaluation and skill refinement.

Keywords: Vision-Language-Action Models · World Models · Expert Routing · Robot Manipulation · Continual Skill Refinement

1 Introduction

Physical intelligence is increasingly required to follow open-ended user intent rather than a fixed list of benchmark tasks [1, 5, 11]. To meet this demand,

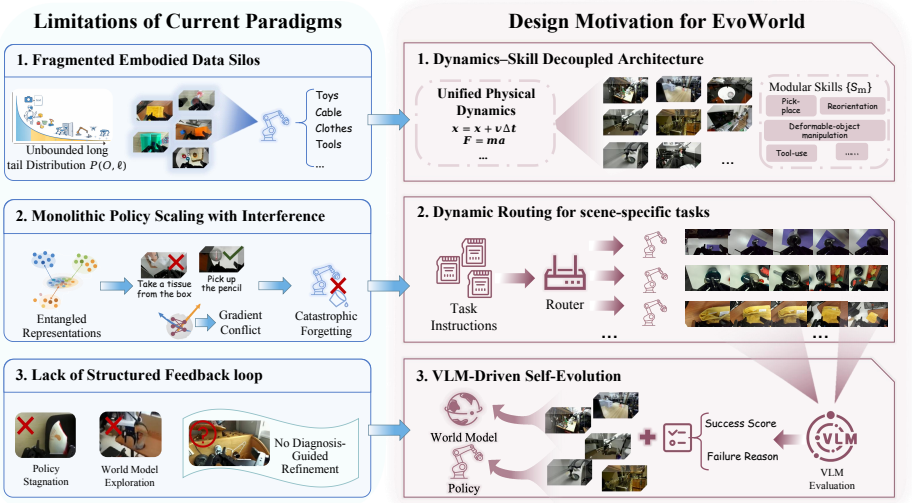


Fig. 1: Motivation and system overview of EvoW. Left: current embodied-learning pipelines face three structural bottlenecks: (1) fragmented embodied data silos with long-tail task/scene distributions, (2) interference and gradient conflict in monolithic policy scaling that induce catastrophic forgetting, and (3) the lack of a diagnosis-guided feedback loop for iterative refinement. **Right:** the proposed framework resolves these bottlenecks via three coupled components: (1) a dynamics–skill decoupled architecture that separates reusable physical dynamics from modular task-specialized skills, (2) scene-aware dynamic routing that maps task instructions and grounded observations to the most suitable expert policy, and (3) a VLM-driven self-evolution loop that scores outcomes, identifies failure causes, and feeds diagnostic signals back to world-model adaptation and skill updates.

the field has largely trended toward scaling monolithic Vision-Language-Action (VLA) models and diffusion-style policies to capture a broad spectrum of robotic behaviors [6,8,26]. Large-scale datasets such as DROID and Open X-Embodiment have further fueled this progress by providing unprecedented task and visual coverage [9, 25]. However, despite this progress, a potential misalignment between universal physical dynamics and highly task-specific execution strategies might have been partially overlooked in the pursuit of monolithic scaling, leaving the challenge of pervasive real-world data silos somewhat under-addressed.

Current generalist control approaches rely predominantly on training large VLA models or flow-based policies on heterogeneous, massive data sources to achieve broad manipulation coverage [4–6, 8, 21, 26, 29, 46]. By treating diverse robotic experiences as tokens in a shared transformer-style backbone, these methods aim to absorb both physical dynamics and task-execution patterns through scale [4–6,21,26]. This paradigm assumes that a single, sufficiently large network can generalize across embodiments and environments via a shared latent space. Monolithic methods can face practical limitations under fragmented scene distributions and long-horizon compounding errors [2,27]. In practice, robotic data remain distributed across labs, embodiments, and collection protocols [9,12,25], making continual updates to a single policy costly and vulnerable to forget-

ting when new task families are introduced. Moreover, without explicit feasibility verification during rollout, these systems can drift semantically and become physically inconsistent in open-world settings [2, 27, 31, 45].

We identify the root of these challenges in the lack of fundamental decoupling between universal physical dynamics and specialized, task-specific execution strategies [16–19, 38]. While the underlying laws of physics are largely invariant across scenarios, the "skills" required for manipulation are diverse and highly context-dependent [4, 21, 26, 29]. Current paradigms lack an explicit mapping mechanism to route task semantics to the most suitable algorithmic specialist, forcing a single network to resolve competing gradients from disparate domains [41, 42]. We argue that a more rational architecture for physical intelligence should treat the world model as a universal cognitive foundation while maintaining policies as modular, pluggable skills that can be updated independently [19, 38].

EvoWorld (EvoW) is proposed as a world-model-centric framework for task-to-policy automation that enables continuous skill evolution [19, 38]. Unlike monolithic approaches [4, 21, 26, 29], **EvoW** separates shared physical grounding from task-level execution. It treats VLAs as modular components in an expandable library. The framework operates through a principled pipeline: (i) anchoring the task in a high-fidelity physical scene via semantic retrieval, (ii) performing counterfactual simulation using an action-conditioned video world model [31, 45], (iii) routing the task to the optimal VLA expert through a scene-aware scheduler, and (iv) driving a closed-loop self-evolution cycle via Vision-Language Model (VLM) diagnosis. This design allows the system to autonomously refine the world model and retrain specific skills without the need to re-initialize the entire intelligence substrate.

Experiments demonstrate that EvoW significantly outperforms monolithic baselines [4, 21, 26, 29] in success rates, particularly when operating across diverse and fragmented data silos. By leveraging the world-model-driven closed loop, the system achieves continuous compounding improvements across evolutionary iterations, validating the effectiveness of VLM-based failure diagnosis and expert routing. The results confirm that decoupling dynamics from skills not only mitigates catastrophic forgetting but also enables a more efficient, automated path toward open-ended physical intelligence. We summarize our key contributions as follows:

- Paradigm: We propose a principled world-model-centric formulation that disentangles universal physical dynamics from task-specialized skills, shifting the focus from policy scaling to world-model-driven orchestration.
- Method: We implement the EvoW framework, which operationalizes this decoupling through retrieval-grounded scene construction and a VLM-driven diagnostic loop that enables autonomous, closed-loop self-evolution.
- Validation: Extensive experiments across data silos verify the necessity of expert routing and self-evolution, demonstrating that EvoW achieves compounding performance gains without the overhead of full-system retraining.

2 Related Works

Scaling policies and data. Generalist VLA policies such as RT-1, RT-2, and OpenVLA demonstrate strong instruction-following behavior through scale [5,6, 26]. Industrial foundation efforts (e.g., GR00T N1) further push this direction with larger model and data pipelines [29]. In parallel, large real-robot datasets, including Open X-Embodiment, DROID, and RH20T, broaden scene and task coverage [9, 12, 25]. However, most of these pipelines remain monolithic, where updates are largely global and specialist selection at deployment time is weak. Under fragmented data silos, this often leads to interference and higher continual-update cost.

Manipulation policies and structured primitives. Beyond model/data scaling, manipulation research focuses on control structure, including end-to-end visuomotor learning, reinforcement learning, and imitation learning [24, 28, 36]. Structured formulations such as Transporter Networks and CLIPort improve compositionality and sample efficiency in multi-step tasks by introducing stronger inductive bias for spatial reasoning and action decomposition [35, 43]. Recent dexterity-oriented works further extend policy capacity under richer embodiment and contact settings [44,46]. However, these methods are typically optimized within fixed policy-training pipelines and still lack a diagnosis-guided, module-specific refinement loop that can coordinate grounding, routing, and world-model updates.

Grounding and memory for embodied reuse. Grounding and memory-based approaches improve generalization by reusing prior trajectories, grounded plans, and contextual cues [34]. Their core strength is efficient reuse under recurring task patterns. Yet, in many settings, grounding is not tightly coupled with verifiable rollout generation and failure attribution. As a result, these approaches are less suited to autonomous task-to-policy iteration when scenes and task semantics shift together in open-ended real-robot settings.

World models for long-horizon control. World models support imagination-based rollouts for planning and policy improvement [16–18]. Recent latent-action and video-prediction models improve rollout horizon and perceptual realism [13, 14,40]. Other studies investigate tighter coupling between world models and control policies [7, 22, 23]. A core challenge remains long-horizon robustness: compounding prediction errors and semantic drift still accumulate over time [2, 27].

Automated task generation in simulation and our position. RoboGen shows that closed-loop task generation can scale quickly in simulation [39]. For real-robot deployment, additional constraints are necessary: real-scene grounding, expert routing at deployment, and diagnosis-guided correction. EvoW addresses this gap by combining grounded initialization, task/scene-aware routing, and targeted diagnosis-guided updates in a world-model-centric loop, with explicit decoupling between shared dynamics and task-specialized skills.

Taken together, existing lines of work provide strong components but not an integrated mechanism that jointly addresses grounded scene initialization, deployment-time expert selection, and diagnosis-driven evolution under open-ended real-robot distributions. This motivates our design of EvoW as a closed-

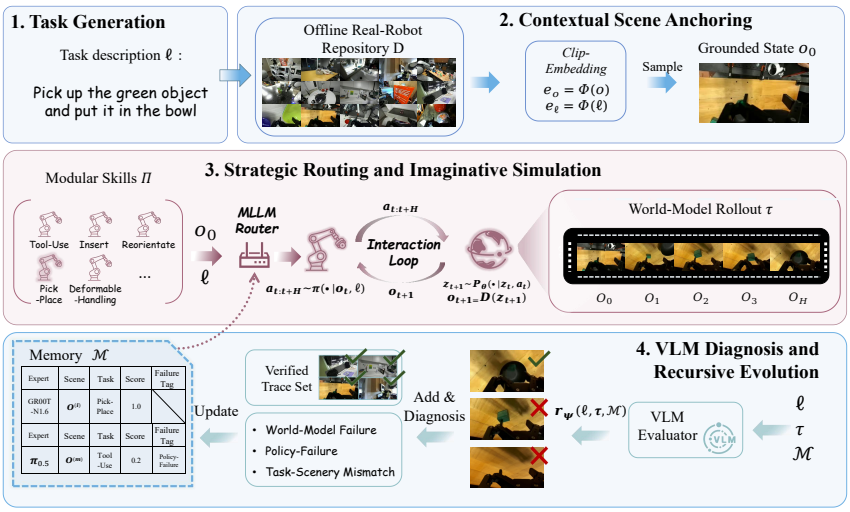


Fig. 2: EvoWorld pipeline overview. (1) Task Generation: construct candidate task specifications for each round. **(2) Contextual Scene Anchoring:** select a semantically aligned real scene from offline experience. **(3) Strategic Routing and Imaginative Simulation:** score candidates via world-model imagination, route to a suitable VLA expert, and execute the selected expert. **(4) VLM Diagnosis and Recursive Evolution:** use success score and failure type to trigger targeted module updates rather than global retraining.

loop system that links these modules through a unified world-model-centric objective, so that policy improvement is both targeted and scalable rather than globally retrained at every iteration.

3 Method

In this section, we delineate the EvoW framework across three strategic dimensions: a formal problem formulation, the modular architectural design, and a diagnosis-driven closed-loop optimization. Our approach emphasizes how Contextual Scene Anchoring, Strategic Routing, and Action-conditioned World-model Rollouts are synergistically coupled to facilitate continuous, autonomous task-to-policy evolution.

