

Act2Goal: From World Model To General Goal-conditioned Policy

Pengfei Zhou^{1,*}, Liliang Chen^{1,*}, Shengcong Chen¹, Di Chen¹,
Wenzhi Zhao¹, Rongjun Jin¹, Guanghui Ren¹, Jianlan Luo^{1,†}
¹Agibot Research

Abstract—Specifying robotic manipulation tasks in a manner that is both expressive and precise remains a central challenge. While visual goals provide a compact and unambiguous task specification, existing goal-conditioned policies often struggle with long-horizon manipulation due to their reliance on single-step action prediction without explicit modeling of task progress. We propose Act2Goal, a general goal-conditioned manipulation policy that integrates a goal-conditioned visual world model with multi-scale temporal control. Given a current observation and a target visual goal, the world model generates a plausible sequence of intermediate visual states that captures long-horizon structure. To translate this visual plan into robust execution, we introduce Multi-Scale Temporal Hashing (MSTH), which decomposes the imagined trajectory into dense proximal frames for fine-grained closed-loop control and sparse distal frames that anchor global task consistency. The policy couples these representations with motor control through end-to-end cross-attention, enabling coherent long-horizon behavior while remaining reactive to local disturbances. Act2Goal achieves strong zero-shot generalization to novel objects, spatial layouts, and environments. We further enable reward-free online adaptation through hindsight goal relabeling with LoRA-based finetuning, allowing rapid autonomous improvement without external supervision. Real-robot experiments demonstrate that Act2Goal improves success rates from 30% to 90% on challenging out-of-distribution tasks within minutes of autonomous interaction, validating that goal-conditioned world models with multi-scale temporal control provide structured guidance necessary for robust long-horizon manipulation. Project page: <https://act2goal.github.io/>

I. INTRODUCTION

Learning robotic manipulation policies requires task specifications that are both expressive and precise. While natural language offers a flexible interface for diverse tasks, it often lacks the necessary granularity for fine-grained manipulation, a limitation that is more pronounced in complex, multi-stage scenarios. Visual goals offer a more precise alternative by directly encoding object configurations, spatial relations, and terminal constraints, avoiding linguistic ambiguity and explicit reward engineering.

Goal-conditioned policies (GCPs) map the current observation and a target visual goal directly to actions [1, 2, 3, 4]. While these methods perform well in short-horizon settings and exhibit reasonable generalization [5, 6], their performance degrades in long-horizon tasks. This limitation arises because standard GCPs operate via direct action prediction [7], lacking

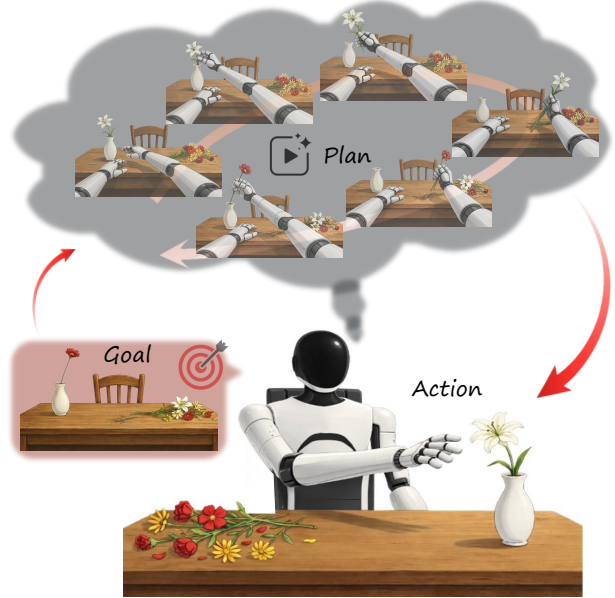


Fig. 1: **Method Overview.** The model receives a visual goal (left), imagines how to achieve it via a goal-conditioned world model (top), and executes the planned actions in the real world (right).

an explicit representation of task progress, intermediate feasibility, or long-horizon consistency. This issue is further exacerbated when GCPs are trained on narrowly scoped demonstration data. Without an explicit model of visual transitions toward the goal, such policies tend to overfit demonstrated state–action mappings and must rely on dense supervision to compensate for the lack of structured intermediate guidance. This limitation becomes particularly pronounced in long-horizon or out-of-distribution settings, where maintaining coherent progress toward the goal requires reasoning beyond locally observed transitions.

To generalize beyond demonstrations, GCPs must therefore explicitly model the visual dynamics required to reach a desired goal. Without such mechanism, the policy cannot distinguish between actions that make meaningful progress toward the goal and those that merely match local state–action correlations observed in demonstrations.

Recent advances in world models provide a promising pathway to address these limitations [8, 9, 10]. Vision–language

*Equal contribution.

†Corresponding author.

