HaWMPO: Hallucination-Aware World Model-based Policy Optimization for Generalist Robot Policy

Anonymous authors
Paper under double-blind review

Abstract

Generalist robot policies have demonstrated strong generalization across robotic manipulation tasks, yet their success rates remain limited in complex long-horizon scenarios. Recent methods improve Visual-Language-Action (VLA) policies through online reinforcement learning on real robots, but such training relies on costly physical interactions, suffers from low sample efficiency, and may introduce hardware and safety risks. World models offer a promising alternative by enabling policy optimization with imagined rollouts. However, long-horizon rollouts generated by world models often suffer from prediction hallucinations, producing biased state transitions that can mislead policy learning. To address this issue, we propose Hallucination-aware World Model-based Policy Optimization (HaWMPO), a closed-loop reinforcement learning pipeline for VLA policy post-training with world models. Specifically, HaWMPO introduces an action-conditioned hallucination-aware model to estimate the reliability of generated image sequences, and incorporates hallucination scores into group relative policy optimization through a Reward-Soft mechanism, suppressing unreliable action chunks during training. On the LIBERO benchmark, HaWMPO achieves the best average success rate, with gains of 15.0 and 2.8 percentage points over the base model and strongest baseline, respectively. Real-world experiments on a G1 robot further validate its effectiveness, raising the average success rate on two manipulation tasks from 67.5% to 80.0%.

1 Introduction

In recent years, generalist robot policies (Kim et al., 2024; Black et al., 2024; Brohan et al., 2023; Kim et al., 2025; Wu et al., 2026; Shi et al., 2025; Brohan et al., 2022; Shukor et al., 2025; Nvidia et al., 2025; Luo et al., 2026; Xu et al., 2026; Joy Future Academy, 2026; JoyAI-RA Team, 2026; Liu et al., 2026) have demonstrated strong multi-task generalization capabilities in robotic manipulation tasks. However, their further improvement is still constrained by the high cost of interaction in real-world environments. This limitation is particularly evident in embodied manipulation tasks, where directly performing online reinforcement learning on physical robots often requires extensive physical trial and error. Such a process is not only sample-inefficient, but also introduces potential risks of hardware damage and safety issues. Therefore, a key challenge for improving general-purpose robotic policies is how to provide closed-loop feedback for reinforcement learning while reducing the reliance on real-world robot interactions. With the rapid development of world models (Wan et al., 2025; Ali et al., 2025; Zhu et al., 2024; Kim et al., 2026), transferring reinforcement learning from the real environment to an imagination space constructed by world models has gradually emerged as a promising paradigm for policy optimization. In this work, a world model can serve as an interactive virtual simulator, providing low-cost and efficient synthetic experience for policy learning. However, generative world models are not strict physical simulators. Their predictions are inherently based on generative modeling, which makes them prone to accumulated prediction errors and hallucinations during long-horizon rollouts. Directly using hallucinated virtual trajectories for reinforcement learning may mislead policy updates, causing the VLA policy to exploit world-model biases rather than learn actions effective in real environments. For example, WoVR (Jiang et al., 2026) shows that hallucinated imagined rollouts can mislead policy optimization by turning

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.

