ITERATIVE SELF-IMPROVEMENT WITH WORLD-EXPECTATION OPTIMIZATION FOR WORLD-ACTION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot foundation models have enabled increasingly general manipulation through large-scale pretraining. Recent world action models (WAMs) further unify action generation and future prediction, providing a foundation for robots to learn both how to act and what their actions cause. As robots interact with their environments, each execution provides new evidence about the effectiveness of their actions and the accuracy of their predictions, creating an opportunity for continued self-improvement beyond pretraining. However, existing self-improvement approaches often evolve policies with external predictive models, without directly improving predictive capabilities within the action-generating model itself. We introduce **World-ISI**, an iterative framework that turns this execution feedback into the joint evolution of action generation, world prediction, and value estimation within a single WAM. To leverage the WAM's own predictions of future task outcomes during self-improvement, we further propose World-Expectation Optimization (**WEO**). WEO uses the difference between observed returns and the model's own value predictions to weight learning from successful actions, while using failed executions in dedicated world- and value-prediction modes and retaining auxiliary prediction losses on successful policy samples. Applied to Cosmos Policy, World-ISI improves average success on 24 RoboCasa tasks from 66.7% to 70.1% after four iterations. In a controlled comparison at 6,000 training steps, WEO achieves 69.0% success versus 67.5% for standard supervised fine-tuning, supporting the effectiveness of learning from the model's own expectations.

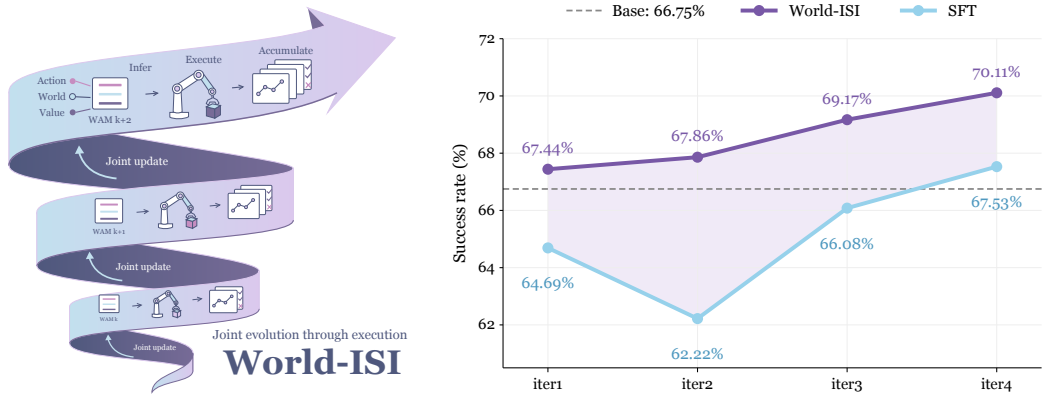

Figure 1: **World-ISI: joint evolution through execution. Left:** A unified world action model repeatedly interacts with the environment, accumulates experience, and jointly updates action generation, world prediction, and value estimation. **Right:** RoboCasa success rates over four iterations, comparing World-ISI with SFT at matched cumulative training steps. World-ISI outperforms SFT at every post-training checkpoint.

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.

1 INTRODUCTION

Robot foundation models pretrained on large-scale, multi-task datasets have substantially improved generalization across environments and instructions. Vision-language-action (VLA) models generate robot actions conditioned on visual observations and language instructions. More recently, world action models (WAMs), such as Cosmos Policy (Kim et al., 2026) and DreamZero (Ye et al., 2026b), jointly model robot actions and future visual observations within a unified model. By jointly modeling robot actions and future visual observations, these models learn to generate actions while anticipating their consequences. This integration allows action generation and future prediction to share representations of the environment. However, these pretrained capabilities do not automatically translate into continual improvement. Learning from new experience collected during deployment remains a key challenge for turning pretrained WAMs into continuously improving agents.

Prior self-improvement frameworks, such as Q-Planning (Giridhar et al., 2026), VLAW (Guo et al., 2026), and RISE (Yang et al., 2026), rely on value functions, reward models, or world models trained separately from the policy to guide learning or action selection. However, co-evolving world modeling and action predicting capabilities within the same model can allow supervision of actual consequences to shape the representations used for action generation, while refining the value estimates that guide subsequent policy updates. We therefore investigate how execution experience can drive the joint evolution of a robot's action generation, future prediction, and value estimation through repeated interaction.

To address this gap, we propose **World-ISI**, an iterative self-improvement framework that enables a single WAM to jointly evolve its action generation, world prediction, and value estimation through accumulated execution experience. Each iteration comprises two stages: experience accumulation and joint model update. The robot collects new trajectories through environment interaction and adds them to the experience retained from previous rounds. The model is then jointly updated using this growing pool of experience, learning from both recent interactions and past executions. The updated model returns to the environment to collect further experience, forming an iterative self-improvement loop in which each stage of learning builds on what came before.

However, directly imitating all accumulated experience can teach the policy to repeat failed actions. Simply keeping successful executions is also not enough: successes that the model did not expect may offer more useful learning signals. To leverage the WAM's predictions of future task outcomes, we propose **World-Expectation Optimization (WEO)**. WEO compares observed returns with the model's collection-time value predictions and assigns greater action-learning weight to successful executions that were less expected to succeed. At the same time, the model's world and value predictions can also be calibrated by actual environment interactions. Observed future states supervise world prediction, while observed task returns supervise value estimation. We train these predictions together with the action policy, so each round updates both the model's behavior and its expectations.

We evaluate World-ISI over four self-improvement iterations on 24 RoboCasa manipulation tasks and further study the framework on WidowX and ALOHA robots. World-ISI improves average RoboCasa success from 66.7% to 70.1%. Under a matched 6,000-step optimization budget and fixed training data, WEO achieves 69.0% average success compared with 67.5% for supervised fine-tuning (SFT). We also examine the roles of prediction supervision by removing the value loss alone or both the world and value losses while retaining expectation-weighted action learning. These comparisons separate the training objective, prediction supervision, and iterative experience collection.

Our contributions are:

- We propose World-ISI, an iterative self-improving framework that jointly updates action generation, world prediction, and value estimation within a unified world action model using self-collected execution experience.
- We introduce World-Expectation Optimization (WEO), which weights successful action learning by the difference between observed returns and the model's own value predictions, while learning from executions through dedicated world and value prediction.
- Our experiments demonstrate that World-ISI improves manipulation performance on RoboCasa tasks and on real-world WidowX and ALOHA robots. Ablation studies further show the benefits of our proposed WEO and World-ISI self-improvement.