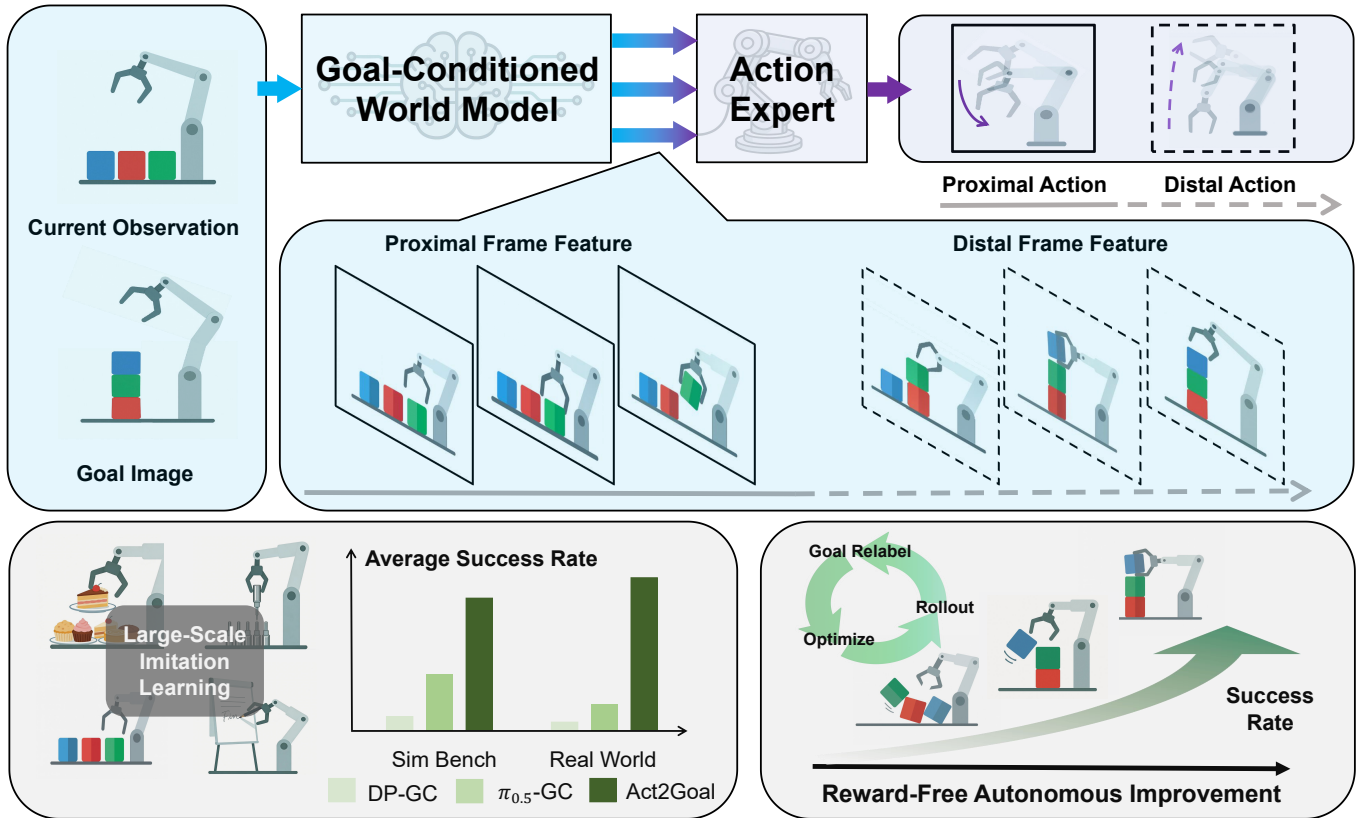


Fig. 2: **System Overview.** We propose Act2Goal, a goal-conditioned policy that integrates a visual world model with multi-scale temporal control to address long-horizon manipulation. After large-scale offline imitation learning, the model shows high performance on seen settings and strong generalization to unseen scenarios. The reward-free online autonomous improvement stage further improve model’s performance through rollout-goal relabel-optimize loop.

world models demonstrate that generative prediction of future visual states—conditioned on task instructions—can effectively support planning and decision-making [11, 12, 13]. Building on this insight, we consider a goal-conditioned analogue: goal-conditioned world model. Instead of predicting the future from high-level language, a goal-conditioned world model produces a plausible sequence of intermediate states bridging the gap between the current observation and a desired visual goal. This capability remedies a fundamental limitation of conventional GCPs by introducing an explicit representation of scene evolution over time, enabling a visually grounded and task-consistent trajectory toward the specified goal.

However, predicting a plausible visual path is only part of the solution. Executing long-horizon manipulation tasks reliably requires addressing a deeper control challenge. A general GCP must balance global consistency—maintaining fidelity to the long-term goal—with local reactivity—responding robustly to perturbations and correcting errors in closed-loop execution. Full-trajectory planning provides global coherence but is brittle under deviations; short-horizon control improves robustness but easily loses directional alignment in extended tasks. This inherent tension poses a fundamental barrier to deploying GCPs as general-purpose robotic controllers.

To bridge this gap, we introduce Act2Goal, a general goal-conditioned policy that integrates a goal-conditioned world model and a novel temporal decomposition mechanism termed Multi-Scale Temporal Hashing (MSTH). MSTH decomposes the generated visual trajectory into dense proximal frames for fine-grained control and sparse, horizon-adaptive distal frames that anchor the global plan. This multi-scale structure enables the policy to reason over long-horizon objectives while reacting quickly to local disturbances during closed-loop execution. We further couple the goal-conditioned world model with an action expert via layer-wise cross-attention, allowing intermediate visual representations to guide low-level motor control in an end-to-end differentiable architecture.

Initialized from large-scale human demonstration data, Act2Goal demonstrates strong zero-shot generalization across unseen objects, rearrangements, and environments. Beyond offline training, Act2Goal supports reward-free online autonomous improvement via Hindsight Experience Replay (HER) [14]. By relabeling its own rollouts as additional goal-achieving trajectories and updating the policy efficiently through LoRA-based finetuning [15], the system rapidly adapts to new real-world scenarios without external supervision. In real-robot experiments, this enables Act2Goal to substantially

improve performance on challenging long-horizon, out-of-distribution tasks, increasing success rates from 0.30 to 0.90 within minutes of autonomous interaction. These results validate the central premise of this work: that goal-conditioned world models, combined with multi-scale temporal reasoning, provide the structured intermediate guidance necessary for robust generalization and closed-loop execution in long-horizon manipulation.

Our main contributions are threefold. First, we present a novel end-to-end goal-conditioned policy that integrates a visual world model with motor control, enabling robust zero-shot generalization across unseen objects, environments, and goals. Second, we introduce Multi-Scale Temporal Hashing (MSTH), a temporal representation that decomposes trajectories into proximal and distal frames to balance long-horizon planning with closed-loop local control. Third, we develop a reward-free online adaptation mechanism based on HER-style goal relabeling with LoRA-based finetuning, enabling rapid autonomous improvement on out-of-distribution tasks.

II. RELATED WORKS

Our Act2Goal policy combines a goal-conditioned world model with an action expert capable of online autonomous improvement. Accordingly, we review related work in goal-conditioned policies, world models for robotic control, and online autonomous improvement.

A. Goal-conditioned Policy

Goal-conditioned policy learning aims to train agents to reach diverse formats of goals, e.g., visual goals [2, 3, 4, 16, 17, 18, 19], tracking points [20, 21] and motion field [22]. Early works enhance imitation learning by relabeling goals [3, 14], while others extend reinforcement learning with structured goal representations [4, 23]. Recent advances further explore long-horizon reasoning [24], keyframe-based planning [24], and goal generation via program synthesis [23, 25].