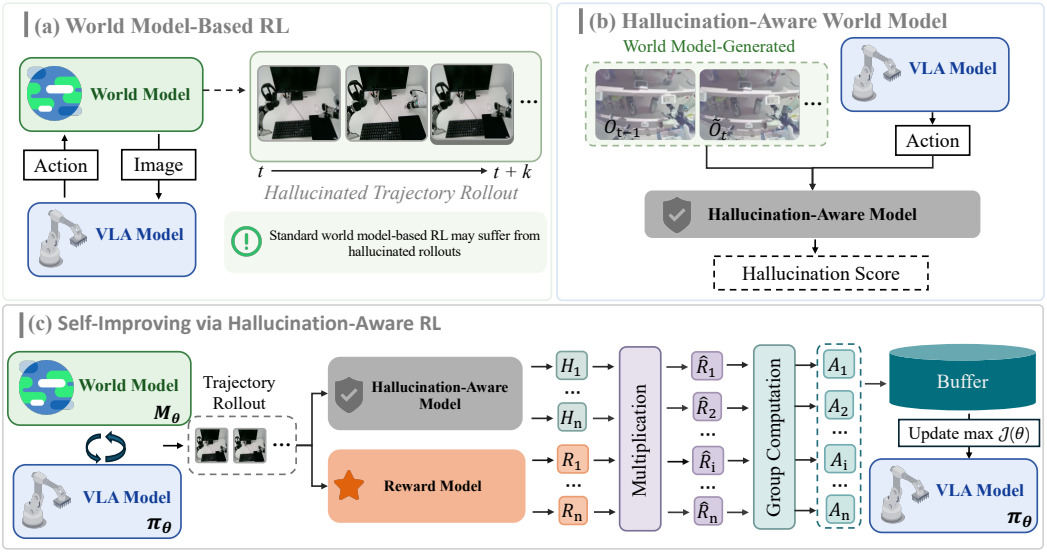


Figure 1: Illustration of the proposed hallucination-aware VLA reinforcement learning pipeline. **(a)** In standard world model-based RL, the VLA policy interacts with a learned world model to generate imagined trajectory rollouts, while hallucinated visual predictions may introduce biased reward signals and lead to unstable policy updates. **(b)** Conditioned on VLA actions, the hallucination-aware model assesses the reliability of imagined trajectory rollouts from the world model and assigns a hallucination score to each generated chunk. **(c)** Our closed-loop pipeline incorporates hallucination scores into GRPO. The multiplication applies $\mathrm{clip}_{[0,1]}(1 - \alpha H)$ to each raw action reward, as defined in Eq. (3). All reported comparisons use $\alpha = 0.3$.

world-model errors into false learning signals. Such hallucinations introduce biased state transitions, causing the policy optimization process to deviate from real-world dynamics and ultimately weakening the stability of reinforcement learning and the performance of the learned policy.

We propose a hallucination-aware world model-based reinforcement learning pipeline for VLA policy optimization. Instead of treating all world-model-generated rollouts equally, our method explicitly estimates their reliability and weights their contributions according to the predicted hallucination level. The pipeline consists of three core components: an action-conditioned world model that generates future image sequences from the current visual context and VLA-predicted action chunks, a hallucination-aware model that evaluates rollout reliability under the given action condition, and a reward model that predicts task completion from generated images. Building on this world model-based RL pipeline, we introduce a Reward-Soft mechanism that, with the fixed coefficient $\alpha = 0.3$, scales every action reward in a chunk by the same reliability weight. More hallucinated chunks therefore contribute less to the return, while no chunk is discarded solely for its score. We validate our method in the LIBERO (Liu et al., 2023) simulation benchmark and on two physical manipulation tasks, reporting task success rates. Overall, the main contributions of this work are summarized as follows:

- We establish a closed-loop VLA reinforcement learning pipeline integrating a world model, a hallucination-aware model, and a reward model. This architecture supports online-style policy optimization without frequent real-world robot interactions.
- We design an action-conditioned hallucination-aware model to quantify the reliability of image sequences predicted by the world model under specified action inputs. This model produces a continuous hallucination score that can be directly leveraged for reinforcement learning.
- We integrate the derived hallucination score into the GRPO optimization pipeline, and propose a Reward-Soft mechanism to penalize severely hallucinated action segments during rollouts. This mitigates the adverse bias of the world model on policy updates in reinforcement learning.

2 Related Work

2.1 Reinforcement Learning for VLA Models

As VLA models advance toward universal robotic policies, research increasingly adopts reinforcement learning (Chen et al., 2026; 2025b; Guo et al., 2025; Zhang et al., 2024; Yu et al., 2025; Yuan et al., 2025) to break limits of supervised imitation learning. Traditional imitation learning depends heavily on offline demonstrations, with poor performance on out-of-distribution cases, long-horizon tasks and sparse rewards. VLA-RL (Lu et al., 2025) adopts a robotic process reward model to mitigate sparse reward issues and optimizes policies via online PPO (Schulman et al., 2017) reinforcement learning. Targeting flow-matching VLA models, the πRL (Chen et al., 2025a) framework devises Flow-Noise and Flow-SDE to resolve intractable action log-likelihood calculation, enabling direct RL optimization and boosting flow-based policy capacity. LWD (Wang et al., 2026) constructs an offline-online RL framework for robot clusters. With distributional implicit value learning and Q-function adjoint matching, it addresses several limitations of pre-trained VLA models, including over-dependence on imitation learning, real-world distribution drift, sparse rewards in long-horizon tasks, underutilized failure data, and limited cross-task generalization across diverse robotic manipulation scenarios.

2.2 World Models for VLA Reinforcement Training

With the development of VLA models, world models have become an important tool for VLA training due to their ability to simulate environment distributions and capture physical dynamics. Recent works have incorporated world models into VLA pre-training, post-training, and reinforcement learning to reduce real-world interaction costs and provide virtual experience. For example, World-Env (Xiao et al., 2025) enables RL post-training without real-world interaction by combining a geometry-aware world simulator with a VLM-guided instant reflector. WMPO (Zhu et al., 2025) integrates a pixel-level video world model with online GRPO (Shao et al., 2024), improving VLA policies through policy-behavior alignment and frame-level action control. However, long-horizon autoregressive generation in world models can accumulate errors and lead to hallucinations. To mitigate this issue, WoVR (Jiang et al., 2026) introduces Keyframe Initialization and Refinement (KIR) and PACE-based policy-world model co-evolution to reduce hallucinations, error accumulation, and distorted RL signals caused by policy drift.