2 Related Work

2.1 World Action Models for Robotics

World action models connect predictive world modeling with robot control. One line of work uses video models as predictive representations: Video Policy (Liang et al., 2025) decodes actions from video features, Genie Envisioner (Liao et al., 2025) builds control and simulation capabilities on a video foundation model, and FLARE (Zheng et al., 2025) uses auxiliary future-latent prediction. Other approaches couple action and future-observation generation more explicitly. UWM (Zhu et al., 2025) and Motus (Bi et al., 2025) support flexible conditioning and prediction modes; Cosmos Policy (Kim et al., 2026) jointly models actions, future frames, and values; and DUST (Won et al., 2025) and DeVA (Zhang et al., 2026) use modality-specialized components. DreamZero (Ye et al., 2026b), LingBot-VA (Li et al., 2026), and Riemann-1.0 (Sun et al., 2026) adopt causal world–action generation; the first two also reuse interaction history through KV caching during closed-loop inference. GigaWorld-Policy (Ye et al., 2026a) instead uses an action-first causal mask that makes future-video generation optional when decoding actions.

2.2 Reinforcement Learning for Robot Policy Improvement

Recent work applies reinforcement learning to post-train robot foundation policies beyond supervised imitation. ConRFT (Chen et al., 2025b), CO-RFT (Huang et al., 2025), and ARFM (Zhang et al., 2025) study offline or hybrid VLA fine-tuning, while RIPT-VLA (Tan et al., 2025), SimpleVLA-RL (Li et al., 2025a), and π_{RL} (Chen et al., 2025a) learn from online interaction. Beyond VLA post-training, Batch Online RL (Dong et al., 2025) studies task-specific diffusion policies under an iterative collect-then-update protocol. WAM-RL (Qian et al., 2026) extends RL post-training to WAMs by updating the video model on successful rollouts and optimizing a separate actor with reconstruction-based rewards. Our method follows the batch-online interaction protocol but directly updates the action policy within a pretrained WAM using observed task returns and its native value predictions, without training a separate Q-function or performing test-time action reranking.

2.3 Self-Improving Robot Policies

Self-improving robot policies alternate deployment, experience collection, and subsequent training. RoboCat (Bousmalis et al., 2023) and SOAR (Zhou et al., 2024) train on self-collected environment trajectories through imitation and hindsight relabeling, but repeated success-filtered behavior cloning can saturate or degrade (Mirchandani et al., 2024). SIME (Jin et al., 2025b) and SOE (Jin et al., 2025a) address this bottleneck through modal-level or on-manifold exploration and selective reuse of self-collected successes; SIME further studies segment-level value-weighted regression. These methods improve conventional imitation policies rather than unified world action models.

Recent systems incorporate value functions or world models into the improvement loop. Q-Planning (Giridhar et al., 2026) learns a separate Q-function from successful and failed environment rollouts over multiple rounds, but keeps the base policy frozen and requires value-guided action selection at inference. VLAW (Guo et al., 2026) alternates real-world collection with separate world-model and VLA updates, using an external reward model to filter real and imagined trajectories. RISE (Yang et al., 2026) optimizes policies from imagined futures scored by a compositional value model. In contrast, we repeatedly redeploy the updated WAM, compute advantages from observed task returns and its native value predictions, and directly internalize the resulting improvement into its action policy.

3 Methods

3.1 Problem Formulation

We consider language-conditioned robotic manipulation, where g denotes the task instruction and $\mathcal{U}$ denotes the single-step action space. At timestep t, the robot receives visual observations o_t and proprioceptive state s_t, forming the context $c_t = (o_t, s_t, g)$. Given this context, the robot must generate an action chunk $a_t = (u_t, \ldots, u_{t+L-1})$ of length L, where $u_t \in \mathcal{U}$.

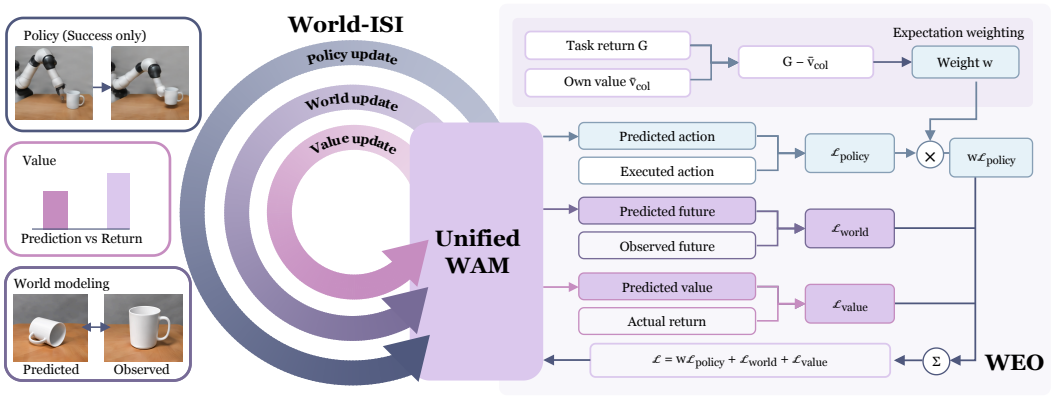

Figure 2: **World-ISI with World-Expectation Optimization (WEO). Left:** Execution experience jointly improves action generation, world prediction, and value estimation within a single WAM. **Right:** To facilitate World-ISI, WEO compares observed task returns with the model's collection-time value predictions to weight the action-denoising loss on successful executions, emphasizing successes that exceed its expectations. Failed executions supply the dedicated world- and value-prediction modes, while successful policy samples also retain auxiliary world and value losses. Failed actions are excluded from imitation. The weighted action loss and unweighted world and value losses jointly update shared model parameters.

The world model predicts future visual and proprioceptive observations given the current context and an action chunk. The value function estimates expected returns and is supervised by observed Monte Carlo returns $G_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$, where r_k is the task reward and γ is the discount factor.

Our unified WAM, parameterized by θ, shares parameters across action generation, world prediction, and value estimation (Kim et al., 2026). Its value predictor conditions on the current context, action chunk, and future observations. Appendices A.4 and A.6 detail the prediction modes and value targets.

3.2 World-ISI: Iterative Self-Improvement from Execution

World-ISI enables a unified WAM to evolve through repeated execution and joint model updates. Starting from a pretrained model θ_0 and an initial experience dataset $\mathcal{D}_0$, each iteration comprises two stages: experience accumulation and joint model update.