Among them, GoalGAIL [3] incorporates HER to learn from suboptimal or state-only demonstrations, improving sample efficiency in imitation settings. CoA [24] proposes generating action sequences in reverse from goal keyframes to maintain long-horizon consistency in manipulation.

These methods, however, typically rely on explicit goal supervision or struggle to align current observations with distant goals. Act2Goal addresses these limitations by introducing a goal-conditioned world model to simulate structured visual trajectories and proposing Multi-Scale Temporal Hashing (MSTH) to support consistent and efficient long-horizon planning, enabling better generalization in unseen tasks.

B. World Model for Robotic Control

World models have become a powerful tool in robotic control, enabling agents to simulate environmental dynamics [26, 27, 28], generate synthetic data for training [29, 30], or serve as learned simulators to guide policy learning [31, 32]. Recent works further combine world models with action experts (AEs) to form policy planning systems [33, 34, 12, 11],

where the world model provides future state feature and the AE predicts actions accordingly.

Among these, GE-Act [13] adopts a bi-model architecture with a world model predicting future visual feature based on language instruction and a transformer-based planner generating actions. WorldVLA [35] jointly predicts vision and action in a unified latent space, aiming for tighter vision-action alignment.

Different from prior works, Act2Goal leverages a purely vision-based goal-conditioned world model to guide policy learning with structured visual trajectories. To the best of our knowledge, this is the first work to integrate a world model into goal-conditioned policy learning.

C. Online Autonomous Improvement

To enhance policy adaptability during deployment, recent studies explore online improvement via either interactive imitation learning such as DAgger [36, 37, 38] or in-context learning (ICL) [39, 40]. However, DAgger-style methods require frequent expert intervention, while ICL methods do not update model weights and often show limited performance on complex tasks.

An alternative line of work leverages Hindsight Experience Replay (HER) [14] and its extensions [41, 42, 43], which relabel transitions by replacing original goals with achieved states, then optimize the policy using reinforcement or imitation learning. While these methods reduce the need for explicit rewards, they still rely on complex reward relabeling or external reward signals.

Act2Goal takes a further step toward fully autonomous online improvement. By combining HER-style relabeling with efficient LoRA-based finetuning, it enables direct policy adaptation from self-collected rollouts, without any task rewards or human annotations. This yields a lightweight, fully self-supervised update mechanism suitable for real-world deployment.

III. FROM WORLD MODEL TO GENERAL GOAL-CONDITIONED POLICY

Our framework is designed to address two key challenges in long-horizon goal-conditioned manipulation: aligning the action policy with high-level goal semantics, and maintaining planning efficiency over extended time scales. As shown in Figure 2, we tackle the first challenge by introducing a **Goal-Conditioned World Model (GCWM)** to guide the policy with imagined visual futures, providing rich, temporally coherent representations. To address the second challenge, we propose **Multi-Scale Temporal Hashing (MSTH)**, which enables the policy to focus on both short-term execution and long-term goal awareness through structured temporal abstraction.

Our learning process consists of three stages. Stage 1 performs joint training of the GCWM and action expert to align their representations. Stage 2 focuses on action adaptation, further improving the policy’s performance. Stage 3 introduces optional autonomous improvement, allowing the model to self-adapt in novel scenarios during deployment. We describe each component in detail below.

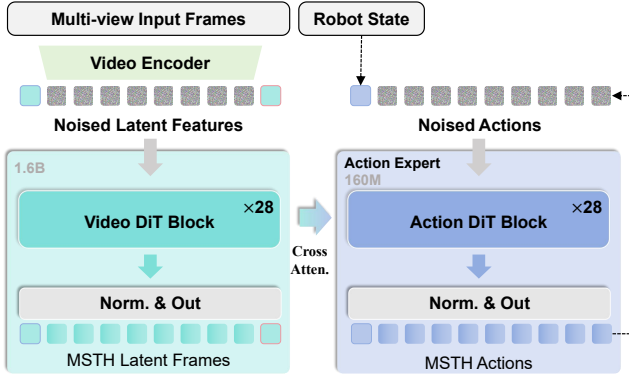


Fig. 3: **Model Architecture.** This figure presents the network architecture of Act2Goal model. On the left, multi-view input frames, including current observation and goal, are encoded into latents via a video encoder and concatenated with noisy latents, then refined into MSTH latent frames through Video DiT blocks. On the right, the robot state and multi-scale features from the world model are fed via cross-attention into isomorphic Action DiT blocks, generating MSTH-structured actions.

A. Goal-Conditioned World Model-Guided Policy

As illustrated in Fig. 3, our goal-conditioned world model builds upon the Genie Envisioner architecture with key modifications tailored for goal-conditioned policy learning [13]. We introduce a goal visual condition that is concatenated with the current observation along the hidden states sequence, while removing all language-conditioning components to create a purely vision-based model.

Our goal-conditioned world model employs a continuous flow matching approach for generative modeling. The process can be abstracted as learning a transformation from random noise to structured visual sequences conditioned on both current observations and goal states:

$$z_{pred} = f_{\theta}(z_t, z_g, \epsilon), \quad (1)$$

where z_t and z_g are VAE-compressed latents of the current observation and goal respectively, ϵ represents random noise inputs, which have the same shape as z_{pred} , and f_{θ} is the flow matching model that generates the latent frames between current observation and goal.

During inference, the model progressively refines the noisy latents through a deterministic flow process [?]:

$$z^{(n+1)} = z^{(n)} + \frac{1}{N} v_{\theta}(z^{(n)}, z_t, z_g), \quad (2)$$

where v_{θ} is the learnt vector field that guides the denoising process over multiple steps $n = 0, 1, \dots, N - 1$. The completed latent frames can be decoded into visual states using the VAE decoder.

Following the GE-Act implementation, our action expert employs a network architecture that is isomorphic to the world model, featuring the same number of DiT blocks but with reduced network width [13]. The action trajectory is

predicted using a flow matching process conditioned on both the proprioceptive state c_p and the multi-scale features c_w from the world model.

