ROLLBACK: EXPERIENCE BACKTRACKING IN WORLD MODELS FOR POLICY IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned world models offer a scalable alternative to costly real-world interaction for improving robot policies. However, existing world-model-based methods largely treat world models as forward-only surrogates of the physical environment, *leaving untapped a capability no physical environment affords: rewinding to any past state to explore counterfactual futures*. In this work, we introduce **Rollback**, a policy improvement paradigm that turns world models from passive simulators into counterfactual engines: it rolls back to critical decision points, branches into alternative actions, and contrasts their outcomes to attribute success to individual decisions. This concentrates imagination where it matters most and distills counterfactual outcomes into fine-grained, action-level supervision without a separately learned value model. Furthermore, we build a scalable training system that unifies distributed world-model search with asynchronous real-robot execution, forming a self-reinforcing cycle: real-world experience continually sharpens the world model, and a sharper world model in turn drives further policy improvement. Experiments on three challenging real-world tasks show that Rollback improves success rates over the strongest baseline by 10.44%, 11.11%, and 11.85% on block extraction, ball tossing, and tabletop curling, respectively, while achieving more stable improvement throughout training.

1 INTRODUCTION

Action-conditioned world models have provided simulated environments for robot policy training, offering a scalable complement to real-world interaction (Hafner et al., 2018; 2019; 2020). By predicting the consequences of policy-generated actions, they allow policies to collect experience and learn from imagined outcomes without physical deployment (Yang et al., 2023; Zhu et al., 2024; Jiang et al., 2025). Recent video world models extend this paradigm to diverse robotic tasks, supporting policy evaluation (Li et al., 2025b; Gemini Robotics Team et al., 2025) and improvement through synthetic experience (Jang et al., 2025; Yang et al., 2026; Wang et al., 2026). These advances further extend to general-purpose robot policies, which can be refined using imagined interactions and task feedback (Xiao et al., 2025; Zhu et al., 2025). Moving exploration and optimization into learned environments makes policy learning less bound to physical trials.

Despite this progress, *existing world-model-based policy learning largely retains the trajectory-centric interaction protocol of real-world training*, using the world model primarily to generate forward rollouts whose intermediate contexts are discarded afterward. This overlooks a capability unavailable in physical RL: the ability to "rewind and reconsider." While a physical environment cannot generally return to an identical intermediate state after execution, a world model can restore a cached temporal context, rewind to an earlier decision point, and explore alternative action continuations from the same history. This enables the policy to reconsider what it could have done differently at a particular decision rather than restarting exploration from scratch. ***Our key insight is therefore to redesign world-model-based policy learning around a training-time experience backtracking primitive***, repeatedly rewinding to and reconsidering decisions for informative data generation. The world model becomes not merely a surrogate simulator, but a rewindable training environment.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

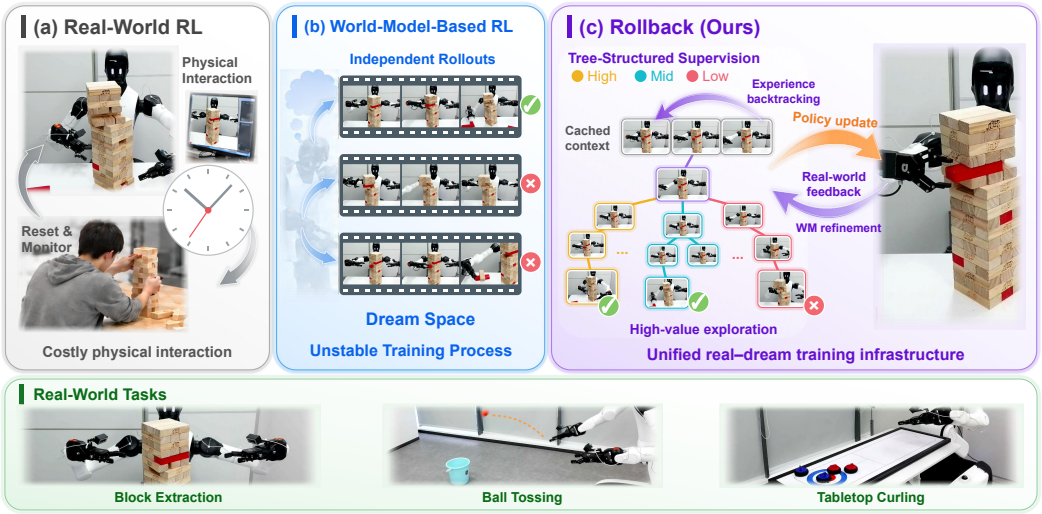


Figure 1: **From real-world interaction to learning in world models.** (a) Real-world RL incurs costly physical interaction. (b) Existing world-model-based RL spends much of its interaction budget on less informative exploration. (c) Our proposed Rollback enables high-value exploration and dense supervision through experience-backtracking tree search.

Fully exploiting this capability, however, requires addressing three key challenges as shown in Figure 1. *(i) Limited informative exploration.* Not all decisions are equally worth revisiting: along a trajectory, a few critical decisions determine the outcome, while alternatives to most others lead to similar results. Independent rollouts cannot identify these consequential decisions or adaptively allocate exploration among them (Ecoffet et al., 2020; Browne et al., 2012). *(ii) Sparse decision-level supervision.* Trajectory-level feedback does not reveal which intermediate actions determine the outcome (Andrychowicz et al., 2017; Tan et al., 2025). Treating continuations independently preserves this credit-assignment problem, whereas learning an additional value model introduces extra modeling assumptions and potential estimation bias (Lu et al., 2025; Schulman et al., 2015; Fujimoto et al., 2018). *(iii) Biased world-model predictions.* Branching from policy-selected contexts may shift interactions beyond the world model's training distribution, causing prediction errors to accumulate or be exploited during optimization (Janner et al., 2019; Yu et al., 2020; 2021; Chen et al., 2026). Without real-world grounding, imagined gains may not transfer to real robots.

To operationalize training-time experience backtracking and address these challenges, we present **Rollback**, a policy improvement paradigm that turns world models from passive simulators into rewindable engines. Rollback comprises three key components: *(i) Experience-backtracking tree search.* Starting from successful and failed real-world trajectories, Rollback rolls back to intermediate decision points and branches into alternative action chunks. Monte Carlo tree search adaptively allocates simulation budget toward decisions whose alternatives lead to divergent outcomes, avoiding exhaustive branching (Browne et al., 2012; Silver et al., 2016; Hubert et al., 2021). *(ii) Tree-structured value estimation.* Since sibling branches share a decision context, contrasting their downstream outcomes isolates the effect of individual actions. Subtree-level outcome aggregation yields fine-grained, action-level advantage supervision without an additional learned value model. *(iii) World model–policy co-evolution.* A scalable training system unifies distributed world-model search with asynchronous real-robot execution. Real-world experience continually refines the world model, which in turn provides more faithful imagination for search and policy improvement.

We evaluate Rollback on three challenging real-world robotics tasks: block extraction, ball tossing, and tabletop curling. Across these tasks, Rollback improves success rates over the strongest baseline by 10.44, 11.11, and 11.85 percentage points, respectively. Under matched interaction budgets, Rollback achieves faster and more consistent improvements in task success rates. Our anonymous project website is available at `https://anonymous.4open.science/w/Rollback/`.

In summary, our contributions in this work can be summarized as follows:

- We propose Rollback, a policy improvement paradigm that exploits experience backtracking to turn world models from passive simulators into counterfactual engines, supported by a training system unifying distributed world-model search with asynchronous real-robot execution.
- We design an experience-backtracking search that concentrates imagination on critical decision points, and a tree-structured value estimation that distills counterfactual outcomes into fine-grained, action-level supervision without a separately learned value model.
- Extensive experiments on three challenging real-world robotics tasks demonstrate that Rollback achieves higher success rates and more stable improvement.

2 Related Work

Robot Policy Learning from Interaction. Robot policies are commonly improved for downstream tasks and deployment conditions using task-specific data and interaction feedback (Kim et al., 2025; Guo et al., 2025b; Physical Intelligence et al., 2025a). For generalist policies including RT-2, OpenVLA, and the π series (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025b), supervised fine-tuning learns task-specific behaviors; OpenVLA-OFT improves adaptation and inference efficiency through optimized action representations and objectives (Kim et al., 2025). Reinforcement learning further exploits execution outcomes: iRe-VLA alternates reinforcement and supervised learning for stability (Guo et al., 2025b); VLA-RL learns a process reward model (Lu et al., 2025); RIPT-VLA uses dynamic sampling and leave-one-out advantages to learn from sparse success without a value network (Tan et al., 2025); and SimpleVLA-RL scales training through efficient sampling and parallelization (Li et al., 2025a). For real-world adaptation, RECAP combines demonstrations, autonomous experience, and expert corrections through advantage-conditioned training (Physical Intelligence et al., 2025a). *However, reinforcement learning on physical robots remains resource-intensive due to costly interactions and labor-intensive environment resets* (Yang et al., 2026; Zang et al., 2026).