Experience accumulation At iteration k, the current model θ_k interacts with the environment to collect trajectories $\mathcal{B}_k$, including executed actions, observed state transitions, task outcomes, and the model's own value predictions. At collection time, the model first proposes an action chunk, predicts its future observations, and then estimates the value conditioned on this imagined future. Appendix A.3 specifies how actions are executed and training windows are reconstructed. The retained experience $\widetilde{\mathcal{B}}_k$ is added to the historical pools as $\mathcal{D}_{k+1} = \mathcal{D}_k \cup \widetilde{\mathcal{B}}_k$.

Joint model update The model is then jointly updated on $\mathcal{D}_{k+1}$ to obtain θ_{k+1}. We use successful trajectories to train action generation, and failed trajectories are used to train world prediction and value estimation. The updated model then collects experience for the next iteration, allowing learning to build on both past and newly collected experience.

3.3 WEO: World-Expectation Optimization

To enable the joint model update in World-ISI, we introduce World-Expectation Optimization (WEO) to exploit the model's own value expectations when learning from accumulated experience.

Learning actions relative to model expectations. WEO gives more action-learning weight to successful executions that exceed the model's expectations. Before execution, we convert the model's value prediction $\hat{v}_t$ to $\bar{v}_t = \mathrm{clip}((\hat{v}_t + 1)/2, 0, 1)$ and store it as the expected task return. After execution, we compute the observed Monte Carlo return G_t and, for each successful sample, use its difference from the stored prediction to determine the action-learning weight:

$$w_t = \min\left(\exp\left(\frac{\mathrm{clip}(G_t - \bar{v}_t, -2, 2)}{\beta}\right), w_{\max}\right). \tag{1}$$

Here, β controls the weighting strength, and $w_{\max}$ caps the weight. Successful actions with lower predicted returns therefore receive larger weights. We multiply each sample's action-denoising loss by w_t, yielding $w_t \ell_{\mathrm{act}}$.

Learning world and value predictions in execution. We also use actual executions to train world prediction and value estimation. World prediction is supervised by the future visual and proprioceptive observations recorded during execution. Value prediction conditions on these observed futures and uses the corresponding observed returns as supervision. Both objectives therefore ground the model's predictions in actual execution outcomes.

Joint world expectation optimization. We construct the three losses using the shared diffusion denoiser. Let z_{act}, z_{world}, and z_{value} denote the clean latent representations of executed action chunks, observed future images and proprioception, and value targets, respectively. We add Gaussian noise to the prediction targets and train the model to reconstruct their clean representations under the conditioning of each training mode. For each target type,

$$\ell_j = \lambda(\sigma)\,\mathrm{MSE}(\hat{z}_j, z_j), \qquad j \in \{\mathrm{act}, \mathrm{world}, \mathrm{value}\}, \tag{2}$$

where $\hat{z}_j$ is the denoised prediction and $\lambda(\sigma)$ is the base model's loss weighting at noise level σ. We jointly optimize

$$\mathcal{L}_{\mathrm{WEO}} = \mathbb{E}\left[m_{\mathrm{act}} w_t \ell_{\mathrm{act}} + m_{\mathrm{world}} \ell_{\mathrm{world}} + m_{\mathrm{value}} \ell_{\mathrm{value}}\right]. \tag{3}$$

The expectation is over sampled experience, noise levels, and diffusion noise. The masks m_{act}, m_{world}, and m_{value} select the active prediction targets: successful policy samples supervise all three targets, failed world-model samples supervise future observations and values, and failed value-only samples supervise values alone. The expectation weight w_t multiplies only the action loss. Appendix A.5 specifies the weighting hyperparameters and sampling probabilities; Appendix A.6 describes how stored predictions are assigned and reused.

All objectives update the same model parameters θ. The resulting model returns to interaction with updated action, world, and value predictions, supplying new experience and new collection-time expectations for subsequent iterations.

4 Experiments

We evaluate World-ISI in simulated and real-world settings. We compare our policy performance with prior methods, track improvements across self-improvement iterations, and assess action-conditioned future-prediction quality. We further evaluate real-world manipulation on two WidowX tasks and one ALOHA task. We also conduct ablations to examine the contributions of world and value supervision, the WEO objective, and iterative training.

4.1 Experimental Setup

RoboCasa simulation benchmark We evaluate on the 24-task RoboCasa kitchen manipulation benchmark (Nasiriany et al., 2024), using a single Franka Emika Panda arm. The tasks cover object pick-and-place, opening and closing doors or drawers, and other kitchen interactions. Evaluation spans five scenes per task with varied layouts and visual styles, using unseen object instances; two scenes additionally contain styles absent from the training data. Starting from the pretrained Cosmos Policy checkpoint, we perform iterative experience collection and joint model updates. We report our policy success rates averaged across all 24 tasks and three evaluation seeds, following Appendix A.2.

Table 1: Task-category comparison on RoboCasa. We report success rates (%) over eight pick-and-place (PnP), six open-or-close (OoC), and ten other tasks. Results from prior work use their respective evaluation protocols and are not controlled comparisons against our reproduced baseline. A dash indicates unavailable category-level results.

Method	PnP	OoC	Others	Average
GR00T-N1 (Bjorck et al., 2025)	22.5	68.3	60.0	49.6
UVA (Li et al., 2025b)	20.8	68.0	62.6	50.0
DP-VLA (Han et al., 2024)	31.8	78.7	64.8	57.3
GR00T-N1 + DreamGen (Jang et al., 2025)	39.9	70.8	63.8	57.6
GR00T-N1.5 + DUST (Won et al., 2025)	42.3	80.7	58.1	58.5
UWM (Zhu et al., 2025)	34.9	79.1	70.5	60.8
π_0 (Black et al., 2024)	41.8	82.3	67.2	62.5
GR00T-N1.5 (NVIDIA GEAR Team, 2025)	53.8	85.7	59.4	64.1
Video Policy (Liang et al., 2025)	47.2	83.0	70.0	66.0
FLARE (Zheng et al., 2025)	–	–	–	66.4
GR00T-N1.5 + HAMLET (Koo et al., 2025)	48.5	87.7	68.0	66.4
Cosmos Policy (Kim et al., 2026)	51.1	90.9	64.7	66.7
+ World2Act (Vuong et al., 2026)	–	–	–	66.3
+ Consistency-Consensus (Ruan et al., 2026)	–	–	–	67.3
+ GeoBoN ($N = 8$) (Zhao et al., 2026)	49.0	90.8	**70.4**	68.4
+ Gated GeoBoN ($N_{\max} = 8$) (Zhao et al., 2026)	–	–	–	67.9
+ World-ISI (Ours)	**53.9**	**93.6**	69.0	**70.1**

Experience Collection We initialize the RoboCasa experience pool with 150 successful trajectories per task and retain failed rollouts for world and value prediction. Each subsequent round targets 20 additional successful trajectories per task, with at most 100 attempts. Both successful and failed trajectories accumulate across rounds. Appendices A.1 and A.3 provide the collection and execution details.