3.1 Notations and Problem Setup

We formalize open-ended task-to-policy synthesis as a language-conditioned decision process. Let ℓ denote a natural language task instruction (e.g., Pick up the green object and put it in the bowl). At each timestep t, o_t represents the visual observation and a_t the executed action. An interaction episode of horizon H is defined as a trajectory $\tau = (o_0, a_0, \ldots, o_H)$, consisting of the initial grounded state and subsequent state-action pairs. To address the limitations of monolithic scaling in fragmented environments, EvoW decomposes this process into universal dynamics and specialized control:

A Universal Video World Model (p_θ): Acting as the system's cognitive engine, this model captures transferable environmental dynamics across diverse scenarios. Given the action-conditioned latent transition, it is formulated as:

$$z_{t+1} \sim p_\theta(z_{t+1} \mid z_t, a_t), \quad o_{t+1} = D(z_{t+1}), \tag{1}$$

where E encodes observations into a latent space $z_t = E(o_t)$, and D decodes predicted future states back into the observation manifold.
A Modular Skill Library (Π): This library aggregates task-specific execution strategies optimized for distinct task families. It is defined as a collection of M specialized VLA experts:

$$\Pi = \{\pi_m\}_{m=1}^{M}, \quad a_{t:t+H} \sim \pi_m(\cdot \mid o_t, \ell), \tag{2}$$

where a high-level Strategic Router q maps the grounded scene and instruction to the optimal expert π_{m^*} based on the experience memory $\mathcal{M}$:

$$m^* = \text{argmax}_m q(m \mid o_0, \ell, \mathcal{M}). \tag{3}$$

This world-model-centric decomposition allows physical dynamics to be shared as a universal prior, while policy capacity is allocated through modular specialization and MLLM-driven routing. As illustrated in Figure 2, this architectural decoupling enables EvoW to bypass the constraints of real-world data silos and achieve continuous self-improvement through an introspective diagnostic loop.

3.2 Overview

The goal of EvoW is to establish a world-model-centric framework that enables autonomous skill acquisition and recursive self-improvement across fragmented data silos. Unlike monolithic VLA paradigms that entangle physical dynamics with task policies, EvoW advocates for a decoupled architecture where a universal Video World Model serves as the cognitive foundation, and Modular Skills act as pluggable experts. As illustrated in Figure 2, the EvoW pipeline operates through four synergized phases:
Phase 1: Task Generation. The cycle begins with the formulation of open-vocabulary task specifications. Rather than relying on a static benchmark, EvoW adaptively constructs candidate instructions to explore the capabilities of the current skill library, ensuring that the evolution process remains open-ended and diverse.
Phase 2: Contextual Scene Anchoring. To bridge the gap between abstract instructions and physical reality, we perform Contextual Scene Anchoring (Grounding). Given a task, we leverage an offline experience manifold $\mathcal{D}$ (e.g., the DROID dataset) to extract a semantically aligned initial state o_0. We employ a contrastive embedding space to compute the similarity between the instruction and visual candidates, sampling an initial grounded state.
Phase 3: Strategic Routing and Imaginative Simulation. Upon anchoring the scene, a high-level Skill Router queries the internal Memory to schedule

the most suitable VLA expert π_m from the skill library. The selected expert interacts with the action-conditioned world model. The world model performs rollouts $\tau = \{o_0, o_1, \ldots, o_H\}$ by predicting latent dynamics $z_{t+1} \sim p_\theta(\cdot|z_t, a_t)$ and decoding them into observation space. This allows the system to visualize the physical consequences of the expert's actions before real-world execution.

Phase 4: VLM Diagnosis and Recursive Evolution. The final phase closes the loop through a diagnosis. A Vision-Language Model (VLM) functions as a reflective observer to evaluate the simulated rollout. Crucially, the VLM performs fine-grained credit assignment, categorizing failures into World-Model Failure, Policy Failure, or Task-Scenery Mismatch. These diagnostic tags trigger targeted updates in the following loop.

3.3 Contextual Scene Anchoring

To bridge the gap between abstract linguistic intent and executable physical behavior, EvoW adopts a Contextual Scene Anchoring paradigm. The core objective is to initialize a semantically aligned real-world observation, thereby constraining the subsequent simulation to a physically plausible manifold rather than generating outcomes in a vacuum.

We define an offline experience manifold $\mathcal{D} = \{(o_i, \xi_i)\}_{i=1}^N$, where each entry represents a high-fidelity snapshot of real-robot interaction consisting of an observation o_i and its associated auxiliary context ξ_i. Given an open-vocabulary task specification ℓ, we first project the language instruction and the visual candidates into a shared latent embedding space:

$$e_\ell = \Phi_L(\ell), \qquad e_{o_i} = \Phi_O(o_i), \tag{4}$$

where Φ_L and Φ_O denote pre-trained dual encoders optimized for cross-modal alignment. To ensure diverse yet relevant initialization, we ground ℓ by sampling an initial state o_0 from the distribution $p(i \mid \ell)$, formulated as a semantic-similarity-based Softmax:

$$o_0 \sim p(i \mid \ell) = \frac{\exp(\langle e_\ell, e_{o_i} \rangle / \tau_{\text{gnd}})}{\sum_{j=1}^N \exp(\langle e_\ell, e_{o_j} \rangle / \tau_{\text{gnd}})}, \tag{5}$$

where τ_{gnd} is a temperature parameter controlling the exploration-exploitation trade-off during grounding.

Once the initial observation o_0 is anchored, an Action-Conditioned Video World Model serves as the primary cognitive engine for Counterfactual Simulation. The process begins by encoding o_0 into a compressed latent space $z_0 = E(o_0)$. The system then generates a temporal rollout $\tau = \{o_0, o_1, \ldots, o_H\}$ through recursive latent transitions: $z_{t+1} \sim p_\theta(z_{t+1} \mid z_t, a_t)$, and $o_{t+1} = D(z_{t+1})$, where p_θ models the stochastic transition dynamics conditioned on the predicted action a_t from the selected VLA expert, and D decodes the latent states back into pixel space.

This hierarchical factorization explicitly disentangles Scene Grounding from Dynamics Prediction. By anchoring the simulation in a real-world experience

manifold $\mathcal{D}$, EvoW effectively mitigates the accumulation of compounding errors and semantic drift that typically plague long-horizon generative models. This ensures that the world model provides a stable and physically consistent substrate for policy evaluation and autonomous skill evolution.

3.4 Strategic Routing and Imaginative Simulation

Given the fragmented nature of real-world robotic data, a single monolithic policy often suffers from catastrophic forgetting and performance degradation across diverse embodiments. To circumvent this, EvoW treats each specialized VLA as a modular expert π_m within a library $\{\pi_m\}_{m=1}^{M}$, where each expert is optimized for specific task families or physical constraints. The core challenge then shifts to Strategic Routing: mapping an open-vocabulary instruction ℓ and a grounded observation o_0 to the most compatible expert.

To facilitate this mapping, we implement a strategic router realized through a Multi-modal Large Language Model (MLLM). The MLLM acts as a high-level reasoning engine that leverages its extensive semantic knowledge to evaluate the compatibility between the task requirements and the functional descriptions of the VLA experts. To enable continuous self-improvement, the router is conditioned on a long-term Experience Memory $\mathcal{M}$, defined as:

$$\begin{aligned} \pi_{m^*} &= q(\ell, o_0, \mathcal{M}), \\ \mathcal{M} &= \{(\ell^i, o_0^i, \pi_m^i, y^i, \kappa^i)\}_{i=1}^{N}. \end{aligned} \tag{6}$$

where each entry captures a historical interaction consisting of the task ℓ, the initial state o_0, the selected expert π_m, the resulting performance score y, and the diagnostic failure type κ.

The routing process operates on a dual-mode logic of Exploration and Exploitation:

Exploration: For unfamiliar task-scene pairs with low semantic density in $\mathcal{M}$, the router encourages diversity by sampling from a broader categorical distribution, allowing the system to discover new "scene-expert" correspondences.

Exploitation: For recurrent contexts, the MLLM synthesizes historical success/failure cues from $\mathcal{M}$ to generate memory-weighted logits, effectively biasing the selection toward experts with proven reliability in similar manifolds.

To further solidify the routing decision, for each candidate expert π_m identified by the MLLM, the world model generates an imagined rollout τ_{π_m}. The router then computes an imaginative rollout score, which reflects the predicted task-consistency of the expert's trajectory. The final expert π_{m^*} is selected by maximizing the joint probability of semantic compatibility and simulated success, thereby ensuring that the execution is grounded in both high-level intent and low-level physical feasibility.

3.5 VLM Diagnosis and Recursive Evolution

Algorithm 1 EvoW closed-loop routing and self-evolution

Require: Task set $\mathcal{L}$, offline experience manifold $\mathcal{D}$, expert library $\Pi = \{\pi_m\}_{m=1}^{M}$, video world model p_θ, MLLM-based router q, VLM evaluator Ψ, experience memory $\mathcal{M}$, success threshold δ.
Ensure: Verified trajectory set $\widehat{\mathcal{T}}$, updated skill library Π, and refined memory $\mathcal{M}$.

$\widehat{\mathcal{T}} \leftarrow \emptyset$
for each round **do**
  (Stage 1: Task Generation) Sample or construct task specification ℓ
  (Stage 2: Contextual Scene Anchoring) Ground initial scene o_0 from $\mathcal{D}$ conditioned on ℓ
  (Stage 3: Strategic Routing & Imaginative Simulation) Select expert $\pi_{m^*} \leftarrow q(\cdot \mid \ell, o_0, \mathcal{M})$; run world-model rollout for candidate scoring; execute selected expert to obtain trajectory τ
  (Stage 4: VLM Diagnosis & Recursive Evolution) Evaluate $(r_\psi, \kappa) \leftarrow \Psi(\tau, \ell, \mathcal{M})$ and update $\mathcal{M} \leftarrow \mathcal{M} \cup \{(\ell, o_0, m^*, r_\psi, \kappa)\}$
  if $r_\psi \geq \delta$ **then**
    $\widehat{\mathcal{T}} \leftarrow \widehat{\mathcal{T}} \cup \{(\tau, \ell, o_0, m^*, r_\psi)\}$
  else
    if $\kappa = \mathrm{TSM}$ **then**
      Update grounding module
    else if $\kappa = \mathrm{PF}$ **then**
      Update routing and/or selected expert
    else
      Update world model
    end if
  end if
end for
return $\widehat{\mathcal{T}}, \mathcal{M}$

To achieve continuous self-improvement while circumventing the computational overhead and catastrophic forgetting associated with global retraining, EvoW implements an introspective diagnosis mechanism leveraging the cross-modal reasoning capabilities of Vision-Language Models (VLMs). This stage functions as a fine-grained credit assignment process, precisely pinpointing the functional bottleneck within the decoupled architecture. Upon the execution of a routed VLA expert π_m that generates a trajectory τ, the VLM evaluator Ψ performs a reflective assessment by contrasting the actual outcome with the task specification ℓ and the initial physical grounding. Formally, the evaluation is defined as:

$$(r_\psi, \kappa) = \Psi(\tau, \ell, \mathcal{M}), \quad r_\psi \in [0, 1], \quad \kappa \in \mathrm{TSM}, \mathrm{PF}, \mathrm{WMF}, \tag{7}$$

where r_ψ represents a continuous success metric and κ denotes the categorical failure attribution. Specifically, Task-Scene Mismatch (TSM) identifies misalignments in initial semantic grounding; Policy Failure (PF) indicates suboptimal

control outputs from the VLA; and World-Model Failure (WMF) signals a divergence between the predicted dynamics and real-world physical constraints. EvoW then orchestrates Targeted Updates based on a predefined reliability threshold δ, ensuring that only the identified module undergoes refinement while keeping stable components intact:

Verified Experience Accumulation ($r_\psi \geq \delta$): High-confidence trajectories are integrated into the Verified Trace Set, serving as high-fidelity demonstrations for future self-imitation learning and router calibration.