World Models for Robotics. World models support robot learning and planning by predicting the consequences of candidate actions before physical execution (Hafner et al., 2018; 2019; Yang et al., 2023). Recent advances span general-purpose world simulation and models tailored to robot interaction. Cosmos-Predict2.5 and LingBot-World advance controllable video generation and interactive simulation (NVIDIA et al., 2025; Robbyant Team et al., 2026). UniSim, IRASim, EnerVerse-AC, and Ctrl-World model action-conditioned visual dynamics to support simulated interaction and policy development (Yang et al., 2023; Zhu et al., 2024; Jiang et al., 2025; Guo et al., 2025a). DreamDojo further transfers interaction knowledge from large-scale human videos to robot world modeling through latent actions (Gao et al., 2026). WorldEval, WorldGym, and the Veo robotics simulator study generated rollouts as proxies for real-world policy evaluation (Li et al., 2025b; Quevedo et al., 2025; Gemini Robotics Team et al., 2025). Beyond simulation, EnerVerse, Genie Envisioner, and LingBot-VA connect future visual prediction with action generation for robot control (Huang et al., 2025; Liao et al., 2025; Li et al., 2026). DreamGen uses generated videos and inferred actions to construct synthetic training trajectories (Jang et al., 2025), while V-JEPA 2-AC supports robotic planning through action-conditioned prediction in latent space (Assran et al., 2025).

World-Model-Based Policy Improvement. Recent work increasingly uses action-conditioned world models as interactive environments for policy improvement (Xiao et al., 2025; Zhu et al., 2025; Sharma et al., 2026; Yang et al., 2026; Jiang et al., 2026). World-Env combines a video simulator with a VLM-guided reflector that supplies task feedback and termination signals (Xiao et al., 2025), while WMPO optimizes policies through interactions with a learned world model (Zhu et al., 2025). World-Gymnast uses VLM-evaluated imagined rollouts for reinforcement learning and investigates iterative world-model updates using real-world experience (Sharma et al., 2026). To obtain informative supervision, RISE combines a controllable dynamics model with a learned progress value model that provides advantages for policy improvement (Yang et al., 2026). WoVR instead emphasizes the reliability of imagined interaction through keyframe-initialized rollouts and world model-policy co-evolution (Jiang et al., 2026). *However, these methods struggle to improve steadily, as forward-only rollouts rarely revisit informative decisions and learned values may be biased.*

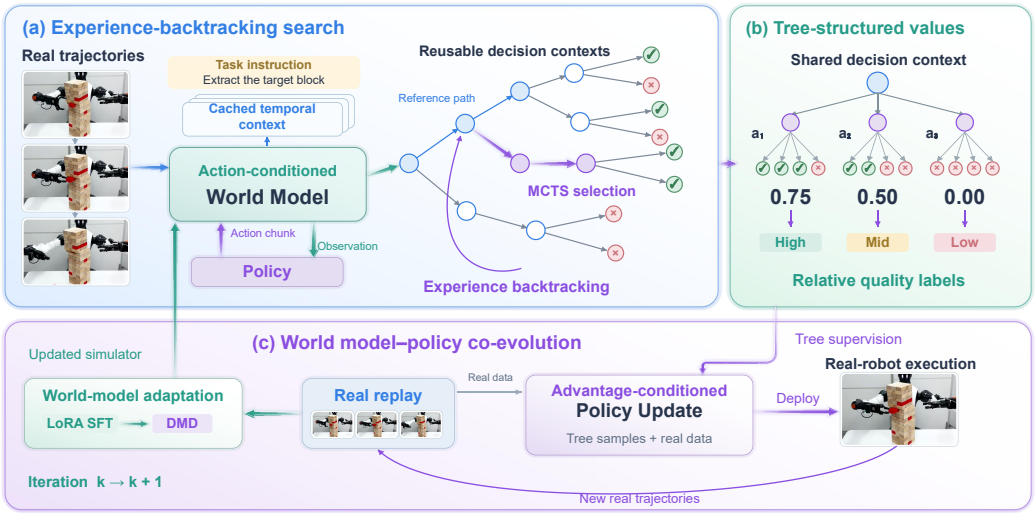


Figure 2: **Overview of Rollback.** (a) Experience-backtracking tree search restores cached world-model contexts to explore informative branches without regenerating shared prefixes. (b) Tree-structured value estimation aggregates downstream outcomes into relative action-quality labels for policy training without an additional value model. (c) World model-policy co-evolution incorporates real trajectories to refine the world model and improve the policy over iterations.

3 Proposed Rollback

In this section, we elaborate on our proposed Rollback, a policy improvement paradigm that exploits experience backtracking in world models. As illustrated in Figure 2, Rollback revisits decision points, compares alternative actions through their downstream outcomes, and distills the results into the policy, while real-world feedback refines the world model.

Problem Definition. Given a language instruction g and an observation o_t comprising multi-view images and proprioception, the policy π_θ generates an action chunk $\mathbf{a}_t \in \mathbb{R}^{H \times d_a}$ with horizon H and action dimension d_a. The action-conditioned world model $\mathcal{W}_\phi$ predicts the next observation and updates its temporal context through $(\hat{o}_{t+1}, m_{t+1}) \sim \mathcal{W}_\phi(\cdot \mid m_t, \mathbf{a}_t, g)$, where m_t denotes the retained temporal context. Our objective is to improve the policy's real-world task success by learning from imagined world-model interactions and real-world experience.

3.1 Experience-Backtracking Tree Search

Independent rollouts expend substantial computation on repeatedly generating shared trajectory prefixes, yet provide limited comparisons of alternative actions from critical intermediate decision contexts. World models enable these comparisons by restoring saved contexts and simulating alternative continuations from previously explored decision points. Building on this capability, Rollback organizes reusable decision contexts into a search tree, allowing exploration to branch from selected nodes without regenerating their shared prefixes.

Decision Context Construction. Experience-backtracking search requires decision contexts for exploring alternative continuations. Using GE-Sim 2.0 as the world model (Qiu et al., 2026), we construct a reference path by sampling an offline expert trajectory and retaining observations at action-chunk boundaries. Each non-terminal node serves as a branching point where the policy proposes alternative actions and the world model simulates their consequences from the corresponding real observation. We cache the world-model context at each imagined node, enabling further continuations from the same decision point without regenerating the shared prefix.

MCTS-Guided Exploration. Building on these reusable decision contexts and inspired by AlphaGo (Silver et al., 2016), we use Monte Carlo tree search (MCTS) (Browne et al., 2012) to concentrate a limited imagination budget on promising regions of the decision space. By propagating rollout outcomes back to earlier decisions, MCTS uses evidence from completed explorations to guide subsequent branch selection. Branches with higher observed success rates receive further exploration, while an exploration bonus preserves opportunities for less-visited alternatives. This adaptive allocation helps discover successful continuations and examine alternative actions around them, producing informative comparisons for policy improvement rather than spending the budget uniformly across decision points. For each node v, we maintain a visit count $N(v)$ and accumulated return $W(v)$, yielding the empirical search value $Q(v) = W(v)/N(v)$. An existing child u is selected according to the PUCT score as follows:

$$U(v, u) = Q(u) + c_{\text{puct}} P(u \mid v) \frac{\sqrt{1 + N(v)}}{1 + N(u)}, \tag{1}$$

where the first term exploits high empirical success rates and the second explores under-visited branches. We assign uniform priors $P(u \mid v)$ over existing children. For continuous action chunks, progressive widening (Couëtoux et al., 2011) permits at most $\lceil c_{\text{pw}} \max(1, N(v))^{\alpha} \rceil$ actions, where $0 < \alpha < 1$. When permitted, a zero-valued virtual child with prior $1/(|\mathcal{C}(v)| + 1)$ competes under the same rule, balancing new actions against existing continuations. Each new action is rolled out until success or the horizon, after which a task-specific reward model returns $y \in \{0, 1\}$. We back up this return via $N \leftarrow N + 1$ and $W \leftarrow W + y$. For informative comparisons, we sample two actions at each new branch and prioritize rollouts in sibling subtrees with different provisional labels.

3.2 Tree-Structured Value Estimation

Tree structure connects an action to multiple downstream outcomes and allows alternative actions to be compared within a shared decision context. We use this structure to obtain chunk-level quality labels without fitting an additional value function.

Chunk-Level Value Estimation. We estimate the value of each action chunk from the outcomes of its downstream continuations. Each chunk $\mathbf{a}_i$ connects a parent node v to an imagined child u_i, whose subtree contains the continuations explored after that action. We obtain their outcomes using a task-specific Robometer (Liang et al., 2026) reward model built on Qwen3-VL-4B (Bai et al., 2025) and fine-tuned on annotated robot videos with binary cross-entropy supervision. Given visual observations and the task instruction, the model predicts task completion, which is converted into a binary outcome $y_\ell \in \{0, 1\}$ using task-specific success criteria and temporal consistency checks. The chunk value is then estimated as follows:

$$\hat{q}(v, \mathbf{a}_i) = \frac{1}{|\mathcal{L}(u_i)|} \sum_{\ell \in \mathcal{L}(u_i)} y_\ell, \tag{2}$$

where $\mathcal{L}(u_i)$ denotes the leaves in the subtree rooted at u_i. This estimate measures the empirical success rate of the explored continuations without requiring an additional learned value critic.

Relative Value Supervision. To convert chunk-level value estimates into policy supervision, we compare alternative actions from the same decision context. Let $\mathcal{C}_{\text{im}}(v)$ denote the imagined children of node v and $q_{\max}(v) = \max_{u_i \in \mathcal{C}_{\text{im}}(v)} \hat{q}(v, \mathbf{a}_i)$. For groups containing at least two actions with $q_{\max}(v) > 0$, we normalize each chunk value by the best sibling value and discretize it into K bins:

$$\rho_i = \frac{\hat{q}(v, \mathbf{a}_i)}{q_{\max}(v)}, \qquad b_i = \min(K, 1 + \lfloor K \rho_i \rfloor). \tag{3}$$

The label $b_i \in \{1, \ldots, K\}$ indicates how well an action performs relative to the best explored alternative. Each retained action contributes a sample $(o_v, g, \mathbf{a}_i, b_i)$ to $\mathcal{D}_{\text{tree}}$ for policy training.