Real-world WidowX robot task Our platform consists of a single WidowX 250S robot arm with six arm joints and a parallel-jaw gripper, equipped with two RGB cameras: one front-facing and one wrist-mounted. The policy receives two 224×224 images, a task description, and a 7-dimensional proprioceptive state comprising six joint angles and the normalized gripper position. Each action specifies six absolute joint-angle targets and a normalized gripper command. We use a nominal control frequency of 5 Hz. Appendix A.3 specifies the action horizon, execution protocol, and training-window construction. Our task suite comprises two multi-object manipulation tasks: (1) Put hotpot ingredient in plate, which requires grasping three ingredients from fixed locations; and (2) Put away bricks, which requires grasping bricks from varying locations, testing adaptation to spatial variation. Appendix A.8 provides the collection, training, and evaluation details.

Real-world ALOHA task We use a bimanual ALOHA platform. The policy receives three 224×224 RGB images, a task instruction, and a 14-dimensional proprioceptive state comprising six joint angles and one gripper position per arm. Each action specifies 14-dimensional joint-position and gripper commands, executed at a control frequency of 25 Hz. The model predicts and executes 50-step action chunks before querying again. The *Vegetable transfer* task comprises three stages: placing the first vegetable into a plate, placing the second vegetable into the same plate, and moving the plate to the center of the table. The left arm transfers the vegetables, and the right arm moves the plate. We evaluate cucumber–eggplant and carrot–eggplant combinations using a task-completion score that credits partial completion of the three stages. Appendix A.9 gives the evaluation protocol.

4.2 Main Results

Overall performance. Table 1 compares World-ISI with prior methods across three RoboCasa task categories. At training step 6,000, World-ISI achieves an average success rate of **70.1%**, improving our reproduced Cosmos Policy baseline by **3.4 percentage points** (66.7% to 70.1%). Improvements across all three categories show that the benefit of learning from accumulated experience extends to different manipulation skills.

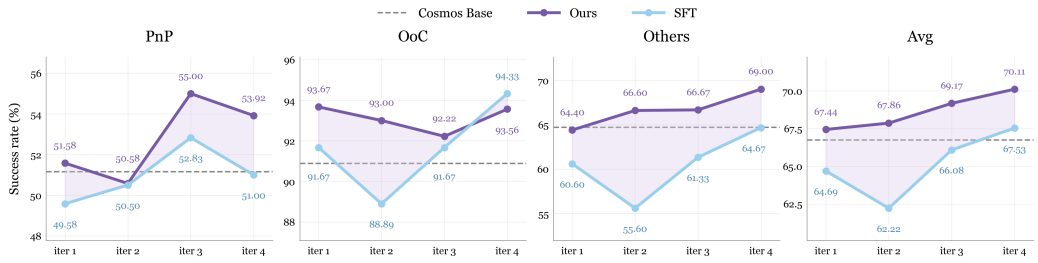

Figure 3: World-ISI performance across four self-improvement iterations on RoboCasa, compared with SFT at matched training steps. Results are grouped into pick-and-place (PnP), door/drawer opening and closing (OoC), other tasks, and all tasks. Success rates are averaged over tasks within each group.

Comparison with other Cosmos Policy extensions. World2Act (Vuong et al., 2026) adds a residual action module trained to align action latents with a separate frozen video world model. Consistency-Consensus (Ruan et al., 2026) selects actions whose predicted futures best agree with the mean of sampled futures. GeoBoN (Zhao et al., 2026) ranks candidates by cross-view depth consistency; Gated GeoBoN invokes this selection only when predicted actions and future motion disagree. These methods retain the pretrained policy backbone. World-ISI jointly updates action, world, and value prediction from accumulated environment interactions. This feedback may help correct both control and prediction errors, contributing to its higher reported success rate (70.1% versus 66.3–68.4%).

Performance across task categories. World-ISI improves success rates across all three task categories, with the largest gain on Others (64.7% to 69.0%, +4.3 points), followed by PnP (51.1% to 53.9%, +2.8 points) and OoC (90.9% to 93.6%, +2.7 points). Others includes microwave and stove switching, sink faucet control and spout adjustment, and three coffee tasks: setting up a mug, pressing the coffee-machine button, and serving the mug. A possible explanation is that these varied device operations and object–device interactions offer more opportunities to learn behaviors that are poorly covered by the initial demonstrations. WEO gives greater weight to unexpectedly successful executions, while world and value supervision calibrates predictions using observed state changes and task outcomes. Together, these updates may help turn newly collected experience into more reliable execution across this diverse group of tasks.

4.3 Iterative Self-Improvement

We evaluate World-ISI over four iterations against a supervised fine-tuning (SFT) baseline initialized from the same pretrained checkpoint and trained on the same initial dataset. SFT retains the same action, world, and value objectives and sampling settings. Its data remain fixed, while World-ISI collects and accumulates additional experience between updates. We compare both methods at matched cumulative training steps, with different interaction budgets, detailed in Appendix A.1.

Figure 3 shows that World-ISI outperforms both the pretrained Cosmos Policy and SFT in overall success rate at every iteration, reaching 70.1% at iteration 4 versus 67.5% for SFT. SFT shows larger fluctuations across checkpoints. World-ISI's overall success rate increases at each iteration, reaching its highest value at the final checkpoint. At that iteration, SFT performs better on OoC, while World-ISI performs better on PnP and Others. These results support the benefit of learning from accumulated interaction experience through joint model updates.

4.4 World Modeling Evaluation

Evaluation Protocol and Metrics We evaluate future prediction on all 24 RoboCasa tasks with 50 rollouts per task from matched initial settings. Each checkpoint controls its own trajectories, and predicted images are compared with observations after executing each full 32-step action chunk. We report three complementary metrics: peak signal-to-noise ratio (PSNR) (Wang & Bovik, 2009) measures pixel fidelity through a logarithmic transformation of mean squared error; structural similarity (SSIM) (Wang et al., 2004) compares local luminance, contrast, and structure; and learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018) measures perceptual distance using

Table 2: Action-conditioned future-prediction quality before and after World-ISI training. ↑/↓ indicates higher/lower is better. Bold denotes the better result for each metric and view.