Semantic Grounding Refinement ($\kappa = \text{TSM}$): If the failure stems from an irrelevant initial scene, the system updates the Router's memory $\mathcal{M}$ to refine the semantic-to-scene grounding manifold, preventing the recurrence of analogous grounding errors.

Modular Policy Plasticity ($\kappa = \text{PF}$): For policy-induced failures, the system triggers Autonomous Retraining for the specific VLA expert π_{m^*}. By utilizing the diagnostic feedback as a corrective signal, the expert's parameters are fine-tuned to align with the successful task manifold.

Dynamics Grounding Refinement ($\kappa = \text{WMF}$): When the simulation fails to capture valid physics, the world model's dynamics parameters are updated using the discrepancy between the rollout τ and the physical ground truth, thereby enhancing the framework's cognitive core.

This targeted evolutionary strategy allows EvoW to bypass the "one-size-fits-all" training bottleneck, achieving continuous compounding improvements in success rates through recursive, module-specific plasticity.

4 Experiments

In this section, we evaluate EvoW along three dimensions: **(1) skill-level performance**, including single-skill success rates (SR), iterative policy improvement, and compositional long-horizon tasks; **(2) task diversity and scene validity**, measuring whether generated tasks remain diverse while preserving grounded executability; and **(3) generative quality**, evaluating the visual fidelity and temporal consistency of imagined trajectories under large-scale rollouts.

4.1 Experimental Setup

To isolate the contributions of our method, we keep VLA backbones, world-model data sources, and evaluation splits fixed across compared variants. We sample tasks from skill families and use grounded initial scenes anchored to a shared offline repository. Across all compared variants, we fix the rollout horizon, task split, scene split, and trajectory budget per task. We first compare base policies and routed variants under identical initialization, and then run iterative refinement with matched update budgets per round to analyze cumulative improvements under a controlled setting.

The benchmark skill space contains five representative manipulation families, denoted as $\mathcal{S}_1$–$\mathcal{S}_5$: $\mathcal{S}_1$ (Pick-and-Place), $\mathcal{S}_2$ (Tool-Use), $\mathcal{S}_3$ (Reorientation),

Table 1: Policy success rates across five representative skill families and four VLA backbones, comparing base policies with EvoW routing. Values are reported as success rates. **EvoW*** denotes our method, and Gain(↑%) is computed relative to the corresponding base policy average.

Backbone	Setting	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$	Avg	Gain(↑%)
GR00T-N1.6	Base	0.38	0.24	0.18	0.28	0.20	0.256	–
	EvoW*	**0.48**	**0.50**	**0.46**	**0.42**	**0.48**	**0.468**	**↑82.8%**
PI-0.5	Base	0.32	0.14	0.24	0.18	0.28	0.232	–
	EvoW*	**0.48**	**0.28**	**0.38**	**0.44**	**0.48**	**0.412**	**↑77.6%**
PI-0	Base	0.40	0.22	0.26	0.32	0.22	0.284	–
	EvoW*	**0.62**	**0.38**	**0.40**	**0.40**	**0.50**	**0.460**	**↑62.0%**
PI-0-fast	Base	0.38	0.18	0.26	0.24	0.26	0.264	–
	EvoW*	**0.66**	**0.38**	**0.44**	**0.36**	**0.42**	**0.452**	**↑71.2%**

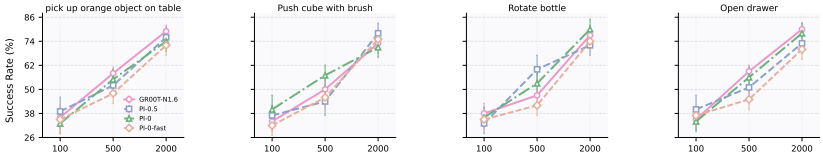


Fig. 3: Task-level refinement trends. Iterative updates consistently raise success rates, and variance bars indicate that gains are stable across runs.

$\mathcal{S}_4$ (Deformable-Object Manipulation), and $\mathcal{S}_5$ (Articulation). Comprehensive implementation details—including pre-training configurations, VLM-evaluator fine-tuning procedures, episode statistics, and rollout curation methods—are deferred to Appendix A.

4.2 Skill-Level Performance

We first investigate whether the routing mechanism independently enhances policy success under fixed model and data constraints, followed by an analysis of the cumulative benefits from iterative refinement. Table 1 details the static SR comparisons between the baseline and routed variants across four VLA backbones and the five skill families.

Ablation Results. As shown in Table 1, routing with **EvoW** consistently improves success rates across all four backbones. Averaged over backbones, SR increases from 0.259 to 0.448 (absolute +18.9 points, relative +73.0%). Per-backbone relative improvements range from +62.0% to +82.8%, indicating that the gain is stable rather than concentrated on a single backbone.

Policy Improvement Over Iterations. Figure 3 reports SR trends over refinement iterations on representative tasks. SR improves across tasks, indicating that diagnosis-conditioned updates in Stage 4 reduce recurrent failure patterns over time. The combined visualization in Figure 5 and qualitative rollouts in Figure 4 further show more stable multi-step execution in later iterations.

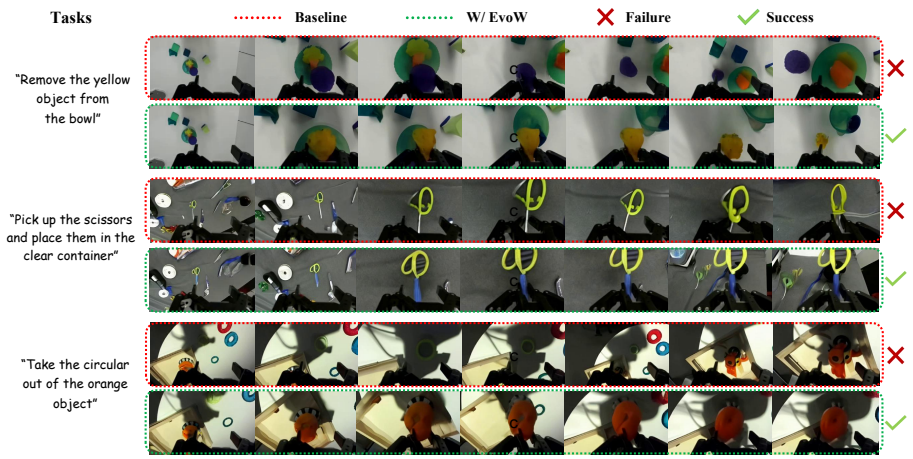


Fig. 4: Qualitative examples from our refinement loop. For each base policy shown in the top row, the bottom row presents the corresponding imagined rollouts in the grounded world model.



Fig. 5: Visualization of compositional long-horizon tasks. Early and late stages involve different skills, and success requires completing the full task sequence.

These results are consistent with the formulation that expert selection contributes immediate performance gains under fixed backbones, and diagnosis-conditioned refinement yields additional improvements over successive rounds by reducing recurrent execution failures.

Compositional Long-Horizon Tasks. Single-skill metrics cannot fully capture robustness under cross-skill transitions. We therefore evaluate compositional long-horizon tasks that chain two or three skills from $\mathcal{S}_1$–$\mathcal{S}_5$. **Results.** Figure 6 reports full-success and partial-success SR on 10 procedurally generated compositional tasks. Full-success SR requires completing all sub-goals with stable rollout–execution consistency, while partial-success SR requires completing at least one sub-goal without instability. Routed variants consistently enlarge the feasible SR range across backbones, especially under the full-success criterion, indicating better cross-stage coordination when tasks require skill switching. Task definitions and per-task statistics are provided in Appendix B.

4.3 Task diversity and scene validity

Beyond execution success, it is critical that EvoW can scale task generation while strictly adhering to grounding constraints to preserve scene validity. Table 2 benchmarks the language and scene diversity of our generated tasks against existing simulation environments, including RoboGen [39]. We report complemen-

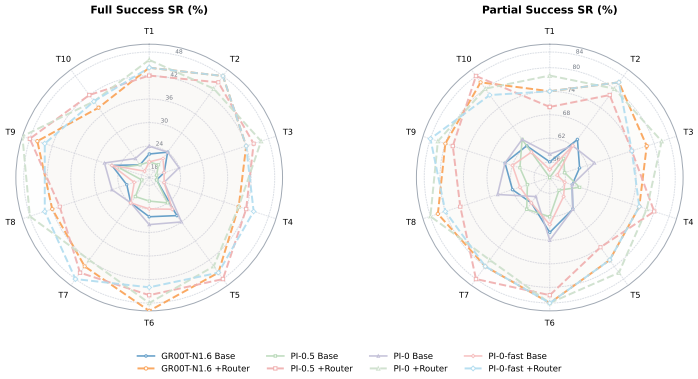


Fig. 6: Compositional long-horizon evaluation under full-completion and partial-completion criteria. Routed variants expand the performance envelope across backbones, indicating improved cross-stage robustness.

Table 2: Task and scene diversity comparison under executability constraints. The metrics indicate that EvoW preserves competitive diversity while maintaining scene validity for grounded deployment on Self-BLEU and CLIP metrics.

Metric	**EvoW**	RoboGen	Behavior-100	RLBench	MetaWorld	ManiSkill2	GenSim
Number of tasks	100	106	100	106	50	20	70
Task Description – Self-BLEU ↓	**0.281**	0.284	0.299	0.317	0.322	0.674	0.378
Task Description – Embedding Similarity (SentenceBert) ↓	0.312	**0.165**	0.210	0.200	0.263	0.194	0.288
Scene Image – Embedding Similarity (ViT) ↓	0.337	**0.193**	0.389	0.375	0.517	0.332	0.717
Scene Image – Embedding Similarity (CLIP) ↓	**0.754**	0.762	0.833	0.864	0.867	0.828	0.932

tary metrics widely used for text and visual diversity analysis: Self-BLEU [30], Sentence-BERT similarity [33], ViT similarity [10], and CLIP similarity [32].

As shown in Table 2, EvoW achieves the lowest Self-BLEU and CLIP similarity and remains competitive on ViT similarity. EvoW is weaker on Sentence-BERT similarity, which is expected because grounded executability constraints reduce free-form language variation. This trade-off is acceptable in our setting: the generated tasks are intended for physically realizable rollouts rather than unconstrained textual diversity.

4.4 Generative quality

We assess whether EvoW's architectural improvements translate to higher-fidelity rollout generation in terms of both visual appearance and temporal consistency, compared with representative world-model baselines such as WorldGym (WPE), IRASim, and Ctrl-World [15, 31, 45]. We use FID [20] and FVD [37] as standard image- and video-level fidelity metrics.

Ablation Results. Figure 7 shows that EvoW achieves lower FID and FVD than the compared baselines. The larger relative gain on FVD indicates stronger temporal consistency over long-horizon rollouts. Under view-conditioned settings, EvoW remains better than Ctrl-World for both third-view and wrist-view comparisons, consistent with improved rollout stability.

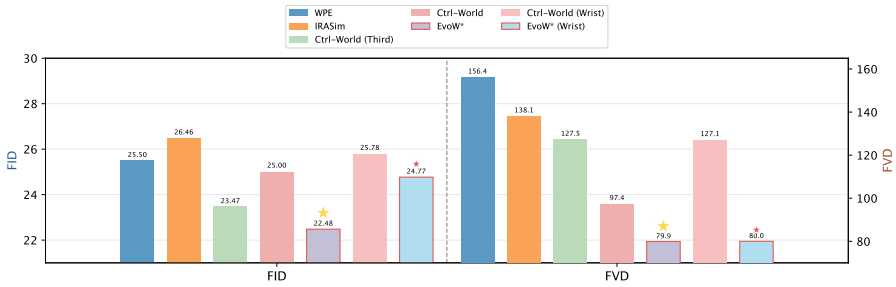


Fig. 7: Generative-quality comparison under a view-conditioned setting. EvoW achieves lower FID and FVD than the compared baselines in this setup. The larger reduction on FVD is consistent with improved temporal coherence, and view-matched expert outputs are associated with more stable rollout dynamics. Red and yellow stars mark EvoW in different views, with yellow indicating the best value under each metric.