The action prediction process can then be formulated as:

$$a_{pred} = g_{\phi}(c_w, c_p, \zeta), \quad (3)$$

where g_{ϕ} is the action flow matching model, ζ are noise inputs for action generation, $c_w = \{h_{world}^1, \dots, h_{world}^L\}$ represents the layered transition features, and c_p is the proprio state condition. The flow matching process during inference for actions follows an iterative refinement:

$$a^{(n+1)} = a^{(n)} + \frac{1}{N} u_{\phi}(a^{(n)}, c_w, c_p), \quad (4)$$

where u_{ϕ} is the learned vector field for action generation.

B. Multi-Scale Temporal Hashing for Visual State and Action

In our framework, the *Multi-Scale Temporal Hashing* (MSTH) mechanism jointly guides the goal-conditioned world model and the action expert, enabling a consistent yet flexible multi-scale temporal abstraction. Given a total imagined trajectory length K , a proximal horizon P , and a vision sampling stride r , MSTH partitions the future trajectory into two segments.

The proximal segment consists of high-frequency short-horizon visual states $\{s_{t+kr}\}_{k=1}^{P/r}$, which capture fine-grained local dynamics. The distal segment contains M sparsely sampled visual states $\{s_{t+d_m}\}_{m=1}^M$, where the indices d_m are determined by logarithmic spacing:

$$d_m = P + \left\lfloor \frac{K - P}{\log(M + 1)} \cdot \log(m + 1) \right\rfloor, \quad m = 1, \dots, M. \quad (5)$$

This logarithmic sampling results in increasing temporal intervals as the horizon extends, providing coarse but goal-aligned long-term guidance.

The predicted action sequence follows the same multi-scale structure, with an important distinction from vision. Proximal actions are predicted at *every* timestep, $\{a_{t+1}, a_{t+2}, \dots, a_{t+P}\}$, enabling dense motor control even when visual states are subsampled by stride r . In contrast, distal actions $\{a_{t+d_m}\}_{m=1}^M$ are aligned with the distal visual states and serve as long-horizon guidance. During deployment, only the proximal actions are executed, while distal predictions remain latent and guide long-term goal adherence.

C. Two-Stage Offline Training

To endow Act2Goal with strong generalization capabilities, we first train the model through large-scale offline imitation learning. The training process consists of two main stages, designed to ensure that the transition trajectory prediction objective aligns closely with the ultimate goal of guiding action planning.

In the first stage, we fine-tune a pre-trained world model to adapt it for the transition trajectory prediction task of generating multi-view video frames following the MSTH distribution between initial observation and goal condition. To enhance

Algorithm 1 Act2Goal Online Autonomous Improvement

- 1) Initialize replay buffer \mathcal{B} and LoRA parameters ϕ
- 2) **While** performance not converged:
 - a) Execute policy for one episode, storing (o, c_p, a, o') at each step
 - b) **For each** transition (o, c_p, a, o') in episode:
 - i) $\mathcal{B} \leftarrow \mathcal{B} \cup \{(o, c_p, a, o')\}$
 - c) **If** $|\mathcal{B}| \geq N$:
 - i) **For** $k = 1$ to K :
 - A) Sample batch $\{(o, c_p, a, o')\} \sim \mathcal{B}$
 - B) $g' \leftarrow o'$ // Relabel achieved visual state as goal
 - C) $\mathcal{L} = \mathbb{E}[\|\pi_\theta(o, c_p, g') - a\|^2]$
 - D) $\phi \leftarrow \phi - \alpha \nabla_\phi \mathcal{L}$
 - ii) $\mathcal{B} \leftarrow \emptyset$

the alignment between transition trajectory prediction and action planning objectives, we jointly train both the transition trajectory prediction task and the action generation task using flow matching.

The training objective for the visual generation component follows the flow matching loss formulation:

$$\mathcal{L}_v = \mathbb{E}_{t \sim U(0,1), z_0, z_1, z_t, z_g} [\|v_\theta(t, \phi_t(z), z_t, z_g) - (z_1 - z_0)\|^2] \quad (6)$$

$$\mathcal{L}_a = \mathbb{E}_{t \sim U(0,1), a_0, a_1, c_w, c_p} [\|u_\phi(t, \psi_t(a), c_w, c_p) - (a_1 - a_0)\|^2]. \quad (7)$$

The joint training objective combines both losses with a balancing coefficient λ . In our experiment, we set $\lambda = 0.1$:

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_v + \lambda \cdot \mathcal{L}_a. \quad (8)$$

This joint optimization ensures that the goal-conditioned world model learns to generate visual trajectories that are not only visually plausible but also actionable, creating a strong foundation for the subsequent policy learning stage.

In the second training stage, we employ behavioral cloning to fine-tune the entire model end-to-end using only the action flow matching loss $\mathcal{L}_{\text{stage2}} = \mathcal{L}_a$, further strengthening its action planning capabilities. This stage focuses on aligning the complete pipeline—from visual perception to action execution—with expert demonstrations. The gradients from the action loss propagate through both the action generation components and the goal-conditioned world model, allowing the visual representations to be optimized specifically for action planning.

This two-stage offline training approach enables Act2Goal to acquire robust world understanding and action generation capabilities that transfer effectively to unseen environments and tasks.

D. Online Autonomous Improvement

While the model exhibits strong generalization capabilities after offline imitation learning, achieving high performance on novel tasks, environments, objects, and motion control chains remains challenging when deployed on physical robots—a

common limitation of imitation learning-based policies. To address this, we introduce an online autonomous improvement framework based on Hindsight Experience Replay (HER), enabling autonomous performance enhancement during deployment [14].

As illustrated in Algorithm 1, the system operates as follows: for a given expected goal condition, the model infers and executes actions. Each inference step—consisting of the starting observation latent, initial proprio state, output action, and resulting observation latent after execution—is automatically collected in a replay buffer on the edge device. Crucially, regardless of whether the transition successfully achieves the intended goal, we automatically relabel the goal condition as the robot’s observation at the end of the inference step, eliminating the need for manual labeling of transitions. When the replay buffer reaches a predetermined capacity threshold, we perform a fixed number of end-to-end training iterations using the buffer transitions, following the same approach as Stage 2 of offline imitation learning. To ensure efficient on-device training, only the additionally introduced LoRA layers in the Act2Goal model are updated, while the base model parameters remain frozen [15]. After completing the fixed iteration training, the replay buffer is cleared, and the system continues collecting new data through rollouts, repeating this cycle until performance meets expectations. We set maximum inference counts for different tasks; exceeding this threshold triggers an automatic robot reset. During real-robot experiments, the only required human intervention is manually restoring or modifying the experimental setup when necessary.