Advantage-Conditioned Policy Update. Following RECAP (Physical Intelligence et al., 2025a), we train the policy conditioned on discretized advantage labels, using our tree-derived relative value bins as chunk-level supervision. We instantiate the policy with $\pi_{0.5}$ (Physical Intelligence et al., 2025b) and encode each bin in the policy prompt. For a normalized action chunk $\mathbf{a}$ and Gaussian

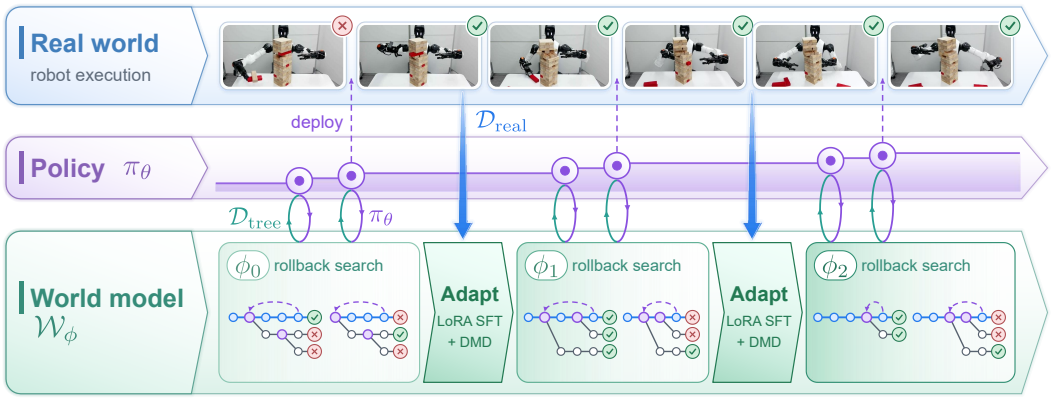


Figure 3: **Training process of Rollback.** The robot continually executes the latest policy and uploads real trajectories $\mathcal{D}_{\text{real}}$. The policy is updated with tree supervision $\mathcal{D}_{\text{tree}}$ from rollback search, and the world model is periodically adapted to real trajectories via LoRA SFT and DMD.

noise $\epsilon \sim \mathcal{N}(0, I)$, we construct $\mathbf{x}_s = (1-s)\mathbf{a} + s\epsilon$ at flow time s and optimize the flow-matching objective (Lipman et al., 2023) as follows:

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{\substack{(o,g,\mathbf{a},b)\sim\mathcal{D}_{\text{mix}} \\ s,\epsilon,\tilde{b}\sim p(\tilde{b}|b)}} \left[\left\| v_\theta(\mathbf{x}_s, s \mid o, g, \tilde{b}) - (\epsilon - \mathbf{a}) \right\|_2^2 \right], \tag{4}$$

where $\mathcal{D}_{\text{mix}}$ combines tree-derived samples with offline demonstrations and robot-collected trajectories. Successful real trajectories receive the highest bin, and failed trajectories receive the lowest bin. During training, we randomly omit the advantage condition with probability p_{drop}. During tree search and deployment, we condition on the highest bin and apply classifier-free guidance (Ho & Salimans, 2022) through $v_{\text{guided}} = v_\varnothing + \beta(v_K - v_\varnothing)$, where β controls the guidance strength.

3.3 World Model-Policy Co-Evolution

As the policy evolves, its observation-action distribution may shift beyond the world model's training data, allowing policy optimization to exploit inaccurate predictions. We therefore alternate policy improvement with world-model adaptation through supervised fine-tuning (SFT) and distribution matching distillation (DMD). At iteration k, the current policy collects successful and failed trajectories $\mathcal{D}_{\text{new}}^{(k)}$, which are added to the replay set as $\mathcal{D}_{\text{real}}^{(k+1)} = \mathcal{D}_{\text{real}}^{(k)} \cup \mathcal{D}_{\text{new}}^{(k)}$. We first adapt the world model to these real transitions using its action-conditioned prediction objective:

$$\mathcal{L}_{\text{SFT}}^{(k)}(\phi) = \mathbb{E}_{\tau\sim\mathcal{D}_{\text{real}}^{(k+1)}} \left[\sum_t \ell_{\text{WM}}\big(\phi; o_{\le t}, \mathbf{a}_t, o_{t+1}, g\big) \right]. \tag{5}$$

We then apply distribution matching distillation (DMD) (Yin et al., 2024) to obtain a few-step generator for efficient imagined interaction. The resulting world model supports experience-backtracking search to generate new tree-structured supervision.

3.4 Training Process

As shown in Figure 3, Rollback runs real-robot execution and world-model search concurrently, exchanging information only through a shared trajectory store and asynchronous weight transfer. The robot continually executes the latest policy it has received and uploads each labeled trajectory to $\mathcal{D}_{\text{real}}$; new weights take effect only between episodes, so physical data collection never waits for training. Meanwhile, GPU workers repeatedly run experience-backtracking search from offline demonstrations and the growing $\mathcal{D}_{\text{real}}$, and the policy is updated on $\mathcal{D}_{\text{mix}}$ with Equation 4 whenever M new trees are completed. Every R policy rounds, the world model is adapted to the accumulated real trajectories (Section 3.3). Full details are given in Algorithm 1 and Appendix A.5.

4 EXPERIMENTS

We evaluate Rollback on three real-world manipulation tasks to assess policy performance and data efficiency. We investigate the contributions of experience-backtracking search, tree-structured supervision, and world-model adaptation to iterative policy improvement.

4.1 EXPERIMENTAL SETUP

Tasks and Evaluation. We evaluate our proposed Rollback and baselines on the AGIBOT Genie-02 equipped with OmniPicker grippers (AGIBOT, 2026; 2025) across three real-world tasks spanning contact-rich manipulation, precise control, and competitive interaction:

- *Block Extraction.* The robot removes LEGO blocks one at a time from a stacked tower without collapsing it, testing precise manipulation under stability constraints.
- *Ball Tossing.* The robot throws a ball into a bucket, with success determined by whether the ball lands inside. This task tests precise control of the throwing motion and release timing.
- *Tabletop Curling.* A human and the robot each deliver two stones. Later shots can displace earlier stones, and the winner is determined by the score based on proximity to the target center. This task tests strategic selection and precise execution against an opponent.

For each task, we select six initial positions (nine for Tabletop Curling) and run 15 trials per position (25 for Block Extraction); per-position results are in Appendix A.3.

Implementation Details. *(1) World Model Training.* We initialize the world model from pre-trained GE-Sim 2.0 (Qiu et al., 2026) and train it in two stages: supervised fine-tuning followed by distribution matching distillation (Yin et al., 2024). We fine-tune the model on task-specific data for 10K steps using 32 NVIDIA H200 GPUs with a global batch size of 128. We then perform DMD on the same dataset for 2,500 steps using 16 NVIDIA H200 GPUs with a global batch size of 16. *(2) Policy Warm-up.* We initialize the policy from $\pi_{0.5}$ (Physical Intelligence et al., 2025b) and pre-train it on task-specific expert-collected data using RECAP-style quality conditioning (Physical Intelligence et al., 2025a). We use five quality bins, assigning successful trajectories to the highest bin 5 and failed trajectories to the lowest bin 1. Training runs for 100K steps on 16 NVIDIA H200 GPUs with a global batch size of 64. *(3) Reward Model Training.* We initialize the reward model from Robometer-4B (Liang et al., 2026), built on Qwen3-VL-4B (Bai et al., 2025), and fine-tune it on task-specific robot videos annotated with success labels. Conditioned on visual observations and the task instruction, the model predicts task completion using a binary classification head trained with binary cross-entropy loss. *(4) Policy Improvement in World Models.* We run policy improvement with one physical robot and 48 NVIDIA H200 GPUs, allocating 40 GPUs to world-model-based tree search and 8 to policy training with DDP. Following world-model warm-up, each outer iteration adds newly collected real trajectories to the replay dataset and adapts the world model through 2,000 steps of LoRA-based (Hu et al., 2022) SFT followed by 100 steps of DMD.

Baselines. We compare Rollback with imitation learning, reinforcement learning, and world-model-based baselines. All methods use the same $\pi_{0.5}$ initialization and comparable compute budgets. Implementation details and data composition are provided in the Appendix. *(1) SFT.* We fine-tune $\pi_{0.5}$ (Physical Intelligence et al., 2025b) on task-specific expert demonstrations using supervised learning. *(2) RECAP.* We adapt RECAP (Physical Intelligence et al., 2025a) to $\pi_{0.5}$, using a learned value model to estimate advantages for conditioned policy training on collected real-world data. *(3) RISE.* RISE[1] (Yang et al., 2026) learns from imagined interactions with a compositional world model that couples action-conditioned dynamics with a learned progress value model.

4.2 MAIN RESULTS

Table 1 compares Rollback with the baselines across three real-world tasks. Rollback achieves the highest success rate on all tasks, reaching 69.33% on Block Extraction, 40.00% on Ball Tossing, and 64.44% on Tabletop Curling. Rollback average success rate exceeds SFT, RECAP, and RISE

[1]For a fair comparison, we reimplement RISE with our world model and five RECAP-style advantage bins.