Fig. 8: Qualitative temporal-consistency comparison for long-horizon rollouts. EvoW better preserves object identity, contact continuity, and motion progression across frames, consistent with stronger long-horizon world-model stability.

Qualitative evidence in Figure 8 is consistent with these quantitative trends: EvoW better preserves object identity and contact progression across time, while baseline rollouts more often show collapse behaviors such as object disappearance and broken motion continuity. Overall, EvoW improves both video-level fidelity and long-horizon temporal stability.

5 Conclusion

We present EvoW, a framework that maps open-ended task semantics to verified training trajectories and executable manipulation policies. EvoW grounds each task in task-consistent real-world scenes, synthesizes scene-consistent episodes with a decoupled world model, and closes the loop via VLM-based routing and verification. This design decouples shared dynamics from task-specialized skills and enables targeted updates through failure diagnosis.

Our results indicate that scene grounding and automated VLM feedback are complementary for scaling open-ended manipulation, yielding reliable task execution and continual improvement over iterations. Future work includes (i) expanding and curating the anchor repository to better match open-ended tasks, (ii) improving long-horizon world-model fidelity with uncertainty-aware rollout and filtering, and (iii) strengthening routing, verification, and failure attribution with task-aware critics and calibrated confidence signals.

References

1. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R.J., Jeffrey, K., Jesmonth, S., Joshi, N.J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., Zeng, A.: Do as i can, not as i say: Grounding language in robotic affordances. In: CORL (2022)
2. Asadi, K., Misra, D., Kim, S., Littman, M.L.: Combating the compounding-error problem with a multi-step model (2019), https://arxiv.org/abs/1905.13320
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report (2025), https://arxiv.org/abs/2511.21631
4. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L.X., Tanner, J., Vuong, Q., Walling, A., Wang, H., Zhilinsky, U.: π_0: A vision-language-action flow model for general robot control (2026), https://arxiv.org/abs/2410.24164
5. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M.G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.W.E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., Zitkovich, B.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: CORL (2023)
6. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N.J., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, K.H., Levine, S., Lu, Y., Malla, U., Manjunath, D., Mordatch, I., Nachum, O., Parada, C., Peralta, J., Perez, E., Pertsch, K., Quiambao, J., Rao, K., Ryoo, M., Salazar, G., Sanketi, P., Sayed, K., Singh, J., Sontakke, S., Stone, A., Tan, C., Tran, H., Vanhoucke, V., Vega, S., Vuong, Q., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., Zitkovich, B.: Rt-1: Robotics transformer for real-world control at scale. In: RSS (2023)
7. Cen, J., Yu, C., Yuan, H., Jiang, Y., Huang, S., Guo, J., Li, X., Song, Y., Luo, H., Wang, F., Zhao, D., Chen, H.: Worldvla: Towards autoregressive action world model (2025), https://arxiv.org/abs/2506.21539
8. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. In: RSS (2023)
9. Collaboration, E., O'Neill, A., Rehman, A., Gupta, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., Tung, A.,

Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Gupta, A., Wang, A., Kolobov, A., Singh, A., Garg, A., Kembhavi, A., Xie, A., Brohan, A., Raffin, A., Sharma, A., Yavary, A., Jain, A., Balakrishna, A., Wahid, A., Burgess-Limerick, B., Kim, B., Schölkopf, B., Wulfe, B., Ichter, B., Lu, C., Xu, C., Le, C., Finn, C., Wang, C., Xu, C., Chi, C., Huang, C., Chan, C., Agia, C., Pan, C., Fu, C., Devin, C., Xu, D., Morton, D., Driess, D., Chen, D., Pathak, D., Shah, D., Büchler, D., Jayaraman, D., Kalashnikov, D., Sadigh, D., Johns, E., Foster, E., Liu, F., Ceola, F., Xia, F., Zhao, F., Frujeri, F.V., Stulp, F., Zhou, G., Sukhatme, G.S., Salhotra, G., Yan, G., Feng, G., Schiavi, G., Berseth, G., Kahn, G., Yang, G., Wang, G., Su, H., Fang, H.S., Shi, H., Bao, H., Amor, H.B., Christensen, H.I., Furuta, H., Bharadhwaj, H., Walke, H., Fang, H., Ha, H., Mordatch, I., Radosavovic, I., Leal, I., Liang, J., Abou-Chakra, J., Kim, J., Drake, J., Peters, J., Schneider, J., Hsu, J., Vakil, J., Bohg, J., Bingham, J., Wu, J., Gao, J., Hu, J., Wu, J., Wu, J., Sun, J., Luo, J., Gu, J., Tan, J., Oh, J., Wu, J., Lu, J., Yang, J., Malik, J., Silvério, J., Hejna, J., Booher, J., Tompson, J., Yang, J., Salvador, J., Lim, J.J., Han, J., Wang, K., Rao, K., Pertsch, K., Hausman, K., Go, K., Gopalakrishnan, K., Goldberg, K., Byrne, K., Oslund, K., Kawaharazuka, K., Black, K., Lin, K., Zhang, K., Ehsani, K., Lekkala, K., Ellis, K., Rana, K., Srinivasan, K., Fang, K., Singh, K.P., Zeng, K.H., Hatch, K., Hsu, K., Itti, L., Chen, L.Y., Pinto, L., Fei-Fei, L., Tan, L., Fan, L.J., Ott, L., Lee, L., Weihs, L., Chen, M., Lepert, M., Memmel, M., Tomizuka, M., Itkina, M., Castro, M.G., Spero, M., Du, M., Ahn, M., Yip, M.C., Zhang, M., Ding, M., Heo, M., Srirama, M.K., Sharma, M., Kim, M.J., Irshad, M.Z., Kanazawa, N., Hansen, N., Heess, N., Joshi, N.J., Suenderhauf, N., Liu, N., Palo, N.D., Shafiullah, N.M.M., Mees, O., Kroemer, O., Bastani, O., Sanketi, P.R., Miller, P.T., Yin, P., Wohlhart, P., Xu, P., Fagan, P.D., Mitrano, P., Sermanet, P., Abbeel, P., Sundaresan, P., Chen, Q., Vuong, Q., Rafailov, R., Tian, R., Doshi, R., Martín-Martín, R., Baijal, R., Scalise, R., Hendrix, R., Lin, R., Qian, R., Zhang, R., Mendonca, R., Shah, R., Hoque, R., Julian, R., Bustamante, S., Kirmani, S., Levine, S., Lin, S., Moore, S., Bahl, S., Dass, S., Sonawani, S., Tulsiani, S., Song, S., Xu, S., Haldar, S., Karamcheti, S., Adebola, S., Guist, S., Nasiriany, S., Schaal, S., Welker, S., Tian, S., Ramamoorthy, S., Dasari, S., Belkhale, S., Park, S., Nair, S., Mirchandani, S., Osa, T., Gupta, T., Harada, T., Matsushima, T., Xiao, T., Kollar, T., Yu, T., Ding, T., Davchev, T., Zhao, T.Z., Armstrong, T., Darrell, T., Chung, T., Jain, V., Kumar, V., Vanhoucke, V., Guizilini, V., Zhan, W., Zhou, W., Burgard, W., Chen, X., Chen, X., Wang, X., Zhu, X., Geng, X., Liu, X., Liangwei, X., Li, X., Pang, Y., Lu, Y., Ma, Y.J., Kim, Y., Chebotar, Y., Zhou, Y., Zhu, Y., Wu, Y., Xu, Y., Wang, Y., Bisk, Y., Dou, Y., Cho, Y., Lee, Y., Cui, Y., Cao, Y., Wu, Y.H., Tang, Y., Zhu, Y., Zhang, Y., Jiang, Y., Li, Y., Li, Y., Iwasawa, Y., Matsuo, Y., Ma, Z., Xu, Z., Cui, Z.J., Zhang, Z., Fu, Z., Lin, Z.: Open x-embodiment: Robotic learning datasets and rt-x models (2025), https://arxiv.org/abs/2310.08864

10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
11. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., Florence, P.: Palm-e: An embodied multimodal language model. In: ICML (2023)

12. Fang, H.S., Fang, H., Tang, Z., Liu, J., Wang, C., Wang, J., Zhu, H., Lu, C.: Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot (2023), https://arxiv.org/abs/2307.00595
13. Gao, S., Zhou, S., Du, Y., Zhang, J., Gan, C.: Adaworld: Learning adaptable world models with latent actions. In: ICML (2025)
14. Garrido, Q., Nagarajan, T., Terver, B., Ballas, N., LeCun, Y., Rabbat, M.: Learning latent action world models in the wild (2026), https://arxiv.org/abs/2601.05230
15. Guo, Y., Shi, L.X., Chen, J., Finn, C.: Ctrl-world: A controllable generative world model for robot manipulation. In: ICLR (2026), https://iclr.cc/virtual/2026/poster/10011332
16. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. In: ICLR (2020)
17. Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J.: Learning latent dynamics for planning from pixels. In: ICML (2019)
18. Hafner, D., Lillicrap, T., Norouzi, M., Ba, J.: Mastering atari with discrete world models. In: ICLR (2021)
19. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse domains through world models (2024), https://arxiv.org/abs/2301.04104
20. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf
21. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi_{0.5}$: a vision-language-action model with open-world generalization (2025), https://arxiv.org/abs/2504.16054
22. Jang, J., Ye, S., Lin, Z., Xiang, J., Bjorck, J., Fang, Y., Hu, F., Huang, S., Kundalia, K., Lin, Y.C., Magne, L., Mandlekar, A., Narayan, A., Tan, Y.L., Wang, G., Wang, J., Wang, Q., Xu, Y., Zeng, X., Zheng, K., Zheng, R., Liu, M.Y., Zettlemoyer, L., Fox, D., Kautz, J., Reed, S., Zhu, Y., Fan, L.: Dreamgen: Unlocking generalization in robot learning through video world models (2025), https://arxiv.org/abs/2505.12705
23. Jiang, Y., Gu, Y., Tsang, I.W., Shou, M.Z.: Olaf-world: Orienting latent actions for video world modeling (2026), https://arxiv.org/abs/2602.10104
24. Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., Levine, S.: Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation (2018), https://arxiv.org/abs/1806.10293
25. Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M.K., Chen, L.Y., Ellis, K., Fagan, P.D., Hejna, J., Itkina, M., Lepert, M., Ma, Y.J., Miller, P.T., Wu, J., Belkhale, S., Dass, S., Ha, H., Jain, A., Lee, A., Lee, Y., Memmel, M., Park, S., Radosavovic, I., Wang, K., Zhan, A., Black, K., Chi, C., Hatch, K.B., Lin, S., Lu, J., Mercat, J., Rehman, A., Sanketi, P.R., Sharma, A., Simpson, C., Vuong, Q., Walke, H.R., Wulfe, B., Xiao, T., Yang, J.H., Yavary, A., Zhao, T.Z., Agia, C., Baijal, R., Castro, M.G., Chen, D., Chen, Q., Chung, T., Drake, J., Foster, E.P., Gao, J., Guizilini, V., Herrera, D.A., Heo,