IV. EXPERIMENTS

Our experiments aim to systematically evaluate the effectiveness of the proposed ACT2GOAL model across both offline and online settings. We assess the model’s ability to generalize to novel scenarios, and to autonomously improve itself through interactions during deployment. Specifically, we study three key aspects: (1) the generalization capability of policies trained offline using imitation learning, measured under both in-domain (ID) and out-of-domain (OOD) scenarios; (2) the model’s capacity for online autonomous improvement, enabled by lightweight on-device fine-tuning of LoRA layers with self-relabelled transitions; and (3) the contribution of our proposed *Multi-Scale Temporal Hashing* (MSTH) mechanism, evaluated through qualitative analysis and ablations on both ID and OOD tasks.

A. Generalization Capability Evaluation

To comprehensively evaluate the generalization capability and versatility of Act2Goal after offline imitation training, we conducted extensive experiments in both simulated benchmarks and real-world robotic tasks. We selected four distinct tasks (Move can pot, Pick dual bottles, Place empty cup, Place shoe) from the Robotwin 2.0 simulation benchmark [44]. For each test episode in the simulator, the final goal image was



Fig. 4: **Real World Evaluation.** This figure illustrates in-domain and out-of-domain test configurations for three real-world tasks: Whiteboard Word Writing, Dessert Plating, and Plug-In Operation. For each task, Head View Goal displays the target, while Model Rollouts shows the robot’s execution process; these setups are used to evaluate the model’s generalization ability using the success rate as the metric.

TABLE I: **Comparison of Performance on Robotwin 2.0 Simulation Benchmark.** Act2Goal outperforms baselines in all Easy-mode tasks and 3 Hard-mode tasks, showing superior generalization.

	Model/Task	Move Can	Pick Bottles	Place Cup	Place Shoe
Easy	DP-GC	0.18	0.04	0.03	0.04
	$\pi_{0.5}$ -GC	0.54	0.13	0.16	0.30
	HyperGoalNet	0.11	0.08	0.08	0.01
	Act2Goal	0.62	0.80	0.64	0.52
Hard	DP-GC	0.00	0.00	0.00	0.00
	$\pi_{0.5}$ -GC	0.42	0.06	0.04	0.06
	HyperGoalNet	0.00	0.00	0.00	0.00
	Act2Goal	0.13	0.43	0.13	0.15

obtained via fixing the environment seed to ensure the reproducibility and consistency of the goal configuration across all test episodes, extracting it from successfully executed trajectories under this seed, and used as the goal condition for our model. The Robotwin 2.0 benchmark is particularly suitable for assessing generalization as it includes both seen scenarios (Easy mode) and highly challenging unseen scenarios (Hard mode).

We compare Act2Goal with several state-of-the-art policy architectures, including: (1) DP-GC, which conveys the SigLIP features of both the current observation and the goal image to a DiT-style Action Expert via cross-attention; (2) HyperGoalNet, a recent high-performing open-source goal-conditioned policy; and (3) the $\pi_{0.5}$ -GC baseline, which uses a fixed language condition for all data while incorporating both a goal image and the current raw observation as visual inputs to $\pi_{0.5}$ [3, 19, 45, 46].

The results in Table I demonstrate that Act2Goal significantly outperforms all these baselines. Notably, it exhibits an overwhelming advantage in the Hard mode, particularly in unseen scenarios, highlighting its superior generalization capability.

To further validate the generalization capability of Act2Goal

TABLE II: **Comparison of Performance on Real-World Manipulation Tasks.** Act2Goal demonstrates remarkable performance across all real-world tasks, significantly outperforming all baseline methods. The robust performance in OOD settings strongly validates its superior generalization ability. All results are obtained without any online autonomous improvement, using only offline imitation learning.

	Model/Task	Whiteboard Word Writing	Dessert Plating	Plug-In Operation
ID	DP-GC	0.00	0.10	0.00
	$\pi_{0.5}$ -GC	0.23	0.18	0.00
	HyperGoalNet	0.00	0.08	0.00
	Act2Goal	0.93	0.75	0.45
OOD	DP-GC	0.00	0.00	0.00
	$\pi_{0.5}$ -GC	0.20	0.05	0.00
	HyperGoalNet	0.00	0.00	0.00
	Act2Goal	0.90	0.48	0.30

in realistic scenarios, we conducted real-world experiments on an AgiBot Genie-01 robot using three challenging manipulation tasks. These tasks are characterized by objectives that are difficult to precisely specify via language, making them particularly suitable for goal-conditioned policies, and they require fine-grained control for successful execution. The data for these test tasks are part of the full dataset used for offline imitation training, and we did not perform any task-specific supervised fine-tuning (SFT). For each task, we designed both in-domain (ID) and out-of-domain (OOD) test configurations to thoroughly assess the model’s generalization ability, using success rate as the evaluation metric. The three real-world tasks are designed as follows:

- **Task 1: Whiteboard Word Writing.** The robot writes English words on a whiteboard. The ID test involves writing words seen in the training data (which includes 200 words), whereas the OOD test requires writing unseen words with novel character combinations, testing the model’s compositional generalization.
- **Task 2: Dessert Plating.** The robot arranges desserts

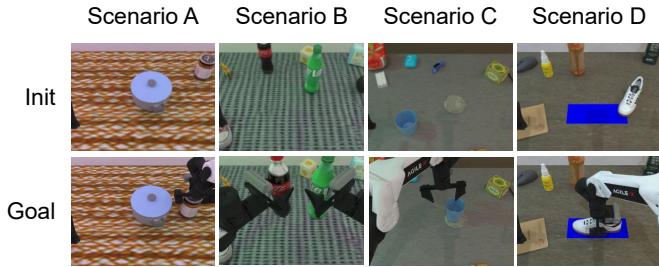


Fig. 5: **Online Autonomous Improvement Scenarios.** This figure illustrates four OOD scenarios from the RoboTwin 2.0 benchmark, corresponding to the hard testing modes of Move Can Pot, Pick Dual Bottles, Place Empty Cup, and Place Shoe. These scenarios serve as the testbed for verifying the effectiveness of autonomous improvement.

on a plate according to a visual example. The ID test uses dessert types, plate styles, and backgrounds seen during offline imitation training, while the OOD test introduces significant visual variations. This task evaluates the policy’s goal-following capability under strong visual distractions.