Table 1: **Performance on three real-world tasks.** Success rates (%) of Rollback and imitation-learning, reinforcement-learning, and world-model-based baselines, with standard deviations across initial positions.

Method	Block Extraction	Ball Tossing	Tabletop Curling	Average
$\pi_{0.5}$ + SFT	48.67 $\pm$20.77	23.33 $\pm$24.86	48.89 $\pm$27.08	40.30
$\pi_{0.5}$ + RECAP	56.00 $\pm$17.71	25.56 $\pm$22.87	51.11 $\pm$31.62	44.22
$\pi_{0.5}$ + RISE	58.89 $\pm$15.45	28.89 $\pm$24.10	52.59 $\pm$33.41	46.79
Rollback (Ours)	**69.33** $\pm$12.82	**40.00** $\pm$15.78	**64.44** $\pm$26.25	**57.93**

Table 2: **Ablation study on Block Extraction.** We evaluate world-model adaptation and tree-structured value estimation via target contact rate, task success rate, and the average number of consecutive extractions; the full framework performs best on all metrics.

Variant	Target Contact (%)	Task Success (%)	Consecutive Extractions
Frozen World Model	78.67	64.67	1.90
Direct Reward Supervision	77.33	59.33	1.73
Rollback (Full)	**89.33**	**69.33**	**2.60**

by 17.63, 13.71, and 11.14 percentage points, respectively. Compared with RISE, which also learns through world-model interaction, Rollback improves success rates by 10.44, 11.11, and 11.85 percentage points on the three tasks. These gains span contact-rich manipulation, precise throwing, and competitive interaction, indicating that the benefits of search-based training transfer to physical execution without requiring world-model search at deployment.

4.3 Ablation Studies

We conduct ablations on Block Extraction to examine the contributions of world-model adaptation and tree-structured value estimation. We report three metrics over all evaluation trials: *Target Contact* measures whether the pushing motion hits the target block; *Task Success* measures complete extraction without collapsing the remaining structure; and *Consecutive Extractions* counts the average number of blocks extracted in a row before failure. *(1) Frozen World Model.* We retain the world model obtained after initial task-specific training and disable subsequent updates from real interaction. *(2) Direct Reward Supervision.* We replace subtree-aggregated values with individual rollout rewards for quality-conditioned policy training while retaining tree search. All variants use the same policy initialization and computation budget. As shown in Table 2, removing either component degrades all three metrics; direct reward supervision lowers task success most (59.33%), and a frozen world model reduces consecutive extractions from 2.60 to 1.90.

4.4 Training Dynamics

Figure 4 tracks the real-world success rate of Rollback and RISE on Ball Tossing and Tabletop Curling over 8 hours of training. Starting from the RECAP policy, Rollback improves Ball Tossing steadily from 25.56% to 40.00%, whereas RISE fluctuates and ends at only 28.89%. Keeping the world model frozen instead yields only 26.67%, indicating that continually refining the world model with real experience is essential for sustained improvement.

4.5 Qualitative Results

Figure 6 visualizes a search tree on Tabletop Curling. Sibling subtrees branching from the same restored context lead to markedly different outcomes: the upper subtree brings the red stones to rest in the house, whereas in the lower subtree the first stone overshoots. Aggregating leaf outcomes yields subtree values of 0.67 and 0.25, which determine the relative labels of the two root actions (Section 3.2). More examples for all three tasks are provided in Appendix A.6.

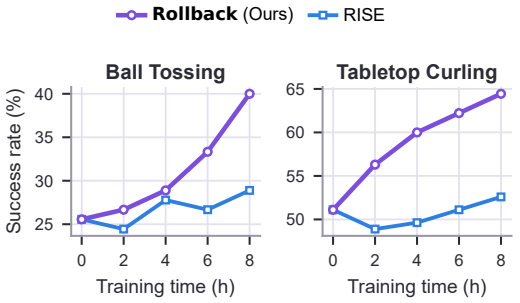


Figure 4: **Training dynamics.** Real-world success rate of Rollback and RISE versus wall-clock training time.

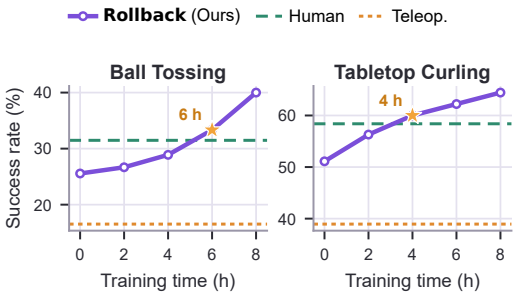


Figure 5: **Surpassing human performance.** Success rate versus training time against human players and teleoperation (dashed). Stars mark when Rollback first exceeds humans.

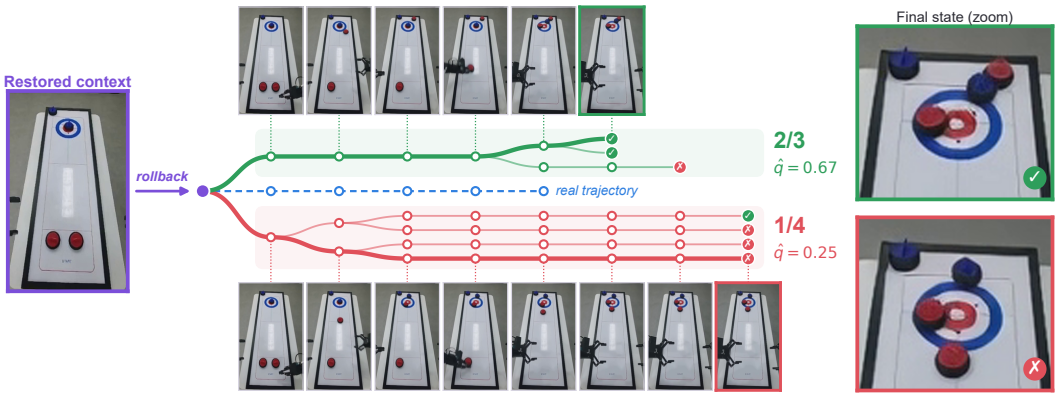


Figure 6: **Experience-backtracking search tree on Tabletop Curling.** From a restored context, Rollback branches into imagined continuations (solid) beside the real trajectory (dashed); leaf outcomes are aggregated into subtree values $\hat{q}$.

4.6 Surpassing Human Performance

We compare Rollback with human players performing each task by hand and with teleoperated data collection, both under the same initial positions and success criteria; each is averaged over three people. On Ball Tossing, human players and teleoperation achieve 31.48% and 16.53%, respectively, and Rollback surpasses the human average after 6 hours of training (33.33%), reaching 40.00% after 8 hours (Figure 5). On Tabletop Curling, human players and teleoperation achieve 58.39% and 38.94%, and Rollback surpasses humans after 4 hours, reaching 64.44% after 8 hours. Notably, although Rollback is initialized from teleoperated demonstrations far below human level, it surpasses human players on both tasks by learning from counterfactual imagination.

5 Conclusion

We presented Rollback, a policy improvement paradigm that exploits experience backtracking in world models. By revisiting decision contexts and aggregating the outcomes of alternative continuations, Rollback concentrates imagination on informative comparisons and obtains chunk-level supervision without a learned value model. A unified training system couples this search with real-robot execution, jointly improving the policy and the world model through real-world feedback. Experiments on three challenging real-world tasks demonstrate higher success rates and more stable improvement than competing baselines. Nevertheless, Rollback relies on a task-specific reward model and on world-model fidelity over long contact-rich horizons. Future work includes learning outcome detectors jointly with the world model and extending Rollback to multi-task policies.

AI USE STATEMENT

Generative AI tools were used solely for grammar correction and linguistic refinement of author-written text. All edits were reviewed by the authors, who take full responsibility for the manuscript.

ETHICS STATEMENT

All real-robot experiments were conducted in a controlled laboratory under human supervision with safety termination. Human players and teleoperators participated voluntarily and gave informed consent. Trained policies should be safety-evaluated before deployment.

REPRODUCIBILITY STATEMENT

The main paper and appendix describe the method, pseudocode, implementation details, training infrastructure, and evaluation protocol. We will release the code, checkpoints, and evaluation scripts.

REFERENCES

AGIBOT. AGIBOT OmniPicker Product Manual, 2025. URL https://www.agibot.com/DOCS/APC2026.

AGIBOT. AGIBOT G2 Universal Embodied Intelligent Robot, 2026. URL https://www.agibot.com/products/G2.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight Experience Replay, 2017. URL https://arxiv.org/abs/1707.01495.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning, 2025. URL https://arxiv.org/abs/2506.09985.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control, 2024. URL https://arxiv.org/abs/2410.24164.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-Time Execution of Action Chunking Flow Policies, 2025. URL https://arxiv.org/abs/2506.07339.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control, 2023. URL https://arxiv.org/abs/2307.15818.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of Monte Carlo Tree Search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

Zengjue Chen, Peidong Liu, Jiawei Li, and Qi Wang. HaWMPO: Hallucination-Aware World Model-based Policy Optimization for Generalist Robot Policy, 2026. URL https://arxiv.org/abs/2609.09941.