M., Hsu, K., Hu, J., Irshad, M.Z., Jackson, D., Le, C., Li, Y., Lin, K., Lin, R., Ma, Z., Maddukuri, A., Mirchandani, S., Morton, D., Nguyen, T., O'Neill, A., Scalise, R., Seale, D., Son, V., Tian, S., Tran, E., Wang, A.E., Wu, Y., Xie, A., Yang, J., Yin, P., Zhang, Y., Bastani, O., Berseth, G., Bohg, J., Goldberg, K., Gupta, A., Gupta, A., Jayaraman, D., Lim, J.J., Malik, J., Martín-Martín, R., Ramamoorthy, S., Sadigh, D., Song, S., Wu, J., Yip, M.C., Zhu, Y., Kollar, T., Levine, S., Finn, C.: Droid: A large-scale in-the-wild robot manipulation dataset. In: RSS (2024)
26. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: Openvla: An open-source vision-language-action model. In: CORL (2024)
27. Lambert, N., Pister, K., Calandra, R.: Investigating compounding prediction errors in learned dynamics models (2022), https://arxiv.org/abs/2203.09637
28. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. JMLR **17**(39), 1–40 (2016)
29. NVIDIA, Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L.J., Fang, Y., Fox, D., Hu, F., Huang, S., Jang, J., Jiang, Z., Kautz, J., Kundalia, K., Lao, L., Li, Z., Lin, Z., Lin, K., Liu, G., Llontop, E., Magne, L., Mandlekar, A., Narayan, A., Nasiriany, S., Reed, S., Tan, Y.L., Wang, G., Wang, Z., Wang, J., Wang, Q., Xiang, J., Xie, Y., Xu, Y., Xu, Z., Ye, S., Yu, Z., Zhang, A., Zhang, H., Zhao, Y., Zheng, R., Zhu, Y.: Gr00t n1: An open foundation model for generalist humanoid robots (2025), https://arxiv.org/abs/2503.14734
30. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of ACL (2002)
31. Quevedo, J., Sharma, A.K., Sun, Y., Suryavanshi, V., Liang, P., Yang, S.: Worldgym: World model as an environment for policy evaluation (2025), https://arxiv.org/abs/2506.00613
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
33. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Conf. Empirical Methods Natural Lang. Process. (2019)
34. Sarch, G., Wu, Y., Tarr, M.J., Fragkiadaki, K.: Open-ended instructable embodied agents with memory-augmented large language models. In: EMNLP (2023)
35. Shridhar, M., Manuelli, L., Fox, D.: Cliport: What and where pathways for robotic manipulation. In: CORL (2021)
36. Torabi, F., Warnell, G., Stone, P.: Behavioral cloning from observation (2018), https://arxiv.org/abs/1805.01954
37. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges (2018), https://arxiv.org/abs/1812.01717
38. Wang, Y., Zhang, F., Zhan, D.C., Zhao, L., Wang, K., Bian, J.: Co-evolving latent action world models (2025), https://arxiv.org/abs/2510.26433
39. Wang, Y., Xian, Z., Chen, F., Wang, T.H., Wang, Y., Fragkiadaki, K., Erickson, Z., Held, D., Gan, C.: Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. In: ICML (2024)
40. Wang, Z., Shi, C., Hu, J., Rohling, K., Martín-Martín, R., Zhang, A., Stone, P.: Factored latent action world models (2026), https://arxiv.org/abs/2602.16229
41. Wu, S., Luo, X., Zhang, J., Xie, J., Song, J., Shen, H.T., Gao, L.: Policy contrastive decoding for robotic foundation models (2025), https://arxiv.org/abs/2505.13255

42. Yang, J., Lin, K., Li, J., Zhang, W., Lin, T., Wu, L., Su, Z., Zhao, H., Zhang, Y.Q., Chen, L., Luo, P., Yue, X., Li, H.: Rise: Self-improving robot policy with compositional world model (2026), https://arxiv.org/abs/2602.11075
43. Zeng, A., Florence, P., Tompson, J., Welker, S., Chien, J., Attarian, M., Armstrong, T., Krasin, I., Duong, D., Wahid, A., Sindhwani, V., Lee, J.: Transporter networks: Rearranging the visual world for robotic manipulation. In: CORL (2020)
44. Zhao, T.Z., Tompson, J., Driess, D., Florence, P., Ghasemipour, K., Finn, C., Wahid, A.: Aloha unleashed: A simple recipe for robot dexterity. In: CORL (2024)
45. Zhu, F., Wu, H., Guo, S., Liu, Y., Cheang, C., Kong, T.: Irasim: A fine-grained world model for robot manipulation (2024), https://arxiv.org/abs/2406.14540
46. Zhu, M., Zhu, Y., Li, J., Wen, J., Xu, Z., Liu, N., Cheng, R., Shen, C., Peng, Y., Feng, F., Tang, J.: Scaling diffusion policy in transformer to 1 billion parameters for robotic manipulation. In: ICRA (2025)

Acknowledgements

This work was supported by the National Science and Technology Major Project under Grant 2022ZD0114903.

The authors also acknowledge support from the Tsinghua University Initiative Scientific Research Program (Student Academic Research Advancement Program) and the Beijing Natural Science Foundation Undergraduate "Qiyan Program" (Grant No. 26QY0429). Sen Cui additionally acknowledges support from the Science and Technology Project of Beijing Municipal Science & Technology Commission (Grant No. Z251100008125030) and the Shuimu Scholar Program of Tsinghua University.

Appendix

A Experimental Details

A.1 Compact Implementation Notes

Given a generated instruction ℓ, EvoW retrieves a grounded initial scene from $\mathcal{D}$, scores candidate experts with imagined rollouts, and keeps an executed trajectory only when the VLM evaluator judges it reliable.

Formally, the grounded initial scene is selected by nearest-neighbor retrieval in the joint language–vision space,

$$o_0 = \arg\max_{o \in \mathcal{D}} \mathrm{sim}(f_\ell(\ell), f_o(o)), \tag{8}$$

where $f_\ell(\cdot)$ and $f_o(\cdot)$ denote the text and observation encoders, and $\mathrm{sim}(\cdot,\cdot)$ is cosine similarity. This form makes explicit that grounding is a retrieval problem rather than a free-form generation step.

For expert $\pi_m \in \mathit{\Pi}$, the imagined rollout of horizon H is

$$\hat{\tau}_m = \{\hat{o}_t, a_t\}_{t=0}^{H-1}, \qquad a_t = \pi_m(\hat{o}_t, \ell), \qquad \hat{o}_{t+1} \sim p_\theta(\cdot \mid \hat{o}_t, a_t), \tag{9}$$

where p_θ is the video world model. The router combines the routing score q from the main text and a rollout score,

$$m^\star = \arg\max_m \left[\lambda\, q(m \mid o_0, \ell, \mathcal{M}) + (1-\lambda)\, s_{\mathrm{wm}}(\hat{\tau}_m)\right], \tag{10}$$

where $\lambda \in [0,1]$ balances semantic matching and imagined physical validity.

Let $(r_\psi, \kappa) = \mathit{\Psi}(\tau, \ell, \mathcal{M})$ denote the evaluator output in the main text, where $r_\psi \in [0,1]$ is the scalar success score. A rollout is added to the verified buffer only if

$$y(\tau, \ell) = \mathbb{1}[r_\psi \geq \delta], \tag{11}$$

where δ is the reliability threshold. Failed cases also receive a coarse diagnosis tag from $\{\mathrm{TSM}, \mathrm{PF}, \mathrm{WMF}\}$.

A.2 Experimental Protocol

- **Data and evaluator.** Following Ctrl-World [15], the world model uses DROID [25]. The evaluator is Qwen3-VL-4B-Instruct fine-tuned on real trajectories and manually annotated synthetic trajectories [3].
- **Skills and backbones.** We study five skill families ($\mathcal{S}_1$–$\mathcal{S}_5$) and four VLA backbones: GR00T-N1.6, PI-0.5, PI-0, and PI-0-fast.
- **Backbone adaptation.** Each expert is adapted on the matched skill/scene subset from the official checkpoint with a unified budget of 5k optimization steps. For GR00T-N1.6, we follow the official training configuration with 8 GPUs and a global batch size of 64. For the PI series (PI-0.5, PI-0, and

PI-0-fast), we use the OpenPI training entry on a DROID/LeRobot-style fine-tuning dataset, with the default chunk-000 split, 8 data-loading workers, no shuffling, and joint-velocity supervision enabled. We preserve the released backbone-specific settings for optimizer, learning-rate schedule, batch construction, input preprocessing, action representation, and loss design, and only replace the training data with our matched skill/scene subset while normalizing the total budget for a fair comparison. Concretely, PI-0.5 uses action dimension 32 and horizon 15 with batch size 16, PI-0 uses horizon 10 with batch size 8, and PI-0-fast uses action dimension 8, horizon 10, and maximum token length 180 with batch size 8.

- **Evaluation budget.** Unless otherwise noted, each reported setting uses 50 interaction episodes. Static SR uses 50 sampled task–scene pairs per skill family and backbone.
- **Refinement checkpoints.** Intermediate results are reported after 100, 500, and 2000 rounds of closed-loop refinement during synthetic-data post-training. Across all experiments, the pipeline yields 7,042 episodes that pass verification.
- **Curation.** Failed rollouts are tagged as grounding mismatch, expert-selection mismatch, or world-model failure, which trigger module-specific updates.

A.3 Evaluation Metrics and Reporting Conventions

- **Single-skill SR.** Table 1 reports success rate under fixed budgets.
- **Iterative-refinement SR.** Fig. 3 reports mean±std across repeated runs.
- **Compositional SR.** Full-success requires all sub-goals; partial-success requires at least one sub-goal.
- **Diversity and generation.** We report Self-BLEU, Sentence-BERT, ViT, CLIP, FID, and FVD.

Unless otherwise noted, each reported setting uses 50 interaction episodes. Table 1 uses $N = 50$ task–scene pairs per backbone–skill pair. Table 3 reports checkpoints at 100/500/2000 closed-loop refinement rounds; these numbers are refinement iterations, not sample counts. At each checkpoint, we evaluate 5 parallel experimental groups, each with 100 stochastic samplings, and report mean±std as integer percentages. Tables 5 and 6 report task-wise percentages on 10 fixed compositional tasks. Diversity metrics use the full generated set, and FID/FVD use the shared rollout set.

For completeness, the reported success-rate metrics can be written in a unified form. Given N evaluation episodes and binary success labels $y_i \in \{0, 1\}$, the single-skill success rate is

$$\mathrm{SR} = \frac{1}{N} \sum_{i=1}^{N} y_i. \tag{12}$$

For iterative-refinement plots with R repeated runs (here $R = 5$, corresponding to the 5 parallel experimental groups, each evaluated with 100 stochastic

Table 3: Per-task success-rate statistics (mean±std, %) based on 100 samplings and 5 parallel experimental groups at each checkpoint; all values are rounded to integer percentages.