	Cosmos Policy (Base)			World-ISI (Ours)		
View	**PSNR (dB) ↑**	**SSIM ↑**	**LPIPS ↓**	**PSNR (dB) ↑**	**SSIM ↑**	**LPIPS ↓**
Agentview Left	21.36	0.843	0.084	**23.93**	**0.874**	**0.066**
Agentview Right	21.11	0.843	0.083	**23.64**	**0.872**	**0.065**
Eye-in-Hand	15.73	0.630	0.399	**18.14**	**0.688**	**0.291**
Overall	19.40	0.772	0.189	**21.90**	**0.812**	**0.141**

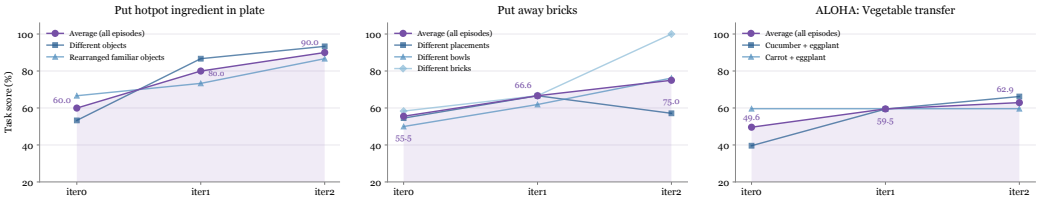

Figure 4: Real-world task-completion scores on WidowX (left and middle) and ALOHA (right). The x-axes show self-improvement iterations.

deep visual features. Higher PSNR and SSIM and lower LPIPS indicate better predictions. Overall scores average all valid image pairs across tasks, rollouts, and views; per-view scores average pairs from the corresponding camera. Appendix A.7 provides the evaluation seed, implementation details, and sample counts.

Improvement of World Modeling Table 2 shows improvements in overall PSNR (19.40 to 21.90 dB), SSIM (0.772 to 0.812), and LPIPS (0.189 to 0.141), indicating better pixel fidelity, local structural similarity, and perceptual similarity on the evaluated rollouts. Repeated environment interaction supplies observations of the states reached by the policy, allowing world-model updates to learn the actual consequences of its actions. The observed gains are consistent with this calibration through execution feedback. All three views improve in PSNR, SSIM, and LPIPS. The left camera shows the largest PSNR increase (21.36 to 23.93 dB), while the wrist camera shows the largest SSIM increase (0.630 to 0.688) and LPIPS reduction (0.399 to 0.291). The wrist view has the largest initial perceptual error, leaving more room for improvement. Differences in visibility, occlusion, and the image area occupied by moving objects may also affect how much each view benefits from additional interaction data.

4.5 Real-World Experiments

WidowX results Figure 4 shows that the average object-completion score improves over two self-improvement iterations on both tasks, from 60.0 to 90.0 for *Put hotpot ingredient in plate* and from 55.5 to 75.0 for *Put away bricks*. Both hotpot test settings improve. For bricks, performance improves with different bricks and different bowls, reaching 100.0 in the former setting, while performance with different placements rises initially and then falls back close to its initial level. Additional interaction experience may improve grasping and placement behaviors that transfer across objects and containers. However, its benefits may transfer less consistently to unfamiliar object positions, which can require different reaching paths and arm configurations.

ALOHA results Figure 4 shows that the average task-completion score for *Vegetable transfer* increases from 49.6 before self-improvement to 59.5 after the first iteration and 62.9 after the second. The gain comes from the cucumber–eggplant setting, which improves from 39.6 to 66.2, while the carrot–eggplant setting remains at 59.6. These results suggest that additional execution experience may help refine grasping and placement behaviors for combinations that were initially less reliable.

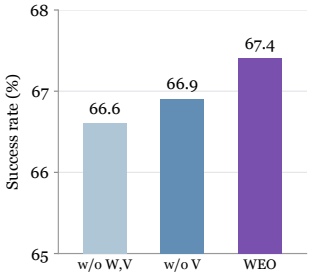

Figure 5: World (W) and value (V) loss ablations on RoboCasa at 3,000 steps.

Variant	Iter. 1 3k	Iter. 2 4k	Iter. 3 5k	Iter. 4 6k
WEO vs. SFT				
SFT	64.7	62.2	66.1	67.5
WEO	**67.4**	**67.8**	**68.9**	**69.0**
Non-iterative WEO vs. World-ISI				
WEO (Non-iterative)	67.4	67.8	68.9	69.0
World-ISI (Iterative + WEO)	67.4	**67.9**	**69.2**	**70.1**

Table 3: Objective and iteration ablations on RoboCasa. Values are mean success rates (%); 3k–6k denote cumulative training steps. Bold marks the best score within each comparison.

4.6 Ablation Studies

How do world and value modeling contribute to policy improvement? We compare full WEO with variants that remove the value loss or both the world and value losses, while retaining expectation-weighted action learning. All variants use matched training-step and batch-size budgets, with sampling adjusted to exclude modes without active objectives; Appendix A.5 details the corresponding loss and sampling configurations. We further examine the data sources for prediction supervision in Appendix B.1. Figure 5 shows that success increases from 66.6% without either prediction loss to 66.9% with world supervision and 67.4% with both losses. These results suggest that both prediction objectives contribute to policy improvement under the evaluated configurations. World supervision encourages the shared model to capture the consequences of executed actions, while value supervision connects these consequences to observed task outcomes. These objectives also allow failed executions to contribute to learning without imitating their actions. Together, they may help the model learn representations useful for action generation beyond successful-action imitation alone.

Is WEO effective over imitation learning? To examine whether WEO provides benefits beyond additional experience collection, we compare it with SFT on the same fixed training dataset. Both methods use identical action/world/value losses, sampling probabilities, and optimization settings, and neither collects additional rollouts. The only difference is the action-learning weight: SFT uses unit weights, while WEO uses expectation-based weights. As shown in Table 3, WEO outperforms SFT at all four training checkpoints, reaching 69.0% success after 6,000 steps compared with 67.5% for SFT. These gains suggest that WEO can improve learning from existing experience without expanding the dataset. By emphasizing successful actions whose outcomes exceeded the model's collection-time expectations, WEO may make better use of the available successes.