Adrien Couëtoux, Jean-Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud, and Nicolas Bonnard. Continuous upper confidence trees. In *International Conference on Learning and Intelligent Optimization*, pp. 433–445, 2011.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore, 2020. URL https://arxiv.org/abs/2004.12919.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing Function Approximation Error in Actor-Critic Methods, 2018. URL https://arxiv.org/abs/1802.09477.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, Qianli Ma, Seungjun Nah, Loic Magne, Jiannan Xiang, Yuqi Xie, Ruijie Zheng, Dantong Niu, You Liang Tan, K. R. Zentner, George Kurian, Suneel Indupuru, Pooya Jannaty, Jinwei Gu, Jun Zhang, Jitendra Malik, Pieter Abbeel, Ming-Yu Liu, Yuke Zhu, Joel Jang, and Linxi "Jim" Fan. DreamDojo: A Generalist Robot World Model from Large-Scale Human Videos, 2026. URL https://arxiv.org/abs/2602.06949.

Gemini Robotics Team, Krzysztof Choromanski, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Isabel Leal, Fangchen Liu, Anirudha Majumdar, Andrew Marmon, Carolina Parada, Yulia Rubanova, Dhruv Shah, Vikas Sindhwani, Jie Tan, Fei Xia, Ted Xiao, Sherry Yang, Wenhao Yu, and Allan Zhou. Evaluating Gemini Robotics Policies in a Veo World Simulator, 2025. URL https://arxiv.org/abs/2512.10675.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A Controllable Generative World Model for Robot Manipulation, 2025a. URL https://arxiv.org/abs/2510.10125.

Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving Vision-Language-Action Model with Online Reinforcement Learning, 2025b. URL https://arxiv.org/abs/2501.16664.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning Latent Dynamics for Planning from Pixels, 2018. URL https://arxiv.org/abs/1811.04551.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to Control: Learning Behaviors by Latent Imagination, 2019. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with Discrete World Models, 2020. URL https://arxiv.org/abs/2010.02193.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Siyuan Huang, Liliang Chen, Pengfei Zhou, Shengcong Chen, Zhengkai Jiang, Yue Hu, Yue Liao, Peng Gao, Hongsheng Li, Maoqing Yao, and Guanghui Ren. EnerVerse: Envisioning Embodied Future Space for Robotics Manipulation, 2025. URL https://arxiv.org/abs/2501.01895.

Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Mohammadamin Barekatain, Simon Schmitt, and David Silver. Learning and Planning in Complex Action Spaces, 2021. URL https://arxiv.org/abs/2104.06303.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, Loic Magne, Ajay Mandlekar, Avnish Narayan, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Yinzhen Xu, Xiaohui Zeng, Kaiyuan Zheng, Ruijie Zheng, Ming-Yu Liu, Luke Zettlemoyer, Dieter Fox, Jan Kautz, Scott Reed, Yuke Zhu, and Linxi Fan. DreamGen: Unlocking Generalization in Robot Learning through Video World Models, 2025. URL https://arxiv.org/abs/2505.12705.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to Trust Your Model: Model-Based Policy Optimization, 2019. URL https://arxiv.org/abs/1906.08253.

Yuxin Jiang, Shengcong Chen, Siyuan Huang, Liliang Chen, Pengfei Zhou, Yue Liao, Xindong He, Chiming Liu, Hongsheng Li, Maoqing Yao, and Guanghui Ren. EnerVerse-AC: Envisioning Embodied Environments with Action Condition, 2025. URL https://arxiv.org/abs/2505.09723.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, Yu Wang, Haoran Li, Chao Yu, and Dongbin Zhao. WoVR: World Models as Reliable Simulators for Post-Training VLA Policies with RL, 2026. URL https://arxiv.org/abs/2602.13977.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success, 2025. URL https://arxiv.org/abs/2502.19645.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, Dehui Wang, Dingxiang Luo, Yuchen Fan, Youbang Sun, Jia Zeng, Jiangmiao Pang, Shanghang Zhang, Yu Wang, Yao Mu, Bowen Zhou, and Ning Ding. SimpleVLA-RL: Scaling VLA Training via Reinforcement Learning, 2025a. URL https://arxiv.org/abs/2509.09674.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control, 2026. URL https://arxiv.org/abs/2601.21998.

Yaxuan Li, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. WorldEval: World Model as Real-World Robot Policies Evaluator, 2025b. URL https://arxiv.org/abs/2505.19017.

Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S. Huang, Luke Zettlemoyer, Dieter Fox, Yu Xiang, Anqi Li, Andreea Bobu, Abhishek Gupta, Stephen Tu, Erdem Biyik, and Jesse Zhang. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. In *Robotics: Science and Systems*, 2026. URL https://arxiv.org/abs/2603.02115.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, Liliang Chen, Shuicheng Yan, Maoqing Yao, and Guanghui Ren. Genie Envisioner: A Unified World Foundation Platform for Robotic Manipulation, 2025. URL https://arxiv.org/abs/2508.05635.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. VLA-RL: Towards Masterful and General Robotic Manipulation with Scalable Reinforcement Learning, 2025. URL https://arxiv.org/abs/2505.18719.

NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World Simulation with Video Foundation Models for Physical AI, 2025. URL https://arxiv.org/abs/2511.00062.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi^*_{0.6}$: a VLA That Learns From Experience, 2025a. URL https://arxiv.org/abs/2511.14759.

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, 2025b. URL https://arxiv.org/abs/2504.16054.

Boxiang Qiu, Liliang Chen, Yue Liao, Nan Wang, Lintao Wang, Jiayi Luo, Wenzhi Zhao, Shengcong Chen, Di Chen, Ye Li, et al. GE-Sim 2.0: A roadmap towards comprehensive closed-loop video world simulators for robotic manipulation. *arXiv preprint arXiv:2605.27491*, 2026.

Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. WorldGym: World Model as An Environment for Policy Evaluation, 2025. URL https://arxiv.org/abs/2506.00613.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing Open-source World Models, 2026. URL https://arxiv.org/abs/2601.20540.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation, 2015. URL https://arxiv.org/abs/1506.02438.

Ansh Kumar Sharma, Yixiang Sun, Ninghao Lu, Yunzhe Zhang, Jiarao Liu, and Sherry Yang. World-Gymnast: Training Robots with Reinforcement Learning in a World Model, 2026. URL https://arxiv.org/abs/2602.02454.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. doi: 10.1038/nature16961.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive Post-Training for Vision-Language-Action Models, 2025. URL https://arxiv.org/abs/2505.17016.

Yixuan Wang, Rhythm Syed, Fangyu Wu, Mengchao Zhang, Aykut Onol, Jose Barreiros, Hooshang Nayyeri, Tony Dear, Huan Zhang, and Yunzhu Li. Interactive World Simulator for Robot Policy Training and Evaluation, 2026. URL https://arxiv.org/abs/2603.08546.

Junjin Xiao, Yandan Yang, Xinyuan Chang, Ronghan Chen, Feng Xiong, Mu Xu, Wei-Shi Zheng, and Qing Zhang. World-Env: Leveraging World Model as a Virtual Environment for VLA Post-Training, 2025. URL https://arxiv.org/abs/2509.24948.

Jiazhi Yang, Kunyang Lin, Jinwei Li, Wencong Zhang, Tianwei Lin, Longyan Wu, Zhizhong Su, Hao Zhao, Ya-Qin Zhang, Li Chen, Ping Luo, Xiangyu Yue, and Hongyang Li. RISE: Self-Improving Robot Policy with Compositional World Model, 2026. URL https://arxiv.org/abs/2602.11075.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning Interactive Real-World Simulators, 2023. URL https://arxiv.org/abs/2310.06114.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based Offline Policy Optimization, 2020. URL https://arxiv.org/abs/2005.13239.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: Conservative Offline Model-Based Policy Optimization, 2021. URL https://arxiv.org/abs/2102.08363.

Hongzhi Zang, Shu'ang Yu, Hao Lin, Tianxing Zhou, Zefang Huang, Zhen Guo, Xin Xu, Jiakai Zhou, Yuze Sheng, Shizhe Zhang, Feng Gao, Wenhao Tang, Yufeng Yue, Quanlu Zhang, Xinlei Chen, Chao Yu, and Yu Wang. RLinf-USER: A Unified and Extensible System for Real-World Online Policy Learning in Embodied AI, 2026. URL https://arxiv.org/abs/2602.07837.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A Fine-Grained World Model for Robot Manipulation, 2024. URL https://arxiv.org/abs/2406.14540.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. WMPO: World Model-based Policy Optimization for Vision-Language-Action Models, 2025. URL https://arxiv.org/abs/2511.09515.