- **Task 3: Plug-In Operation.** The robot performs insertion tasks. The ID test involves inserting a metal workpiece into a corresponding hole, as seen in training. The OOD test requires inserting a cylindrical drink bottle into a cup holder—an unseen task that assesses the model’s skill transfer capability.

As summarized in Fig. 4 and Table II, Act2Goal demonstrates remarkable performance across all real-world tasks, significantly outperforming all baseline methods. Its robust performance in OOD settings strongly validates the superior generalization ability of the proposed approach.

B. Analysis of Online Autonomous Improvement

We validated the effectiveness of online autonomous improvement in both simulated and real-world environments, and systematically analyzed best practices for enhancing robot manipulation performance using Hindsight Experience Replay (HER).

In the Robotwin 2.0 simulation environment, we selected one hard-mode scenario from each of the four tasks described in Section 4.2 (shown in Fig. 5) for multi-round online training, evaluating the success rate after each round. As shown in the left part of Fig. 6, the model’s performance improved consistently over approximately three rounds of training before converging, with a maximum success rate improvement of up to 8× compared to the pre-trained baseline. We also compared three data selection strategies for rollout usage: (1) using only successful rollouts, (2) using all rollouts regardless of success, and (3) using only failed rollouts. We found that utilizing all rollouts yielded the best performance. Notably, even using only failed rollouts still led to clear improvement, demonstrating that the HER-based strategy—extracting useful experience from failures—is effective for goal-conditioned

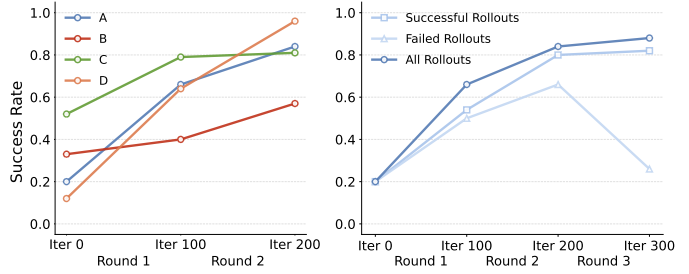


Fig. 6: **Online Training Performance in Robotwin 2.0.** This figure presents two key findings from Robotwin 2.0 simulations: (Left) Multi-round success rates of four hard-mode scenarios, showing consistent improvement over 3 rounds before convergence. (Right) Performance of three data selection strategies for rollouts: using all rollouts yields optimal results, while even failed-only rollouts enable noticeable improvement.

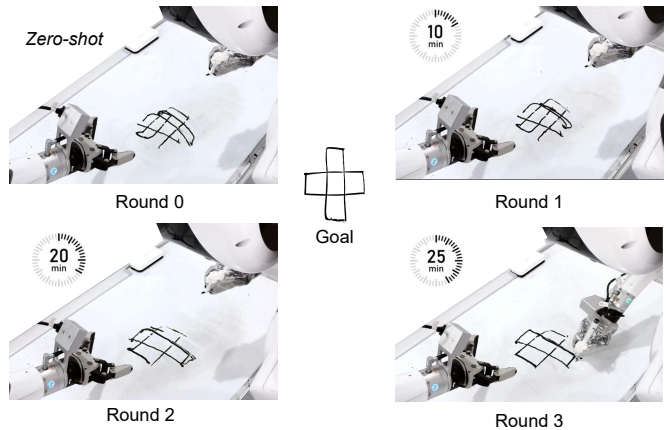


Fig. 7: **Online Training Performance in Real-World Unseen Scenarios.** When tasked with drawing an unseen pattern, the model initially performs poorly. However, as online training progresses (from left to right, top to bottom), the drawing quality improves steadily.

robot manipulation. The corresponding results can be found in the right part of Fig. 6.

In real-world experiments, we tasked the robot with whiteboard writing (Task 1 in Sec. 4.3). While the model exhibits strong generalization in whiteboard word writing, we assessed its capability to draw novel patterns that are substantially different from writing. As shown in Fig. 7, the model’s initial performance on this unfamiliar case was limited. Nevertheless, through online fine-tuning, it demonstrated consistent improvements in drawing quality in 15 minutes. On the OOD Plug-In task, the model also demonstrates a consistent increase in success rate throughout online training (from 0.30 to 0.90). Detailed information can be found in our project page.

C. Effectiveness of MSTH

Fig. 8 presents three representative examples of video clips generated by our goal-conditioned world model. As illustrated, the model is capable of producing both fine-grained proximal frames, which offer detailed short-term guidance essential for