Task	Backbone	Round=100	Round=500	Round=2000
Place mug on coaster (S1)	GR00T-N1.6	36±5	58±3	79±3
Place mug on coaster (S1)	PI-0.5	39±7	52±5	76±5
Place mug on coaster (S1)	PI-0	33±5	55±5	74±5
Place mug on coaster (S1)	PI-0-fast	35±7	48±5	72±5
Push cube with spatula (S2)	GR00T-N1.6	34±5	50±5	73±5
Push cube with spatula (S2)	PI-0.5	37±5	44±7	78±5
Push cube with spatula (S2)	PI-0	40±7	57±5	71±5
Push cube with spatula (S2)	PI-0-fast	32±5	46±5	75±5
Rotate bottle upright (S3)	GR00T-N1.6	38±5	47±5	77±5
Rotate bottle upright (S3)	PI-0.5	33±5	60±7	72±5
Rotate bottle upright (S3)	PI-0	36±5	53±5	80±5
Rotate bottle upright (S3)	PI-0-fast	35±5	42±5	74±5
Open drawer, place tape (S5)	GR00T-N1.6	35±5	59±3	80±3
Open drawer, place tape (S5)	PI-0.5	40±7	51±5	73±5
Open drawer, place tape (S5)	PI-0	34±5	56±5	78±5
Open drawer, place tape (S5)	PI-0-fast	37±5	45±5	70±5

samplings at a checkpoint), the displayed statistics are

$$\mu_{\mathrm{SR}} = \frac{1}{R}\sum_{r=1}^{R} \mathrm{SR}^{(r)}, \qquad \sigma_{\mathrm{SR}} = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left(\mathrm{SR}^{(r)} - \mu_{\mathrm{SR}}\right)^2}. \tag{13}$$

For a compositional task with m_i sub-goals and sub-goal completion indicators $z_{ij} \in \{0, 1\}$, the two appendix criteria are

$$\mathrm{SR}_{\mathrm{full}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\sum_{j=1}^{m_i} z_{ij} = m_i\right], \qquad \mathrm{SR}_{\mathrm{partial}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[\sum_{j=1}^{m_i} z_{ij} \geq 1\right]. \tag{14}$$

A.4 Per-Task SR Statistics

Table 3 reports mean±std SR (%) used in the Sec. 4.2 per-task refinement plot, where 100/500/2000 indicate closed-loop refinement rounds during synthetic-data post-training.

B Compositional Long-Horizon Tasks

B.1 Experimental Setup

We evaluate 10 compositional tasks built from two skill families in $\mathcal{S}_1$–$\mathcal{S}_5$. Each task is tested on four VLA backbones with baseline and routed variants. We report full-success SR and partial-success SR, matching Fig. 6.

B.2 Task Definitions and Skill Composition

Table 4 lists the 10 compositional tasks in our long-horizon benchmark.

Table 4: Task definitions for the compositional long-horizon benchmark.

Task	Instruction (short form)	Skill composition
T1	Pick up the mug and turn the handle to the front.	$\mathcal{S}_1 \to \mathcal{S}_3$
T2	Open the drawer and place the tape on the table.	$\mathcal{S}_5 \to \mathcal{S}_1$
T3	Fold the cloth once and turn the folded side forward.	$\mathcal{S}_4 \to \mathcal{S}_3$
T4	Turn the bottle upright and place it in the tray.	$\mathcal{S}_3 \to \mathcal{S}_1$
T5	Use the spatula to pull the block closer, then place it in the bowl.	$\mathcal{S}_2 \to \mathcal{S}_1$
T6	Open the box lid and put the sponge inside.	$\mathcal{S}_5 \to \mathcal{S}_1$
T7	Pick up the cup and rotate it upright on the table.	$\mathcal{S}_1 \to \mathcal{S}_3$
T8	Gather the cloth and place it in the basket.	$\mathcal{S}_4 \to \mathcal{S}_1$
T9	Use the spoon to pull the drawer out a little, then open it fully.	$\mathcal{S}_2 \to \mathcal{S}_5$
T10	Open the plastic box and place the marker inside.	$\mathcal{S}_5 \to \mathcal{S}_1$

B.3 Per-Task Numerical Results

Tables 5 and 6 report full-success and partial-success SR for all 10 tasks.

Table 5: Full-success rates (%) on compositional long-horizon tasks.

Task	GR00T-N1.6 Base	**GR00T-N1.6 W/ EvoW**	PI-0.5 Base	**PI-0.5 W/ EvoW**	PI-0 Base	**PI-0 W/ EvoW**	PI-0-fast Base	**PI-0-fast W/ EvoW**
T1	22	**44**	20	**42**	24	**46**	20	**44**
T2	24	**48**	20	**46**	24	**44**	22	**48**
T3	20	**42**	18	**44**	24	**46**	20	**42**
T4	18	**40**	18	**42**	20	**40**	20	**44**
T5	28	**46**	24	**48**	30	**44**	26	**46**
T6	26	**50**	22	**46**	28	**48**	24	**44**
T7	24	**44**	22	**46**	24	**42**	24	**48**
T8	22	**42**	18	**40**	26	**48**	20	**44**
T9	26	**46**	24	**48**	28	**50**	26	**44**
T10	20	**38**	20	**42**	22	**40**	18	**40**

Table 6: Partial-success rates (%) on compositional long-horizon tasks.

Task	GR00T-N1.6 Base	**GR00T-N1.6 W/ EvoW**	PI-0.5 Base	**PI-0.5 W/ EvoW**	PI-0 Base	**PI-0 W/ EvoW**	PI-0-fast Base	**PI-0-fast W/ EvoW**
T1	56	**74**	52	**70**	58	**78**	54	**74**
T2	64	**82**	58	**78**	62	**80**	62	**82**
T3	60	**78**	56	**74**	64	**82**	54	**74**
T4	58	**76**	60	**80**	58	**78**	56	**76**
T5	62	**78**	56	**74**	62	**82**	58	**78**
T6	66	**84**	62	**82**	68	**84**	64	**84**
T7	60	**80**	62	**84**	58	**78**	60	**80**
T8	62	**82**	58	**76**	66	**84**	60	**80**
T9	64	**80**	60	**78**	64	**82**	62	**84**
T10	62	**82**	64	**84**	64	**80**	60	**78**

B.4 Discussion

Partial-success SR is consistently higher than full-success SR, indicating that many failures occur after one sub-goal has already been completed. EvoW improves both metrics, suggesting gains in both sub-goal entry and cross-skill transition stability.

C Supplementary Notes for Sec. 4

C.1 Notes on Task Diversity and Scene Validity

Table 2 should be read together with the grounding mechanism in Sec. 3. EvoW balances linguistic diversity with scene executability, so Self-BLEU and CLIP/ViT are the most informative metrics in our setting.

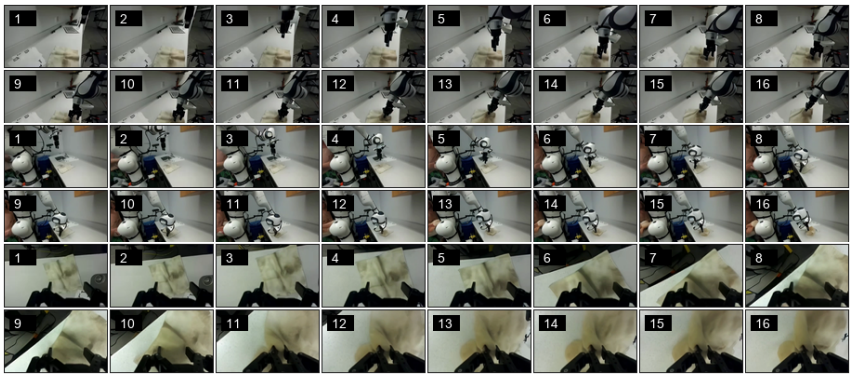
C.2 Notes on Generative Quality

FID and FVD are computed under the same view-conditioned setup as the main text. Better FVD supports the claim that EvoW improves long-horizon rollout stability.

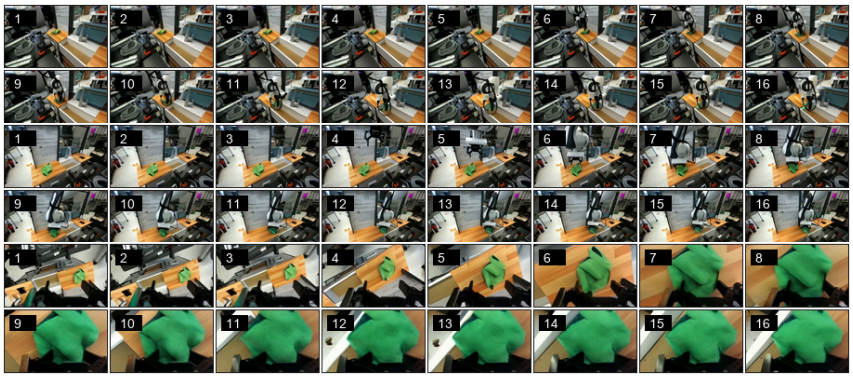
D Qualitative Successful Demo Rollouts

This section visualizes each successful demo by linearly sampling 16 frames from the rollout video. For each demo, we show the three camera streams without extra in-image labels.

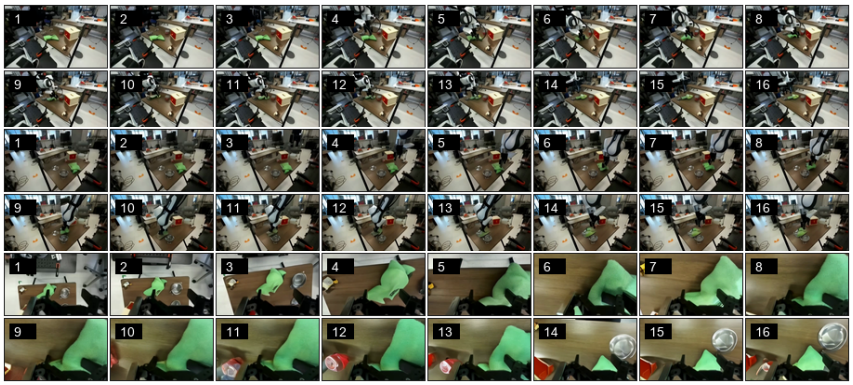
Wipe the table with the white cloth.



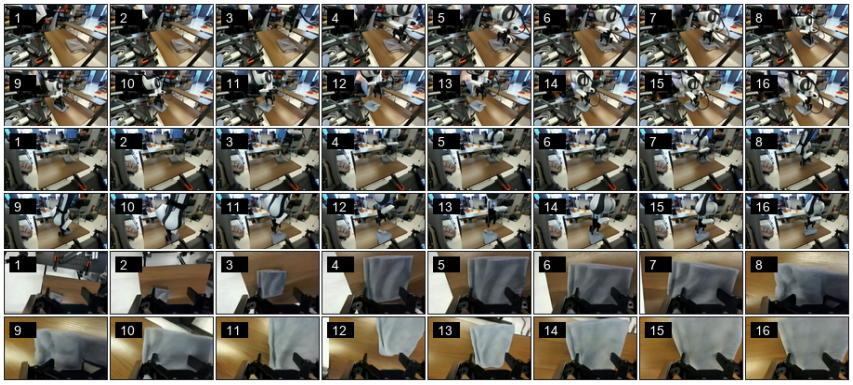
Shift the green cloth to the right.



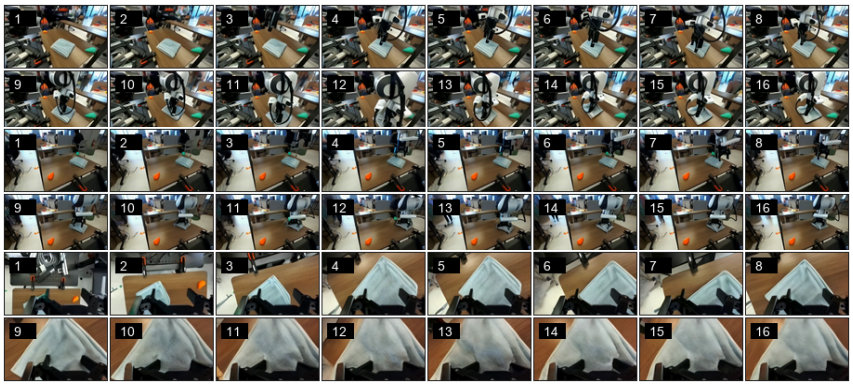
Shift the green towel to the right of the lid.



Reposition the cloth.