Algorithm 1 Rollback training with world-model–policy co-evolution

Require: Offline demonstrations $\mathcal{D}_{\text{demo}}$; initial real replay buffer $\mathcal{D}_{\text{real}}$; policy π_θ; world model $\mathcal{W}_\phi$; task-specific Robometer reward model; number of co-evolution cycles C; policy rounds per cycle R; completed-tree threshold M
Ensure: Updated policy π_θ and world model $\mathcal{W}_\phi$

Initialize an empty tree buffer $\mathcal{B}_{\text{tree}}$
Launch asynchronous real-robot execution with the current rollout policy
for $k = 0, \ldots, C-1$ **do**
  for $r = 1, \ldots, R$ **do**
    Drain newly completed trajectories $\mathcal{D}_{\text{new}}$ and set $\mathcal{D}_{\text{real}} \leftarrow \mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{new}}$
    while fewer than M new trees have completed **do**
      In parallel, sample a seed trajectory $\tau \sim \mathcal{D}_{\text{demo}} \cup \mathcal{D}_{\text{real}}$
      $\mathcal{T}_\tau \leftarrow \textsc{BacktrackingTreeSearch}(\tau, \pi_\theta, \mathcal{W}_\phi)$
      $\mathcal{D}_\tau \leftarrow \textsc{ExtractTreeLabels}(\mathcal{T}_\tau, K)$
      Append $\mathcal{D}_\tau$ to $\mathcal{B}_{\text{tree}}$
    end while
    Pause policy sampling; retain unfinished trees and cached world-model states
    Construct $\mathcal{D}_{\text{mix}}$ from $\mathcal{B}_{\text{tree}}$, $\mathcal{D}_{\text{demo}}$, and $\mathcal{D}_{\text{real}}$
    Update θ on $\mathcal{D}_{\text{mix}}$ using the advantage-conditioned policy objective with distributed data parallelism
    Update the rollout weights and synchronize them to search workers and the robot
    Resume unfinished searches; apply new robot weights only between episodes
  end for
  Drain any remaining completed trajectories into $\mathcal{D}_{\text{real}}$
  Adapt $\mathcal{W}_\phi$ on $\mathcal{D}_{\text{real}}$ by SFT using the action-conditioned prediction objective
  Distill the adapted model into the few-step simulator by DMD
end for
return $\pi_\theta, \mathcal{W}_\phi$

A Appendix

A.1 Experiment Details

The main text describes the three tasks and the headline evaluation protocol. Here, we provide the execution-level details needed for reproducibility.

Observation and control interface. All policies receive three RGB views (one head-mounted camera and two wrist cameras) together with the robot's proprioceptive state. Images from both real execution and imagined rollouts are mapped to a common 384×512 canvas before the policy-specific image transforms are applied. The deployed action has 16 dimensions, comprising seven arm and one gripper coordinate per side. The low-level control loop runs at 30 Hz. For block extraction and ball tossing, the policy predicts 50-step action chunks and is executed with real-time chunking (Black et al., 2025) using a delay of 6 steps and a replanning stride of 32 steps. The same observation processing, action interface, and controller settings are used for all compared policies.

Trial initialization and termination. Before each trial, the robot is returned to a task-specific reset configuration sampled from a pool constructed from the beginning of successful demonstration segments. This avoids evaluating every policy from a single fixed arm pose while keeping the reset distribution identical across methods. Object configurations are then randomized according to the task protocol. Block-extraction episodes have a hard limit of 700 control steps (23.3 s), whereas ball-tossing episodes are limited to 120 steps (4.0 s). An episode terminates early once the outcome is determined or a safety stop is issued. A tabletop-curling trial ends after the human and robot have each completed two shots, and the winner is determined from the final stone configuration.

Outcome annotation. Real-robot trajectories are assigned binary success labels from their observed physical outcomes. Block extraction is successful only when the target block is removed while the tower remains intact; a collapse or an unlabeled termination is treated as failure. Ball tossing is successful only when the ball lands in the bucket, and all other terminations are failures. Curling is scored only after all four shots so that displacement by later shots is accounted for. Impor-

tantly, the task-specific Robometer reward model is used to evaluate imagined world-model rollouts only; it does not determine the reported real-world success rates.

A.2 HYPERPARAMETERS

Table 3 lists the hyperparameters of Rollback; task-specific values are given as Block Extraction / Ball Tossing / Tabletop Curling.

Table 3: **Hyperparameters of Rollback.**

Component	Hyperparameter	Value
World model	Initialization	GE-Sim 2.0
	Initial SFT (steps / GPUs / batch)	10K / 32 / 128
	Initial DMD (steps / GPUs / batch)	2,500 / 16 / 16
	Per-iteration adaptation	2,000 LoRA SFT + 100 DMD steps
	Frames per action chunk / frame rate	25 / 16 fps
Policy	Initialization	$\pi_{0.5}$
	Warm-up (steps / GPUs / batch)	100K / 16 / 64
	Advantage bins K	5
	Condition dropout p_{drop}	0.5
	Guidance strength β	1.5
	Action chunk / control frequency	50 steps / 30 Hz
Reward model	Initialization	Robometer-4B
	Success threshold	0.5
Search	PUCT coefficient c_{puct}	1.5
	Widening parameters (c_{pw}, α)	(1.5, 0.5) / (4.0, 0.5) / (1.5, 0.5)
	Actions per new branch	2
	Confirmation count n_{conf}	5 / 9 / 5
	Imagination budget B (action chunks)	256
	Maximum search depth (action chunks)	12 / 2 / 8
	Trees per GPU per policy update	1
System	GPUs for search / policy training	40 / 8 (H200)
	Policy rounds per world-model update R	5
	Physical robots	1

A.3 PER-POSITION RESULTS

Tables 4–6 report success rates at each initial position. Every policy is evaluated for 25 trials per position on Block Extraction and 15 trials per position on Ball Tossing and Tabletop Curling. For Block Extraction, we additionally report the average number of blocks extracted in a row before failure; for Tabletop Curling, we report the average score obtained against the human opponent.

Table 4: **Per-position results on Block Extraction** (success rate, %; 25 trials per position).

Method	P1	P2	P3	P4	P5	P6	Overall	Consec. Extr.
$\pi_{0.5}$ + SFT	64.00	80.00	44.00	40.00	44.00	20.00	48.67	1.60
$\pi_{0.5}$ + RECAP	48.00	84.00	40.00	48.00	72.00	44.00	56.00	1.78
Frozen World Model	72.00	76.00	48.00	68.00	60.00	64.00	64.67	1.90
Direct Reward Supervision	76.00	64.00	48.00	64.00	64.00	40.00	59.33	1.73
$\pi_{0.5}$ + RISE	60.00	80.00	33.33	66.67	53.33	60.00	58.89	2.00
Rollback (Ours)	72.00	56.00	56.00	64.00	84.00	84.00	**69.33**	**2.60**

A.4 DETAILED TRAINING ALGORITHM

Algorithm 1 summarizes the overall training loop. Tree search and real-robot execution proceed asynchronously, whereas policy updates are synchronized across workers. The world model remains

Table 5: **Per-position results on Ball Tossing** (success rate, %; 15 trials per position).

Method	P1	P2	P3	P4	P5	P6	Overall
$\pi_{0.5}$ + SFT	13.33	60.00	0.00	0.00	46.67	20.00	23.33
$\pi_{0.5}$ + RECAP	0.00	60.00	20.00	13.33	46.67	13.33	25.56
Frozen World Model	0.00	46.67	26.67	13.33	53.33	20.00	26.67
$\pi_{0.5}$ + RISE	13.33	73.33	13.33	13.33	20.00	40.00	28.89
Rollback (Ours)	26.67	60.00	33.33	46.67	53.33	20.00	**40.00**

Table 6: **Per-position results on Tabletop Curling** (success rate, %; 15 trials per position).

Method	P1	P2	P3	P4	P5	P6	P7	P8	P9	Overall	Score
$\pi_{0.5}$ + SFT	60.00	86.67	33.33	26.67	60.00	0.00	40.00	80.00	53.33	48.89	0.35
$\pi_{0.5}$ + RECAP	60.00	66.67	13.33	26.67	60.00	0.00	60.00	73.33	100.00	51.11	0.44
Frozen World Model	53.33	53.33	13.33	6.67	33.33	46.67	80.00	100.00	86.67	52.59	0.51
$\pi_{0.5}$ + RISE	20.00	86.67	20.00	26.67	40.00	80.00	20.00	80.00	100.00	52.59	0.43
Rollback (Ours)	86.67	80.00	60.00	20.00	20.00	80.00	80.00	80.00	73.33	**64.44**	**0.79**

fixed during a block of policy updates and is subsequently adapted using the accumulated real-robot feedback. Algorithm 2 details the experience-backtracking search, and Algorithm 3 gives the exact rule used to extract tree-structured supervision.

Each call to EXPANDANDROLLOUT first restores m_v, samples an action chunk from $\pi_\theta(\cdot \mid o_v, g, b = K)$ using a fresh noise seed, and steps $\mathcal{W}_\phi$. The resulting world-model state is cached at the new child. The highest-bin policy is then rolled out until success or D_τ, caching every intermediate node. The routine returns the terminal leaf and the number of simulated action chunks consumed. In SELECTPUCTPW, existing children use the PUCT score defined in the main text with uniform sibling priors. A node may contain at most $\lceil c_{\mathrm{pw}} \max(1, N(v))^\alpha \rceil$ children. Whenever widening is permitted, a virtual child with zero empirical value and prior $1/(|\mathcal{C}(v)| + 1)$ competes with existing children; selecting it returns v for expansion. Restoring m_v before every expansion ensures that all siblings share the same decision context without regenerating their common trajectory prefix. The budget B counts simulated action chunks and is soft because an initiated rollout and its atomic sibling pair are completed before termination; truncating either would not yield comparable outcomes. In our mixed-only configuration, the first imagined sibling pair is always admitted, whereas later widening is restricted to mixed groups and capped at a task-specific maximum width.