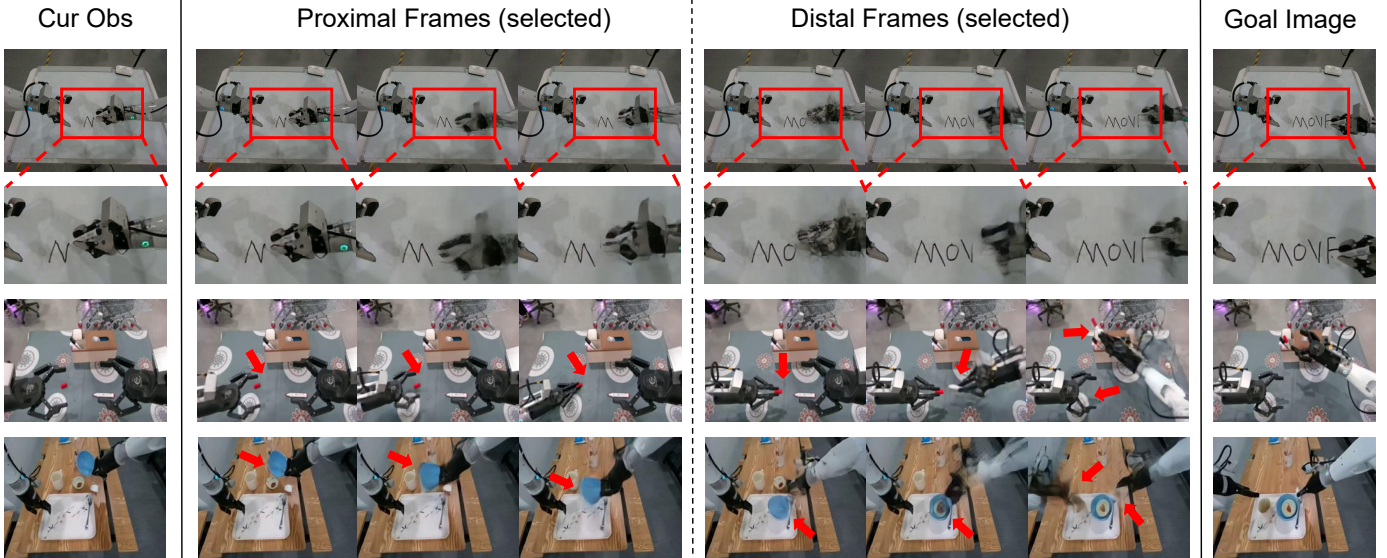


Fig. 8: **Examples of generated videos.** This figure illustrates the generation capability of our goal-conditioned world model through three head-view video clips. The left and right sides show the current observation and the target goal, respectively. For each sequence, we uniformly sample three proximal and three distal frames. The second row provides a zoomed-in view of the red-highlighted regions in the first row. Red arrows in the third and fourth rows highlight the motion of key objects. The distal frames in the bottom row correspond to longer temporal horizons compared to those in the third row.

TABLE III: **Effectiveness of MSTH.** Values denote the success rate of the real-world whiteboard word writing task. The word length definitions: Short (3 letters or fewer), Medium (4 to 6 letters), Long (7 letters or more).

	Model	Short	Medium	Long
ID	w/o MSTH	0.95	0.35	0.10
	w/ MSTH	0.95	0.90	0.90
OOD	w/o MSTH	0.60	0.20	0.00
	w/ MSTH	0.93	0.90	0.88

the action expert to generate accurate and executable actions, and sparser distal frames, which capture the long-horizon structure of the task and effectively guide the model toward the intended long-term goal. This hierarchical visual forecasting enables the model to reason across multiple temporal scales, facilitating both precise control and goal-directed planning.

We also conducted an ablation study to investigate the effectiveness of the proposed Multi-Scale Temporal Hashing (MSTH) mechanism in a real-world whiteboard word writing task, which requires long-horizon, visually grounded planning. As shown in Table III, we compare the success rates across words of varying lengths—short (no more than 3 letters), medium (4–6 letters), and long (no fewer than 7 letters)—under both in-domain (ID) and out-of-domain (OOD) settings.

The MSTH-based policy demonstrates a substantial improvement over the baseline strategy that relies on fixed-horizon action chunking. While both methods perform comparably on short-horizon tasks (e.g., short words), the fixed-

horizon baseline exhibits a drastic performance drop as the planning horizon increases, particularly in long and OOD scenarios. This degradation is primarily due to compounding goal misalignment over extended sequences, where the model struggles to maintain consistent intent.

In contrast, MSTH maintains high success rates across all word lengths and generalizes well to OOD words, indicating its robustness and effectiveness in handling long-horizon dependencies. By dynamically adjusting the temporal abstraction level and enabling better temporal anchoring to final goal, MSTH provides a simple yet principled mechanism for improving long-horizon goal-conditioned manipulation. These results highlight MSTH as a critical design component for scaling visual policy models to complex, temporally extended tasks.

V. CONCLUSION

We propose Act2Goal, a goal-conditioned policy that combines a visual world model with multi-scale temporal control to tackle long-horizon manipulation. The model generates both fine-grained short-term executable control signals and coarse long-term trajectories, enabling effective planning and control. Trained fully offline, Act2Goal achieves high success rates in seen tasks and shows strong generalization to unseen scenarios. To enhance adaptability, we incorporate Hindsight Experience Replay (HER) and LoRA-based finetuning, enabling efficient, reward-free online autonomous improvement within minutes. Together, these components contribute to a scalable and generalizable visuomotor policy for open-ended, real-world manipulation tasks.

REFERENCES

- [1] Leslie Pack Kaelbling. Learning to achieve goals. In *IJCAI*, volume 2, pages 1094–8, 1993.
- [2] Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.
- [3] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32, 2019.
- [4] Xudong Gong, Dawei Feng, Kele Xu, Bo Ding, and Huaimin Wang. Goal-conditioned on-policy reinforcement learning. *Advances in neural information processing systems*, 37:45975–46001, 2024.
- [5] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.
- [6] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *arXiv preprint arXiv:2412.15109*, 2024.
- [7] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023.
- [8] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.
- [9] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.
- [10] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Bajaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.
- [11] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.
- [12] Youpeng Wen, Junfan Lin, Yi Zhu, Jianhua Han, Hang Xu, Shen Zhao, and Xiaodan Liang. Vidman: Exploiting implicit dynamics from video diffusion model for effective robot manipulation. *Advances in Neural Information Processing Systems*, 37:41051–41075, 2024.
- [13] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.
- [14] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.
- [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] Abhinav Jain and Vaibhav Unhelkar. Go-dice: Goal-conditioned option-aware offline imitation learning via stationary distribution correction estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 12763–12772, 2024.
- [17] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Goal-conditioned dual-action imitation learning for dexterous dual-arm robot manipulation. *IEEE Transactions on Robotics*, 40:2287–2305, 2024.
- [18] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.
- [19] Pei Zhou, Wanting Yao, Qian Luo, Xunzhe Zhou, and Yanchao Yang. Hyper-goalnet: Goal-conditioned manipulation policy learning with hypernetworks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- [20] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.
- [21] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, pages 306–324. Springer, 2024.
- [22] Zhao-Heng Yin, Sherry Yang, and Pieter Abbeel. Object-centric 3d motion field for robot learning from human videos. *arXiv preprint arXiv:2506.04227*, 2025.
- [23] Guy Davidson, Graham Todd, Julian Togelius, Todd M Gureckis, and Brenden M Lake. Goals as reward-producing programs. *Nature Machine Intelligence*, 7(2): 205–220, 2025.
- [24] Wenbo Zhang, Tianrun Hu, Yanyuan Qiao, Hanbo Zhang, Yuchu Qin, Yang Li, Jiajun Liu, Tao Kong, Lingqiao Liu, and Xiao Ma. Chain-of-action: Trajectory autoregressive modeling for robotic manipulation. *arXiv preprint arXiv:2506.09990*, 2025.
- [25] Lance Ying, Katherine M Collins, Prafull Sharma, Cedric Colas, Kaiya Ivy Zhao, Adrian Weller, Zenna Tavares, Phillip Isola, Samuel J Gershman, Jacob D Andreas, et al. Assessing adaptive world models in machines with novel games. *arXiv preprint arXiv:2507.12821*, 2025.