Wipe the surface with the cloth.

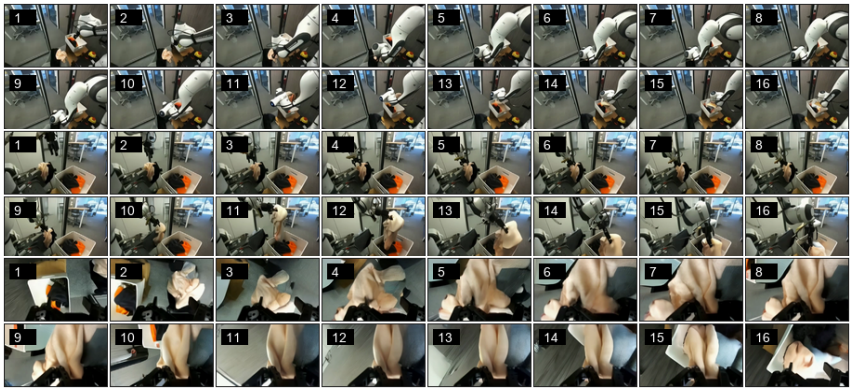


Clean the table with the towel.

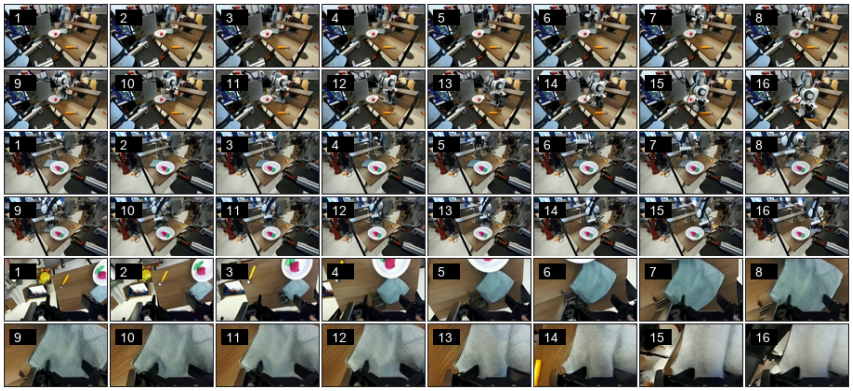




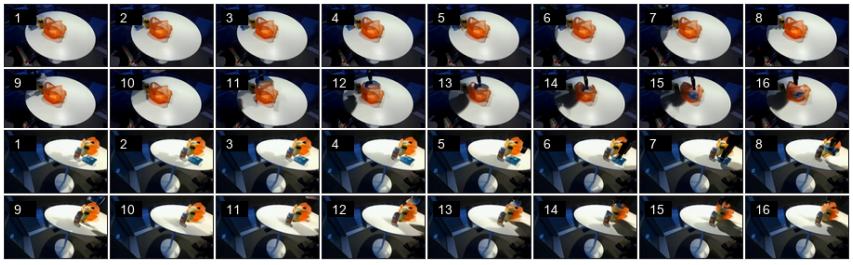
Lift the light coloured cloth from the chair and put it in the container.

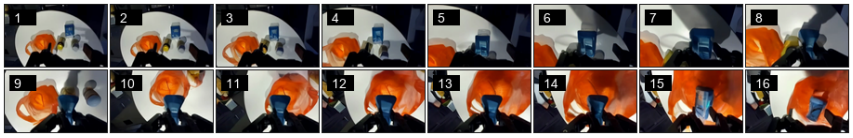


Slide the towel to the right on the desk to a new position.

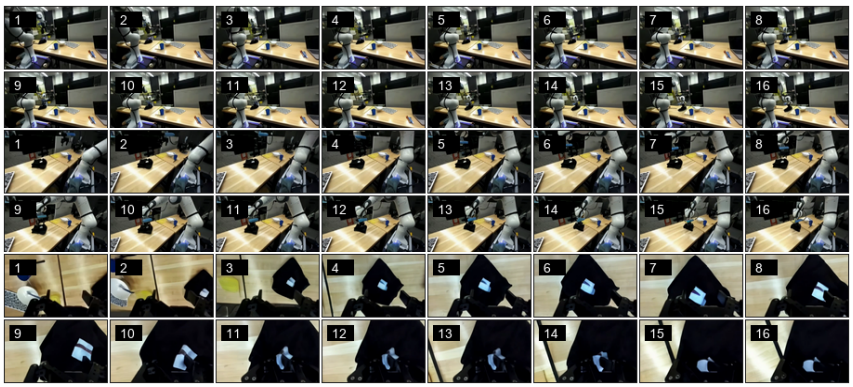


Place the blue container in the plastic bag.





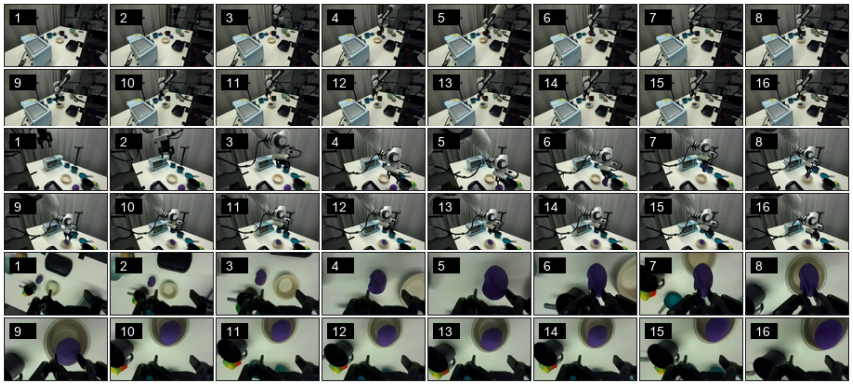
Wipe the table with the towel.



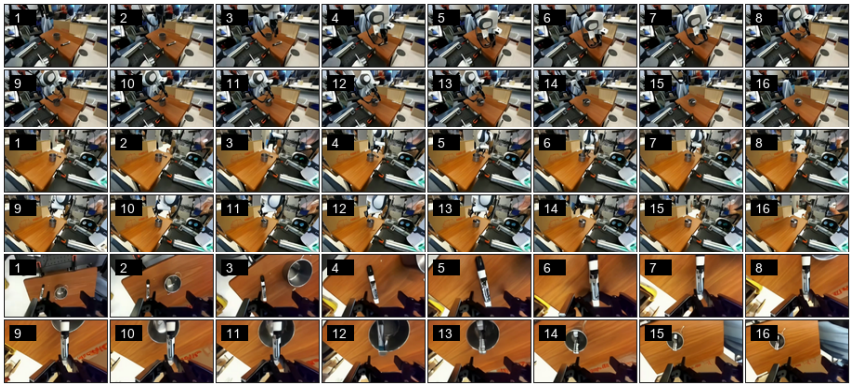
Wipe the table surface with the white towel.



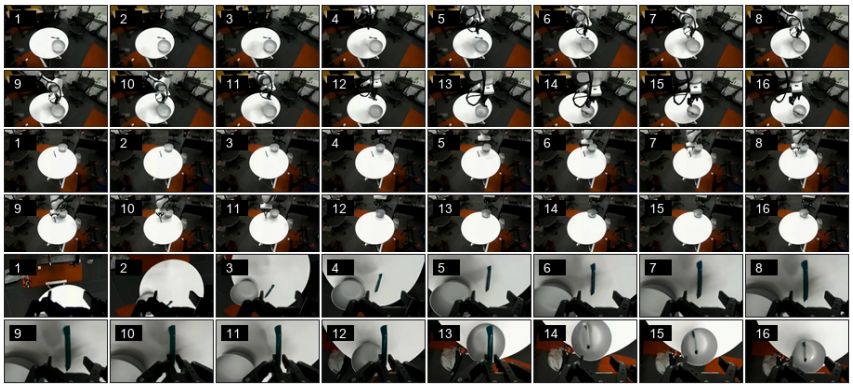
Place the purple plush toy in the white bowl.



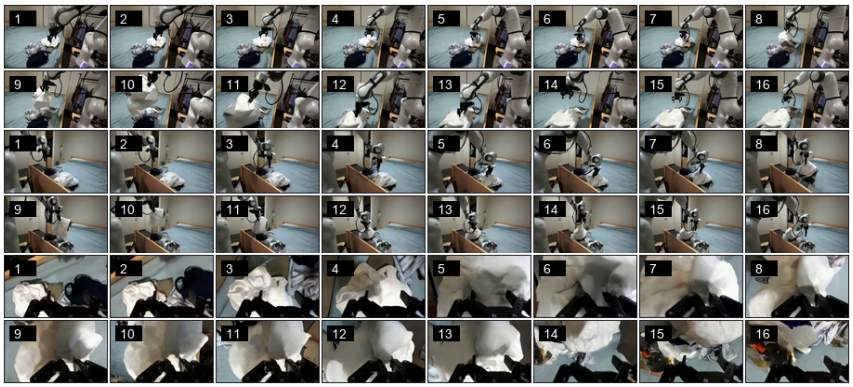
Place the marker in the pot.



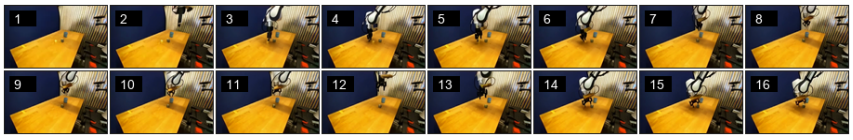
Place the marker inside the white bowl.

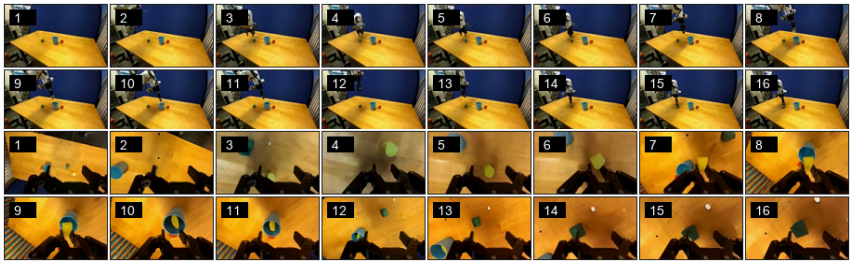


Move one white object from the right heap to the left, then get the blue-white object and fold it on top of the left heap.



Place the yellow block in the blue cup.

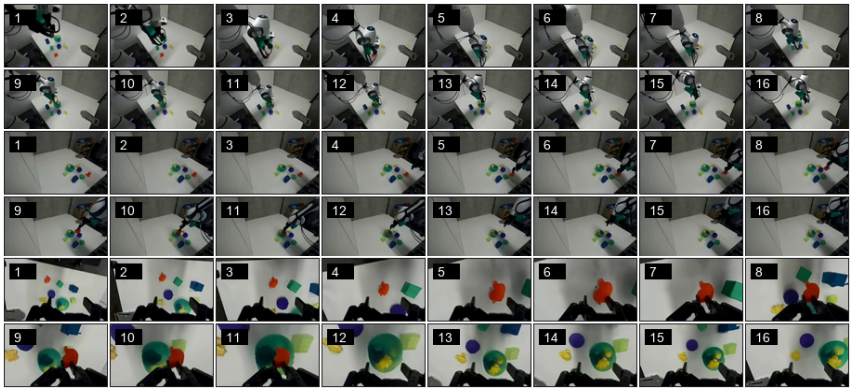




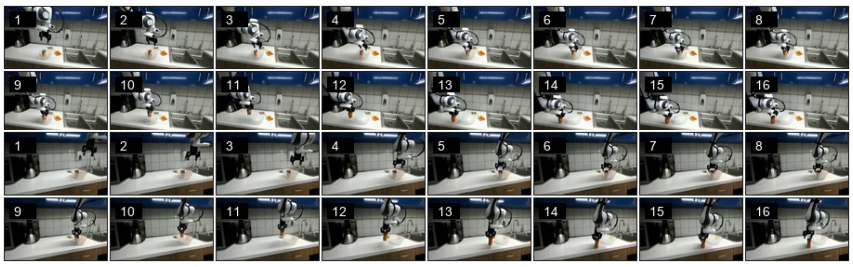
Lift the cup from the table and pour some of its contents in the orange bowl.