A.5 TRAINING INFRASTRUCTURE

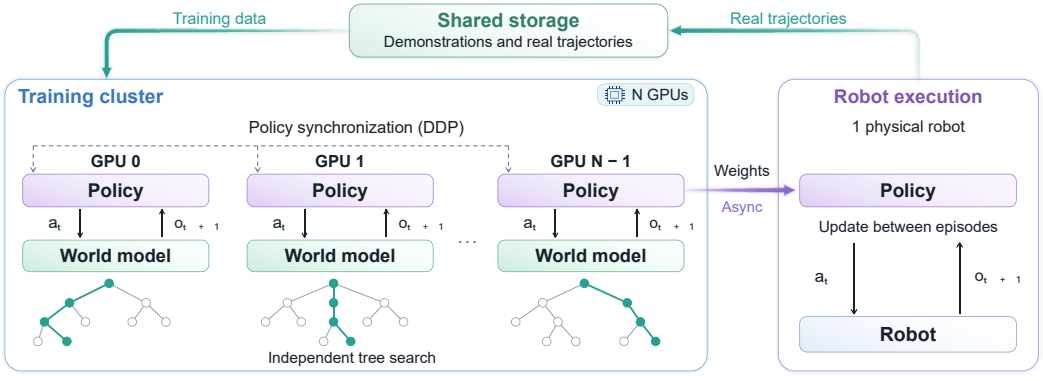


Figure 7: **Rollback training infrastructure.** GPU workers perform independent tree search and synchronize policy updates via DDP. The robot receives updated weights asynchronously, while real trajectories are uploaded to shared storage for subsequent search and training.

Algorithm 2 Experience-backtracking tree search

Require: Seed trajectory τ with instruction g and binary outcome $y_\tau \in \{0, 1\}$; policy π_θ; world model $\mathcal{W}_\phi$; task-specific Robometer reward model; imagination budget B; maximum search depth $D_{\max}$; number of bins K; PUCT coefficient c_{puct}; widening parameters (c_{pw}, α); confirmation count n_{conf}
Ensure: Imagination tree $\mathcal{T}$

Build a reference spine $\mathcal{T}$ from observations of τ at action-chunk boundaries
Cache the root world-model state; materialize and cache other spine states on first use
Mark the final spine node terminal and back up $y_\tau \in \{0, 1\}$ along the spine
Set depth limit D_τ to the spine depth if $y_\tau = 1$, and to $D_{\max}$ otherwise
Initialize $n_{\text{sim}} \leftarrow 0$ and an empty queue $\mathcal{Q}_{\text{mix}}$
while $n_{\text{sim}} < B$ and $\mathcal{T}$ contains an expandable node **do**
    $v \leftarrow \textsc{SelectPUCTPW}(\mathcal{T}, c_{\text{puct}}, c_{\text{pw}}, \alpha)$ ▷ PUCT with progressive widening
    if $v = \varnothing$ **then**
        break
    end if
    $n_{\text{new}} \leftarrow 2$ if v has no imagined child; otherwise $n_{\text{new}} \leftarrow 1$
    for $j = 1, \ldots, n_{\text{new}}$ **do**
        $(\ell, \Delta n) \leftarrow \textsc{ExpandAndRollout}(v, D_\tau)$
        $n_{\text{sim}} \leftarrow n_{\text{sim}} + \Delta n$
        Obtain $y_\ell \in \{0, 1\}$ from the task-specific reward model
        Back up y_ℓ from leaf ℓ to the root: $N \leftarrow N + 1$, $W \leftarrow W + y_\ell$, and $Q \leftarrow W/N$
    end for
    Recompute the relative bins of the imagined children of v
    if the children of v occupy different bins **then**
        Push v into $\mathcal{Q}_{\text{mix}}$
    end if
    while $\mathcal{Q}_{\text{mix}} \neq \varnothing$ and $n_{\text{sim}} < B$ **do**
        Pop a parent p from $\mathcal{Q}_{\text{mix}}$
        while an expandable imagined child $u \in \mathcal{C}_{\text{im}}(p)$ has $N(u) < n_{\text{conf}}$ and $n_{\text{sim}} < B$ **do**
            Choose the least-visited such child $u \leftarrow \arg\min_{u'} N(u')$
            Select an expandable descendant v' within the subtree rooted at u
            $(\ell, \Delta n) \leftarrow \textsc{ExpandAndRollout}(v', D_\tau)$
            $n_{\text{sim}} \leftarrow n_{\text{sim}} + \Delta n$
            Obtain y_ℓ from the reward model; back up y_ℓ and enqueue any new mixed group
        end while
    end while
end while
return $\mathcal{T}$

Figure 7 illustrates the training infrastructure of Rollback, which connects a physical robot to a training cluster with N GPUs through shared storage and asynchronous weight transfer. The robot executes the policy and uploads real trajectories, while each GPU hosts a policy replica and a world-model instance for independent tree search. Offline demonstrations and newly collected trajectories provide search initializations and direct policy supervision. To accommodate variable search durations, policy training is triggered when completed trees reach a global threshold without requiring a fixed contribution from each worker. Workers pause search and synchronize policy updates through distributed data parallelism (DDP), retaining unfinished trees and cached world-model contexts for subsequent continuation. Updated weights are transferred asynchronously to the robot and applied between episodes, keeping the policy fixed throughout each physical rollout. Robot execution thus proceeds independently of cluster updates while continually supplying real experience.

A.6 Additional Qualitative Results

Search tree on Tabletop Curling. Figure 8 shows another search tree with its recorded leaf outcomes. The two root subtrees attain 2/3 and 1/3 successful continuations, yielding subtree values of 0.67 and 0.33, while branches rolled back from the real trajectory probe a late decision point.

Sibling continuations on Ball Tossing. Figure 9 compares four sibling branches expanded from the same context. Although all branches share the same history, the sampled action chunks produce distinct ball trajectories, illustrating the diversity of explored counterfactual continuations.

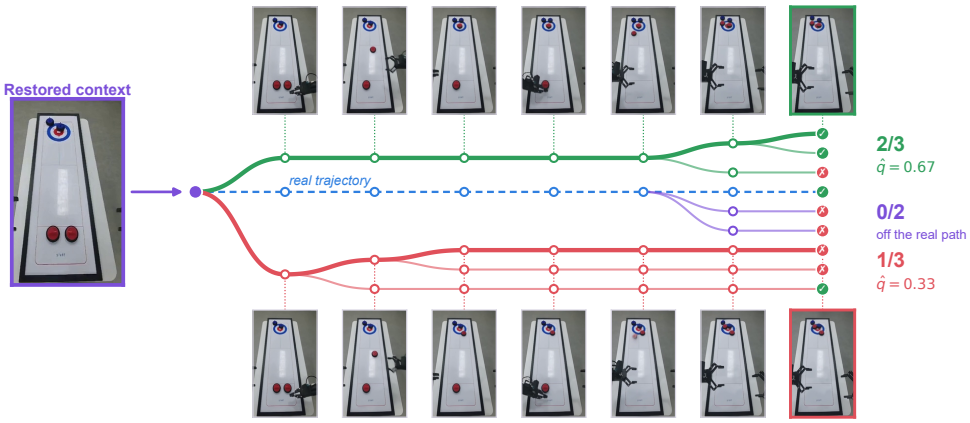


Figure 8: **Search tree on Tabletop Curling with recorded leaf outcomes.** Besides branching from the restored context, Rollback rolls back to an intermediate decision point on the real trajectory (purple) and explores alternative continuations from there.

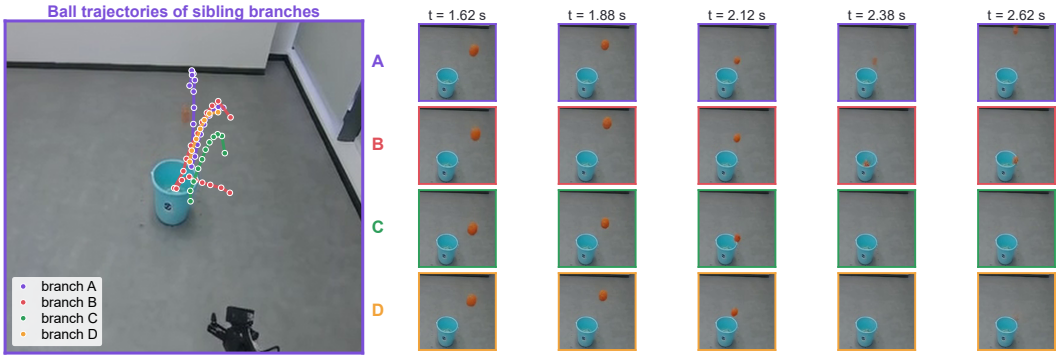


Figure 9: **Sibling continuations on Ball Tossing.** Left: ball trajectories of four sibling branches expanded from the same context. Right: close-ups around the bucket for each branch.

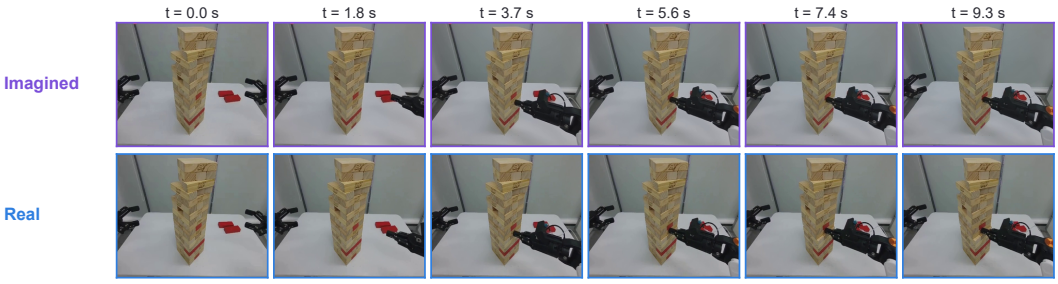


Figure 10: **World-model fidelity on Block Extraction.** Open-loop world-model predictions (top) and the real recording (bottom) under an identical action sequence.