Does continued interaction with the environment matter? We compare World-ISI, which repeatedly interacts with the environment between model updates, with non-iterative WEO, which continues training without further interaction, at matched cumulative training steps. As shown in Table 3, both variants achieve 67.4% success at 3,000 steps. World-ISI is slightly ahead at 4,000 steps (67.9% versus 67.8%) and achieves higher success rates at 5,000 and 6,000 steps, reaching 70.1% versus 69.0% at the final checkpoint. Without further interaction, the model may not observe the actual consequences of its updated actions or identify prediction errors on newly encountered states. World-ISI repeatedly returns the updated model to the environment, allowing subsequent updates to incorporate feedback on its current behavior. This ongoing interaction–update loop may explain its stronger performance at the later checkpoints.

5 Conclusion

We present World-ISI, which iteratively improves a unified world action model through environment interaction. WEO weights action learning by the difference between observed returns and predicted values while jointly updating world and value predictions. Applied to Cosmos Policy, World-ISI raises average RoboCasa success from 66.7% to 70.1% over four iterations and improves future prediction. Ablations and experiments on WidowX and ALOHA further support its effectiveness.

AI USE STATEMENT

In this work, we used generative AI tools to help assist with translation, create and modify scientific figures or images, suggest experimental parameters, summarize existing literature and edit our research paper to improve readability. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. We have reviewed all AI-assisted work. Specifically, we checked AI-assisted image modifications, translations, and language edits for accuracy and consistency with our intended meaning. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix A provides details of data collection, training objectives, sampling configurations, hyperparameters, and evaluation protocols for our RoboCasa and WidowX experiments. Section 4.1 and Appendix A.9 specify the ALOHA setup and evaluation protocol. Appendix B describes additional ablation settings. We distinguish fixed-data and iterative training protocols and report the evaluation metrics and aggregation procedures.

ETHICS STATEMENT

This work studies robot learning in simulation and controlled laboratory settings. Improved task performance does not guarantee safe behavior, particularly under distribution shifts. Deployment beyond the evaluated settings requires further safety validation and appropriate human oversight.

REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

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.

Konstantinos Bousmalis, Giulia Vezzani, Dushyant Rao, Coline Devin, et al. Robocat: A self-improving generalist agent for robotic manipulation. *arXiv preprint arXiv:2306.11706*, 2023.

Kang Chen, Zhihao Liu, Tonghe Zhang, Zhen Guo, Si Xu, Hao Lin, Hongzhi Zang, Quanlu Zhang, Zhaofei Yu, Guoliang Fan, et al. πrl: Online rl fine-tuning for flow-based vision-language-action models. *arXiv preprint arXiv:2510.25889*, 8, 2025a.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. ConRFT: A reinforced fine-tuning method for VLA models via consistency policy, 2025b. URL https://arxiv.org/abs/2502.05450.

Perry Dong, Suvir Mirchandani, Dorsa Sadigh, and Chelsea Finn. What matters for batch online reinforcement learning in robotics?, 2025. URL https://arxiv.org/abs/2505.08078.

Varun Giridhar, Anant Khandelwal, Jeremy A. Collins, Ignat Georgiev, and Animesh Garg. Beyond imitation: Self-improving robot policies via off-policy q-planning. *arXiv preprint arXiv:2608.21204*, 2026.

Yanjiang Guo, Tony Lee, Lucy Xiaoyang Shi, Jianyu Chen, Percy Liang, and Chelsea Finn. VLAW: Iterative co-improvement of vision-language-action policy and world model, 2026. URL https://arxiv.org/abs/2602.12063.

ByungOk Han, Jaehong Kim, and Jinhyeok Jang. A dual process vla: Efficient robotic manipulation leveraging vlm, 2024. URL https://arxiv.org/abs/2410.15549.

Dongchi Huang, Zhirui Fang, Tianle Zhang, Yihang Li, Lin Zhao, and Chunhe Xia. CO-RFT: Efficient fine-tuning of vision-language-action models through chunked offline reinforcement learning, 2025. URL https://arxiv.org/abs/2508.02219.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Yang Jin, Jun Lv, Han Xue, Wendi Chen, Chuan Wen, and Cewu Lu. SOE: Sample-efficient robot policy self-improvement via on-manifold exploration, 2025a. URL https://arxiv.org/abs/2509.19292.

Yang Jin, Jun Lv, Wenye Yu, Hongjie Fang, Yong-Lu Li, and Cewu Lu. SIME: Enhancing policy self-improvement with modal-level exploration, 2025b. URL https://arxiv.org/abs/2505.01396.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Myungkyu Koo, Daewon Choi, Taeyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. Hamlet: Switch your vision-language-action model into a history-aware policy, 2025. URL https://arxiv.org/abs/2510.00695.

Haozhan Li, Yuxin Zuo, Jiale Yu, et al. 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.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025b. URL https://arxiv.org/abs/2503.00200.

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.

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. *arXiv preprint arXiv:2508.05635*, 2025.

Suvir Mirchandani, Suneel Belkhale, Joey Hejna, Evelyn Choi, Md Sazzad Islam, and Dorsa Sadigh. So you think you can scale up autonomous robot data collection?, 2024. URL https://arxiv.org/abs/2411.01813.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024. URL https://arxiv.org/abs/2406.02523.

NVIDIA GEAR Team. GR00T N1.5: An improved open foundation model for generalist humanoid robots. NVIDIA Research, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_5/.

Zezhong Qian, Xiaowei Chi, Yu Qi, Haozhan Li, Zhi Yang Chen, and Shanghang Zhang. WAM-RL: World-action model reinforcement learning with reconstruction rewards and online video SFT, 2026. URL https://arxiv.org/abs/2606.17906.

Bo-Kai Ruan, Teng-Fang Hsiao, Ling Lo, and Hong-Han Shuai. Is the future compatible? diagnosing dynamic consistency in world action models. *arXiv preprint arXiv:2605.07514*, 2026.

Haofeng Sun, Jiangbo Pei, Fei Kang, Zexiang Liu, Yaokun Li, Boyi Jiang, Hua Xue, Cindy Zhou, Wei Li, Yichen Wei, Mengyin An, Fanliang Zhao, Biao Jiang, Zile Wang, Yang Liu, and Yangguang Li. Riemann-1.0: An embodied world action model for physical ai. *arXiv preprint arXiv:2608.27033*, 2026.

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.

An Dinh Vuong, Tuan Van Vo, Abdullah Sohail, Haoran Ding, Liang Ma, Xiaodan Liang, Anqing Duan, Ivan Laptev, and Ian Reid. World2act: Latent action post-training from world model dynamics, 2026. URL https://arxiv.org/abs/2603.10422.

Zhou Wang and Alan C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009. doi: 10.1109/MSP.2008.930649.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. URL https://www.ece.uwaterloo.ca/~z70wang/publications/ssim.html.

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model. *arXiv preprint arXiv:2510.27607*, 2025.