- [26] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.
- [27] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.
- [28] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.
- [29] Haoyu Zhao, Cheng Zeng, Linghao Zhuang, Yaxi Zhao, Shengke Xue, Hao Wang, Xingyue Zhao, Zhongyu Li, Kehan Li, Siteng Huang, et al. High-fidelity simulated data generation for real-world zero-shot robotic manipulation learning with gaussian splatting. *arXiv preprint arXiv:2510.10637*, 2025.
- [30] Liu Liu, Xiaofeng Wang, Guosheng Zhao, Keyu Li, Wenkang Qin, Jiaxiong Qiu, Zheng Zhu, Guan Huang, and Zhizhong Su. Robottransfer: Geometry-consistent video diffusion for robotic visual policy transfer. *arXiv preprint arXiv:2505.23171*, 2025.
- [31] Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, et al. Enerverse-ac: Envisioning embodied environments with action condition. *arXiv preprint arXiv:2505.09723*, 2025.
- [32] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025.
- [33] Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Yue Liao, Peng Gao, Hongsheng Li, Maoqing Yao, et al. Enerverse: Envisioning embodied future space for robotics manipulation. *arXiv preprint arXiv:2501.01895*, 2025.
- [34] Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.
- [35] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.
- [36] 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.
- [37] Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.
- [38] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025.
- [39] Rutav Shah, Shuijing Liu, Qi Wang, Zhenyu Jiang, Sateesh Kumar, Mingyo Seo, Roberto Martín-Martín, and Yuke Zhu. Mimicdroid: In-context learning for humanoid robot manipulation from human play videos. *arXiv preprint arXiv:2509.09769*, 2025.
- [40] Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. Ricl: Adding in-context adaptability to pre-trained vision-language-action models. *arXiv preprint arXiv:2508.02062*, 2025.
- [41] Rui Yang, Meng Fang, Lei Han, Yali Du, Feng Luo, and Xiu Li. Mher: Model-based hindsight experience replay. *arXiv preprint arXiv:2107.00306*, 2021.
- [42] Liam Schramm, Yunfu Deng, Edgar Granados, and Abdeslam Boularias. Usher: Unbiased sampling for hindsight experience replay. In *Conference on Robot Learning*, pages 2073–2082. PMLR, 2023.
- [43] Yongle Luo, Yuxin Wang, Kun Dong, Qiang Zhang, Erkang Cheng, Zhiyong Sun, and Bo Song. Relay hindsight experience replay: Self-guided continual reinforcement learning for sequential object manipulation tasks with sparse rewards. *Neurocomputing*, 557:126620, 2023.
- [44] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.
- [45] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL <https://arxiv.org/abs/2504.16054>.
- [46] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.
- [47] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

APPENDIX

A. Implementation Details

Model. The goal-conditioned world model predicts 4 latent frames (2 proximal, 2 distal), decoded by a 3D VAE into 9 proximal and 9 distal visual frames. The action expert outputs 54 proximal actions (execute 50) and 9 distal actions (for guidance only; not executed). At inference, only actions are generated.

Training. We train the model on the AgiBot World dataset and a small proprietary dataset [47]. Stage 1 fine-tunes a pre-trained 1.6B-parameter Genie Envisioner (7×24 hours, 16×A800). Stage 2 performs end-to-end behavioral cloning (48 hours, 16×A800).

Deployment and Online Learning. The policy is deployed on an AgiBot Genie-01 robot with an NVIDIA RTX 4090. Online autonomous improvement fine-tunes only the LoRA layers (rank 64) using a replay buffer of size 20. Each training round (10 epochs) takes 5 minutes including rollout, backpropagation and environment resetting. Inference latency is 200ms for 50 executable actions.

B. Real World Task Setup

Writing and Painting Tasks. For the tasks of writing and drawing on a whiteboard, the robot gripper was first manually positioned to grasp the marker before task execution. During extended writing trials, the smooth surface of the marker occasionally caused it to slip from the gripper. To ensure reliability and reduce resetting time during batch testing, we used tape to further secure the marker in place.

Bearing Insertion Task. In the demonstrated bearing insertion task, each bearing weighs over 2 kg, with a base diameter of approximately 1 cm. The bearing is inserted into a hole with a diameter of about 1.5 cm.

Dessert Plating Task. To ensure experimental repeatability, we used silicone-made toy desserts for the dessert plating task.

C. Testing Metric

We employ task execution success rate as the performance metric for evaluating the models. Specifically, the real-world success rates reported in Section IV are obtained by manually labeling success or failure over 40 model rollouts per experiment. The success rates in simulation are automatically computed from 90 rollouts. For models in Section IV.B that underwent autonomous improvement, we saved the model weights after each training round and evaluated them individually once the autonomous improvement process was completed.

ACKNOWLEDGMENTS

We gratefully acknowledge the technical support and insightful discussions provided by Yifei Wei, Xindong He, Jingbin Cai, Yuxin Jiang, Siyuan Huang, Qi Qin, Yue Liao, Qi Lv, Sukai Wang, Ruofei Niu, Xiongfeng Cai, Hanwen Shi and Xuan Hu. We also thank Haoyu Cao, Cheng Jing, Rui Wu, Dan Liu and Jia Zeng for their diligent work in data preparation.