Algorithm 3 Extraction of tree-structured relative-value supervision

Require: Imagination tree $\mathcal{T}$; number of bins K
Ensure: Tree-derived training set $\mathcal{D}_{\text{tree}}$

$\mathcal{D}_{\text{tree}} \leftarrow \varnothing$
for each parent v with imagined children $\mathcal{C}_{\text{im}}(v)$ **do**
  if $|\mathcal{C}_{\text{im}}(v)| < 2$ **then**
    continue
  end if
  for each action-child pair (a_i, u_i) of v **do**
    $\widehat{q}(v, a_i) \leftarrow |\mathcal{L}(u_i)|^{-1} \sum_{\ell \in \mathcal{L}(u_i)} y_\ell$
  end for
  $q_{\max}(v) \leftarrow \max_i \widehat{q}(v, a_i)$
  if $q_{\max}(v) = 0$ **then**
    continue ▷ No successful continuation was observed
  end if
  for each (a_i, u_i) **do**
    $\rho_i \leftarrow \widehat{q}(v, a_i)/q_{\max}(v)$
    $b_i \leftarrow \min(K, 1 + \lfloor K\rho_i \rfloor)$
  end for
  if $|\{b_i\}_i| = 1$ **then**
    continue ▷ Settled group; no relative preference signal
  end if
  for each (a_i, b_i) **do**
    $\mathcal{D}_{\text{tree}} \leftarrow \mathcal{D}_{\text{tree}} \cup \{(o_v, g, a_i, b_i)\}$
  end for
end for
return $\mathcal{D}_{\text{tree}}$

World-model fidelity on Block Extraction. Figure 10 compares open-loop world-model predictions with the real recording under an identical action sequence. The predicted frames closely track the real ones over the 9.3 s horizon, including the arm approach and contact with the target block, with a PSNR of 22.7 dB averaged over the three camera views.

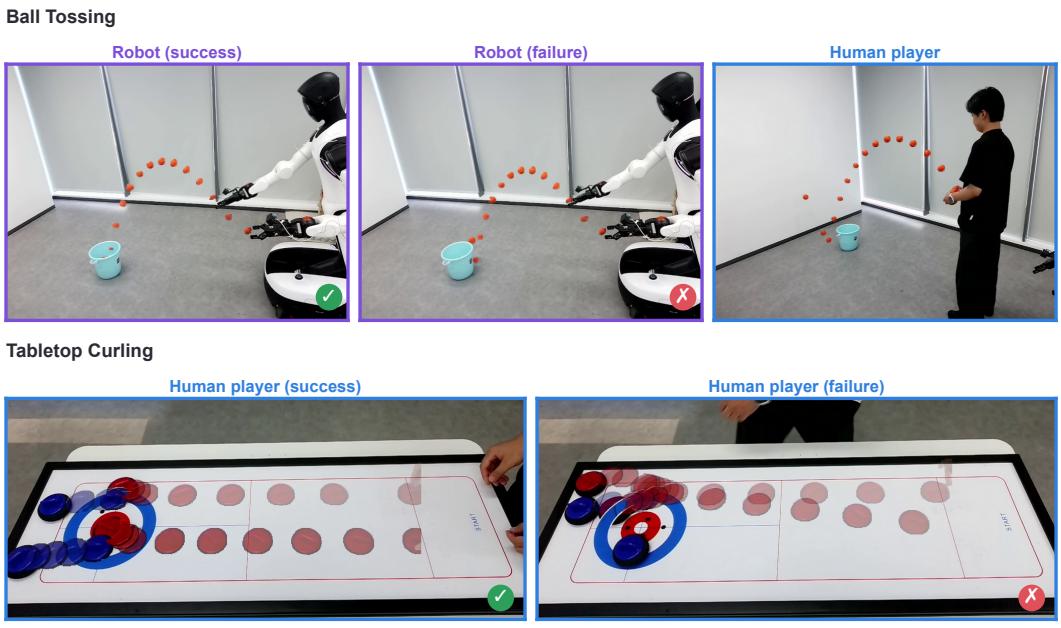


Figure 11: **Real-world executions.** Stroboscopic composites overlay the ball (top) and sliding stones (bottom) over time onto a single frame. Top: ball tossing by the robot and by a human player. Bottom: tabletop curling by human players.

Real-world executions. Figure 11 visualizes real-world executions with stroboscopic composites. In ball tossing, successful and failed robot throws differ mainly in the release that determines the landing point, with failures bouncing off the rim. In tabletop curling, successful human deliveries bring the red stones to rest in the house while displacing the opponent's stones, whereas failed deliveries leave them outside the scoring area.

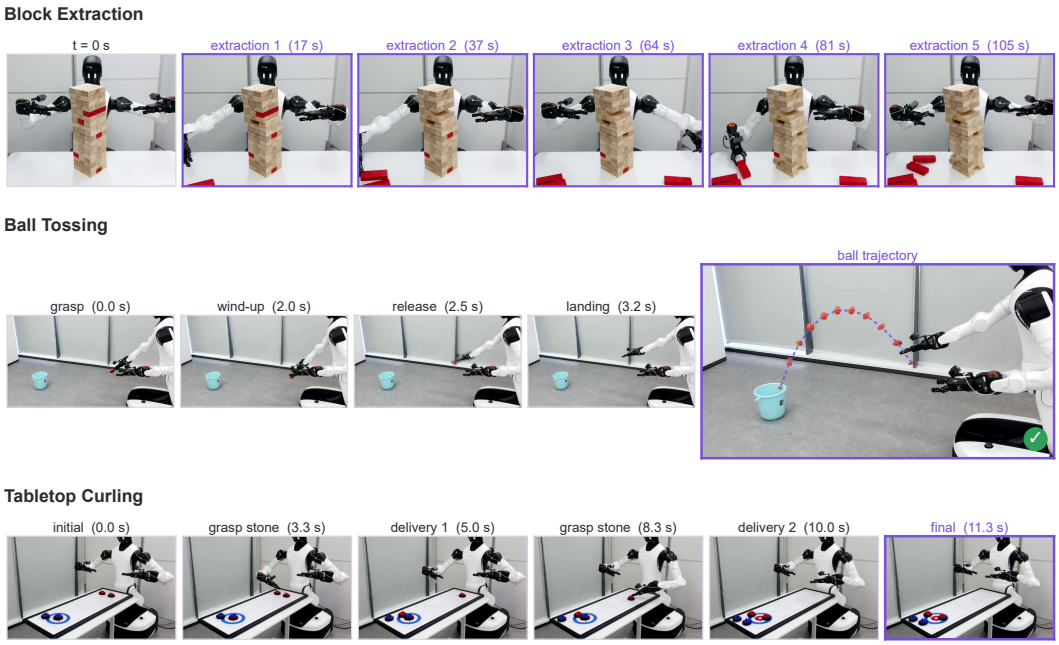


Figure 12: **Third-person views of Rollback on the real robot.** Top: five consecutive extractions without collapsing the tower. Middle: a successful toss, with the ball trajectory overlaid on the final frame. Bottom: two stone deliveries that leave the red stones in the house.

Third-person executions. Figure 12 shows real-robot executions of the final Rollback policy: five consecutive block extractions, a successful toss, and two curling deliveries.

A.7 World-Model Fidelity and Reward Reliability

To measure how far imagined rollouts drift from reality, we replay the recorded actions of eight failed real Block Extraction episodes open-loop in the co-evolved world model, five times each, and compare the predictions with the real recordings (Table 7). The average PSNR is 21.4 dB, highest for the head camera (23.4 dB), whose static background is easier to predict than the wrist views. The per-pixel L1 error grows from 6.8 in the first quarter of each rollout to 11.8 in the last quarter, a 1.78× increase over 150–275 frames, consistent with our choice to branch from real observations rather than roll out long imagined horizons. The reward model reports success on none of the 40 rollouts, all of which replay failed episodes.

Table 7: **Open-loop world-model fidelity on Block Extraction** (40 rollouts; mean ± std).

PSNR (dB) ↑			RGB L1 ↓		Reward false positives
Head	Left wrist	Right wrist	First quarter	Last quarter	
23.41 ± 0.74	20.63 ± 1.71	20.85 ± 0.73	6.79 ± 1.07	11.79 ± 1.67	0 / 40

A.8 Failure Modes

Figure 13 shows the four failure types we observe on Ball Tossing. Short and overshooting throws come from errors in release velocity, and rim bounces from small errors in release direction; these

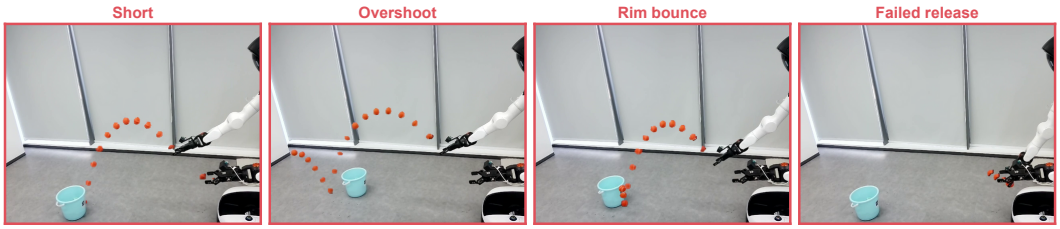


Figure 13: **Typical failures on Ball Tossing.** Stroboscopic composites of unsuccessful throws sampled from real-robot data collected during training.

three types account for most failures. Failed releases, where the gripper opens too late and the ball drops near the robot, are less frequent. All four types are decided within a short window around release, so a single action chunk largely determines the outcome of the whole episode.