Jiazhi Yang, Kunyang Lin, Jinwei Li, Wencong Zhang, et al. Rise: Self-improving robot policy with compositional world model. *arXiv preprint arXiv:2602.11075*, 2026.

Angen Ye, Boyuan Wang, Chaojun Ni, et al. GigaWorld-Policy: An efficient action-centered world–action model, 2026a. URL https://arxiv.org/abs/2603.17240.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b.

Hongyin Zhang, Shiyuan Zhang, Junxi Jin, Qixin Zeng, Yifan Qiao, Hongchao Lu, and Donglin Wang. Balancing signal and variance: Adaptive offline RL post-training for VLA flow models, 2025. URL https://arxiv.org/abs/2509.04063.

Mengqi Zhang, Sahil Khose, Simar Kareer, Yuchen Song, Unnat Jain, and Judy Hoffman. Deva: Decoupled video-action model with physical guidance for robot policy learning. *arXiv preprint arXiv:2607.24159*, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. URL https://richzhang.github.io/PerceptualSimilarity/.

Zesen Zhao, Minkyoung Cho, Hui Shen, Boyuan Zheng, Kunxiao Gao, Yulong Cao, and Z. Morley Mao. Test-time scaling for world action models via zero-shot geometric evaluation, 2026. URL https://arxiv.org/abs/2607.17454.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Zhiyuan Zhou, Pranav Atreya, Abraham Lee, Homer Walke, Oier Mees, and Sergey Levine. Autonomous improvement of instruction following skills via foundation models, 2024. URL https://arxiv.org/abs/2407.20635.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

A Additional Experimental Details

A.1 RoboCasa Data and Optimization Protocols

Collection details. Initial collection covers the settings represented in the 50 original demonstrations per task. When a setting has fewer than two successful collected trajectories, we supplement it with its original demonstration. We collect trajectories in five kitchen scenes with predefined layout–style combinations, using movable objects from the training object-instance split. After initial coverage, collection prioritizes object categories and instances not covered by previous successful trajectories. The initial pool contains 150 successful collected trajectories per task, with failed rollouts retained for dedicated prediction training. Each subsequent collection round targets 20 additional successful trajectories per task, with at most 100 attempts; both successful and failed rollouts accumulate across rounds.

Optimization protocol. Table 4 summarizes the fixed-data and iterative protocols. SFT and non-iterative WEO use the same fixed initial pools, while World-ISI adds experience between updates. All three methods start from the same pretrained checkpoint and are evaluated at cumulative training steps 3,000, 4,000, 5,000, and 6,000. The first iteration comprises 3,000 optimization steps, followed by 1,000 steps per iteration. SFT uses unit action weights with the same action, world, and value losses, sampling probabilities, and optimization hyperparameters as WEO.

Table 4: RoboCasa training protocols. Initial data are counted as successful trajectories per task; failed rollouts and demonstration supplementation are described in the text. SFT retains auxiliary world and value supervision.

Method	Initial successes/task	New collection	Training objective
SFT	150	None	Imitation learning
WEO	150	None	WEO
World-ISI	150	Each round	WEO

A.2 RoboCasa Policy Evaluation

Our RoboCasa policy evaluations use seeds 195, 196, and 197. We compute each task's success rate for each seed and report the arithmetic mean across the three seeds and all 24 tasks. Category-level results average over the tasks in the corresponding category. Results quoted from prior work retain their original evaluation protocols. World-prediction metrics use the separate protocol in Appendix A.7.

A.3 Action Prediction, Execution, and Training Windows

During RoboCasa and WidowX experience collection, the robot predicts an action chunk of length L and executes its first $L/2$ actions before observing the environment and querying the model again. Training chunks are reconstructed from recorded executed actions. At a query boundary, a full-length target concatenates the executed halves of two consecutive predictions, and its future-observation target corresponds to the end of this executed-action window.

RoboCasa uses $L = 32$ and executes 16 actions per query. During RoboCasa experience collection, action, future-state, and value generation each use five denoising steps per query. WidowX uses $L = 16$ and executes eight actions per query. The world-prediction evaluation in Appendix A.7 instead executes full 32-step chunks and uses a separate denoising budget.

A.4 Joint Prediction Modes

Let T denote the terminal timestep and $t^+ = \min(t + L, T)$ the future-observation timestep associated with a training action chunk. Different conditioning masks define three prediction modes of

the unified WAM:

$$\begin{aligned} \text{Policy:} \quad & p_\theta(a_t, o_{t^+}, s_{t^+}, v_t \mid c_t), \\ \text{World model:} \quad & p_\theta(o_{t^+}, s_{t^+}, v_t \mid c_t, a_t), \\ \text{Value prediction:} \quad & p_\theta(v_t \mid c_t, a_t, o_{t^+}, s_{t^+}). \end{aligned} \tag{4}$$

All three modes share model parameters and are trained through denoising losses on their respective prediction targets. The value target v_t is defined in Appendix A.6.

A.5 Weighting Hyperparameters and Sample Allocation

Table 5 lists the RoboCasa training hyperparameters. The masked denoising objective retains the base model's noise-level weighting; expectation weights multiply only action-denoising terms and are not normalized within a batch. World and value losses receive no expectation weighting. Successful policy samples retain their auxiliary future and value targets. Dedicated world and value samples are drawn exclusively from failed trajectories.

Table 5: RoboCasa weighting and sampling hyperparameters. Sampling ratios refer to policy, world-model, and value-only modes, rather than separate loss coefficients.

Hyperparameter	Value
Weighting temperature β	0.5
Maximum action weight $w_{\max}$	10
Return discount γ	1.0
Return–expectation difference clipping	$[-2, 2]$
Additional fixed action-loss multiplier	1
World/value expectation-weight multipliers	1 / 1
Policy/world/value sampling: WEO and SFT	80% / 10% / 10%
Policy/world/value sampling: no value loss	80% / 20% / 0%
Policy/world/value sampling: no world or value loss	100% / 0% / 0%
Stored-value query interval (executed steps)	16
Learning rate	5×10^{-5}

Sampling adjustments for prediction-loss ablations. All variants retain expectation-weighted action learning. Removing the value-prediction loss masks value targets in all training modes and excludes value-only samples from the batch. Removing both prediction losses additionally excludes world-model samples, leaving only successful policy samples. These adjustments prevent samples without an active objective from consuming batch capacity; the resulting sampling proportions are listed in Table 5. Training-step and batch-size budgets are matched, but the allocation of samples across the remaining modes changes. These comparisons therefore measure the effects of the corresponding loss and sampling configurations.