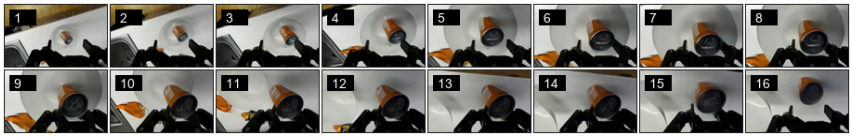


Place the orange thing into the bowl.

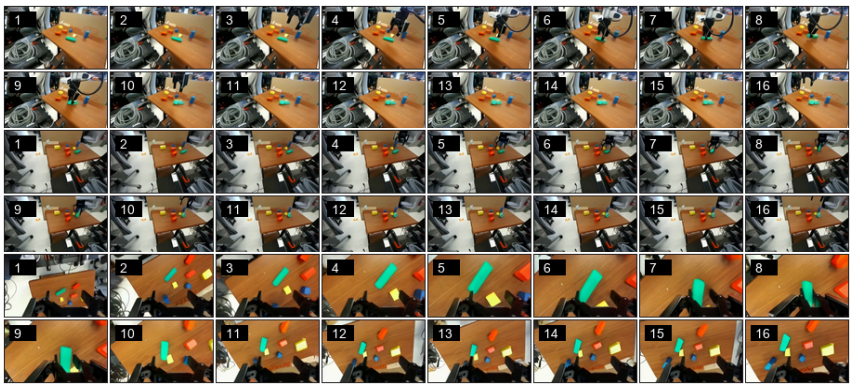


Take the orange can out of the plastic bowl and put it on the counter.

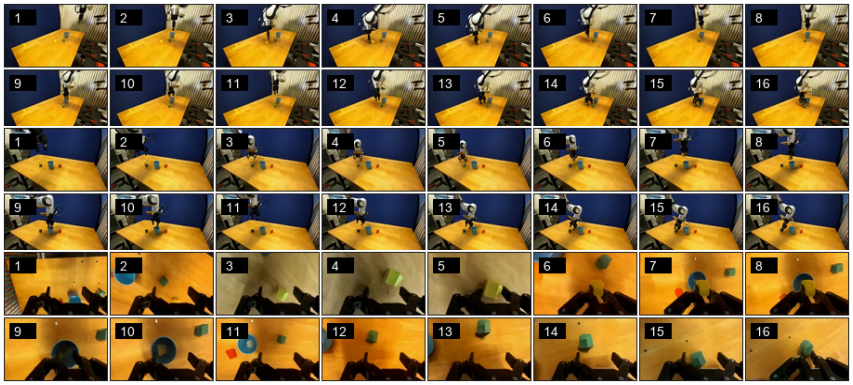




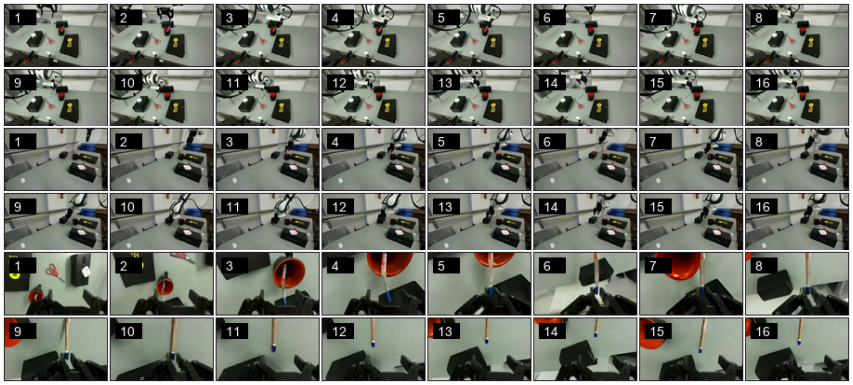
Shift the green rectangular block to the right.



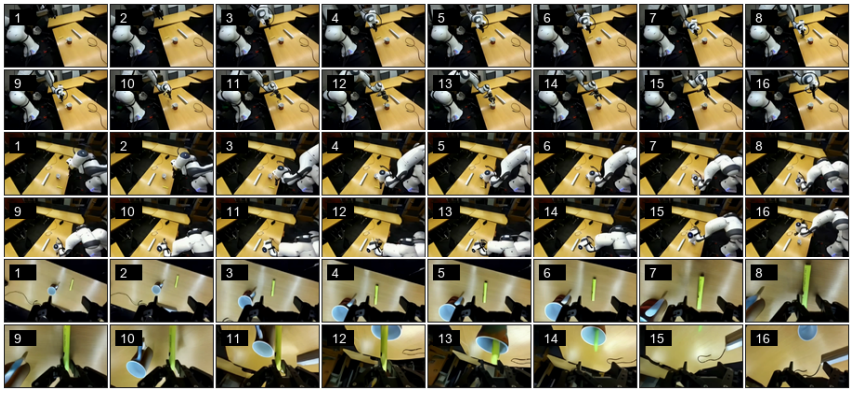
Lift the yellow block from the table and put it in the light blue cup.



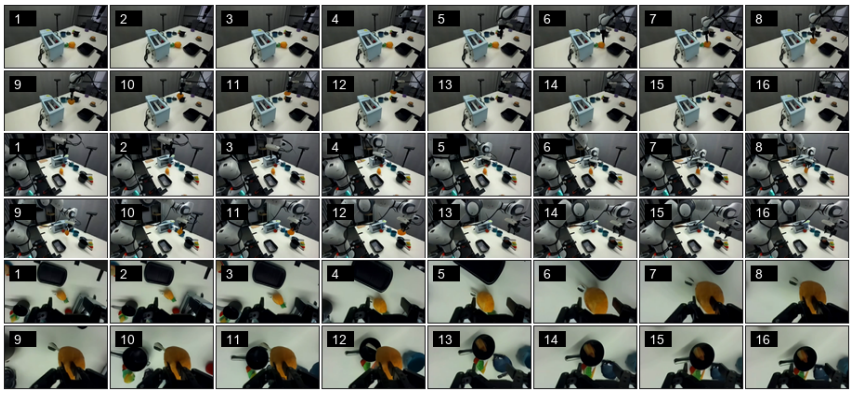
Take the marker out of the mug.



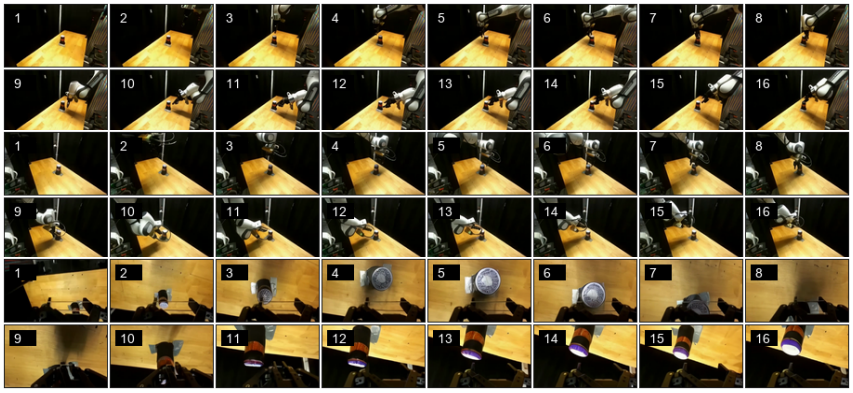
Place the yellow marker inside the red mug.



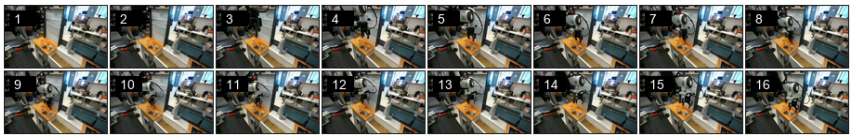
Place the pineapple plush toy in the black pot.

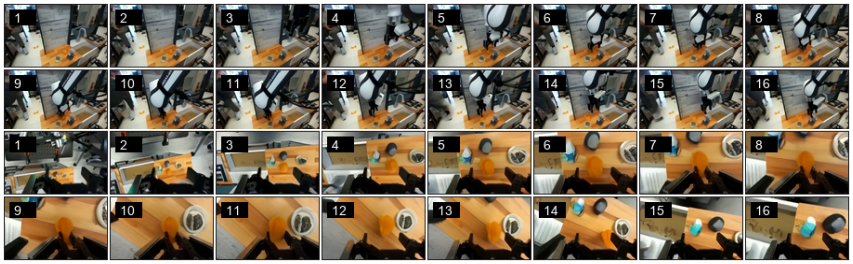


Place the elastic around the bottle.

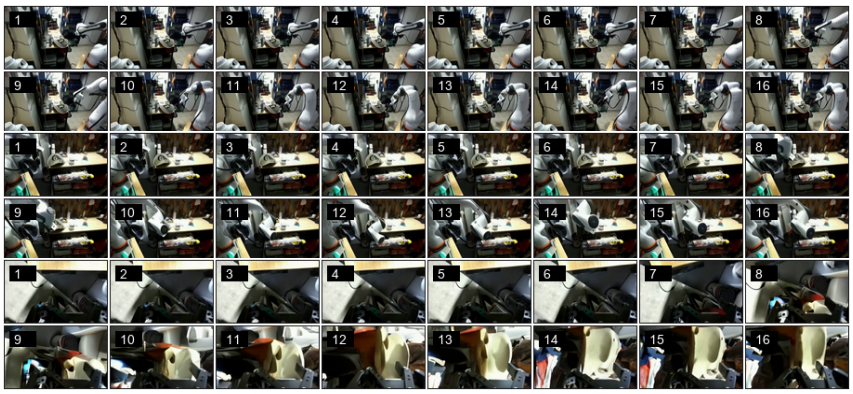


Shift the orange ball slightly to the left.





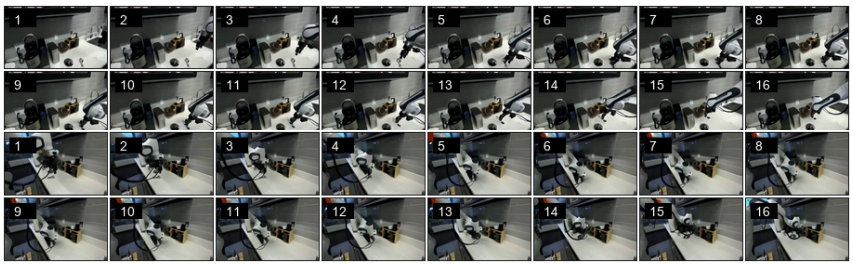
Flip the hat on the table.

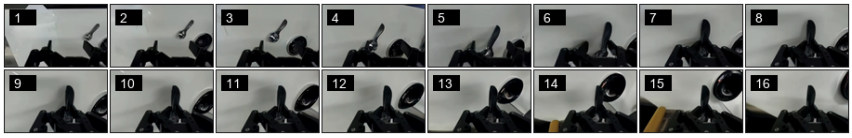


Set the cup upright.



Use the spoon to put some contents of the jar in the cup.





Stir the contents in the pot with the spatula.