A.6 Return Scales and Collection-Time Values

Value prediction is supervised by the observed future return, normalized as $v_t = 2G_{t^+} - 1$. For binary terminal rewards and $\gamma = 1$, observed returns are in $\{0, 1\}$ and value targets are in $\{-1, 1\}$. The targets are grounded in observed Monte Carlo returns, rather than stored predictions or temporal-difference bootstrapping.

The collection-time prediction $\bar{v}_t$ is stored before execution and retained with the trajectory. It conditions on the current context, original action proposal, and model-predicted future observations. Each stored prediction is assigned to the actions executed before the next query and retained unchanged throughout training, including when historical experience is reused. In RoboCasa, the executed 16-step prefix shares the query's stored value. This value evaluates the original proposal and its imagined future; it is not recomputed for the action window later reconstructed for training as described in Appendix A.3. Samples without stored value predictions receive unit action weight.

The implementation subtracts the stored value from the normalized Monte Carlo target when computing action weights. Under the success-only action objective and $\gamma = 1$, the normalized target is $+1$, identical to the successful unnormalized return. Thus, the implemented action-weight signal is exactly $1 - \bar{v}_t$, as in the main-text objective.

A.7 World-Prediction Evaluation

The world-prediction metrics in Table 2 use seed 195, with 50 rollouts per task from matched initial settings and each checkpoint controlling its own trajectories. Full 32-step chunks align predicted and observed futures. Action/world generation uses 5/50 denoising steps, and LPIPS uses AlexNet. Each valid camera-image pair has equal weight in the overall mean; per-view means restrict the same calculation to one camera. The base checkpoint contributes 12,679 valid pairs per camera (38,037 in total), and World-ISI contributes 12,540 per camera (37,620 in total). Rollouts with more valid pairs contribute more to the aggregate, and checkpoint-specific trajectories can differ in length despite matched initial settings.

A.8 WidowX Initialization, Training, and Evaluation

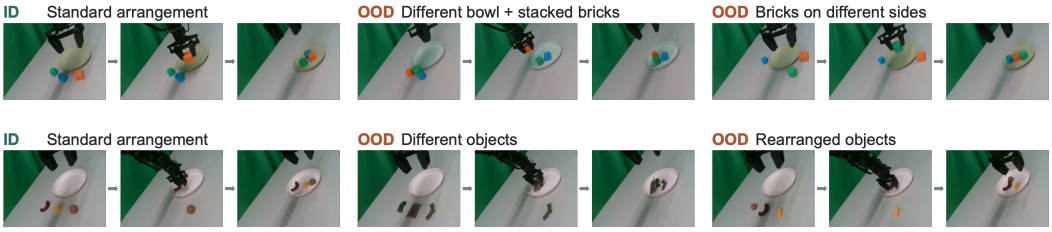

Figure 6: Task examples for WidowX real-world experiments. In-distribution (ID) training configurations and out-of-distribution (OOD) test configurations. ID examples follow the standard object arrangements seen during training, while OOD examples introduce changes in object identity, spatial arrangement, object side placement, and container configuration to evaluate generalization beyond the training distribution.

Policy initialization and iterative updates. For Iteration 0, we manually collect 100 demonstration trajectories per task under the standard in-distribution (ID) configurations shown in Figure 6. We initialize Cosmos Policy from Cosmos-Predict, which is pretrained without action conditioning, and train it on these demonstrations until the training loss decreases to 0.01. In each subsequent iteration, we collect rollouts until 10 successful trajectories are obtained per task, retaining both successful and failed rollouts, and update the model for 5,000 optimization steps on the accumulated demonstrations and rollouts. All collection uses the same ID configurations; out-of-distribution (OOD) configurations are reserved for evaluation. The action-execution and training-window construction follow Appendix A.3.

Evaluation metric. We evaluate 10 trials per task and iteration, with three target objects per trial, and report the mean object-completion score on a scale from 0 to 100. This metric credits partial completion; a score of 100 requires completing all three objects in every trial. For $n_j \in \{0, 1, 2, 3\}$ completed objects in trial j, the score is

$$S = \frac{1}{10} \sum_{j=1}^{10} \frac{100 n_j}{3}. \tag{5}$$

A.9 ALOHA Evaluation

Figure 7 illustrates the standard in-distribution (ID) configuration and the two out-of-distribution (OOD) configurations used for the ALOHA task.

We evaluate *Vegetable transfer* at 16,000, 20,000, and 24,000 training steps, using 10 trials per checkpoint: five for cucumber–eggplant and five for carrot–eggplant. We report the arithmetic mean of the recorded episode-completion scores, multiplied by 100. Partial-completion scores are stored as 0.33 or 0.66, and full completion as 1.0; these stored values are retained when computing the means. The overall curve averages all 10 trials, while each object-combination curve averages its five trials. ALOHA executes the full 50-step prediction at each query, as specified in Section 4.1.



Figure 7: Task examples for ALOHA *Vegetable transfer*. **Left:** the standard ID configuration places the cucumber on the left and the eggplant on the right. **Middle:** the rearranged OOD configuration swaps their positions. **Right:** the different-vegetable OOD configuration replaces the cucumber with a carrot. Each group shows initial, intermediate, and final frames from one successful real-robot rollout. Left and right refer to the camera view.

B Additional Ablations

B.1 Should world and value training emphasize failed executions?

We compare two non-iterative WEO variants that differ in the data used for dedicated world- and value-prediction training. Both use the same successful policy-training pool, joint loss configuration, weighting hyperparameters, and optimization settings. The policy/world/value sampling probabilities remain 80%/10%/10%. In the mixed variant, successful and failed rollouts are sampled with equal probability within each dedicated prediction mode; in the default variant, these modes use failed rollouts only. Successful policy samples retain auxiliary world and value losses in both variants. This comparison therefore tests how to allocate the dedicated prediction-training budget when successful trajectories already supply auxiliary prediction supervision.

Table 6: Data sources for dedicated world and value training. We report average RoboCasa policy success rates (%) at matched training steps for non-iterative WEO. Both variants retain the same successful policy samples and their auxiliary prediction losses.

Prediction-training rollout source	3,000 steps	4,000 steps
Successful + failed (1:1)	64.2	65.8
Failed only (default WEO)	**68.2**	**67.8**

As shown in Table 6, concentrating dedicated prediction training on failed rollouts improves policy success by 4.0 and 2.0 percentage points at 3,000 and 4,000 steps, respectively. These results support the default sampling choice under the evaluated training budgets. The comparison measures downstream policy performance; it does not directly establish which variant has more accurate world or value predictions.