3 Methodology

3.1 Hallucination-Aware Model for World Model

World-model prediction errors can accumulate during imagined rollouts and introduce misleading signals into VLA policy optimization. We introduce an action-conditioned Hallucination-Aware Model (HAM) to estimate the reliability of each generated video chunk. HAM takes an eight-frame predicted chunk $\hat{\mathbf{V}}_c$, its conditioning actions $\mathbf{a}_c$, the current observation I_c, and an anchor image I_0 as inputs. Here, I_0 is the initial observation of the rollout, providing a fixed visual reference.

To construct supervision, we generate video chunks from recorded actions and compare them with the corresponding observed sequences. Four metrics capture complementary properties: depth consistency, MUSIQ image quality, optical-flow trajectory consistency, and DINOv3 similarity. After orienting and min–max normalizing each metric to $q_c^{(m)} \in [0, 1]$, where larger values indicate better quality, we form a composite quality score C_c. The hallucination target reverses its normalized direction:

$$C_c = \frac{\sum_{m=1}^{4} \omega_m q_c^{(m)}}{\sum_{m=1}^{4} \omega_m}, \qquad h_c^\star = \frac{C_{\max} - C_c}{\max(C_{\max} - C_{\min}, 10^{-8})}. \tag{1}$$

Here, ω_m are nonnegative metric weights, and $C_{\min}, C_{\max}$ are the minimum and maximum composite scores on the HAM training manifest. Larger C_c means better quality, and larger $h_c^\star$ means more severe hallucination. Eq. (1) equals one minus C_c only after C_c is min–max normalized by those bounds; it is not $1 - C_c$ on the weighted average itself, and the four weights need not be equal. Appendix A.3 states the labeling-script defaults separately from this formula. These targets

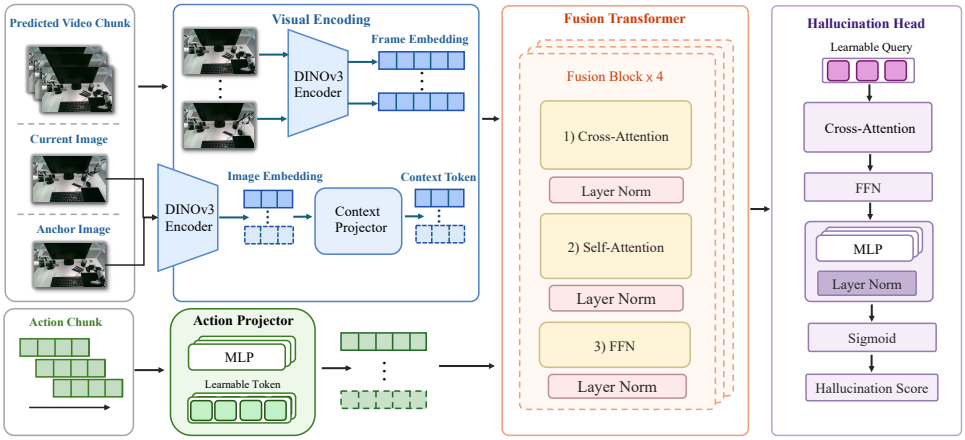


Figure 2: Architecture of HAM. Predicted video frames, the current observation, the initial anchor image, and conditioning actions are encoded and fused to predict a chunk-level hallucination score. Higher scores indicate less reliable predictions.

are proxies for physical consistency; task failure alone does not imply hallucination, and a low score does not guarantee physical validity.

As illustrated in Fig. 2, a frozen DINOv3 encoder extracts visual features, while the Context and Action Projectors map contextual images and actions into a shared feature space. A four-layer Fusion Transformer combines these inputs through action-to-visual cross-attention, self-attention, and feed-forward layers. A learnable query aggregates the fused representations, followed by an MLP and sigmoid activation to produce a continuous chunk-level score. HAM is fitted by regression to $h_c^\star$:

$$\hat{h}_c = f_\psi(\hat{\mathbf{V}}_c, \mathbf{a}_c, I_c, I_0), \qquad \mathcal{L}_{\text{HAM}} = \frac{1}{|\mathcal{D}_{\text{HAM}}|} \sum_{c \in \mathcal{D}_{\text{HAM}}} \ell_{\text{reg}}(\hat{h}_c - h_c^\star), \tag{2}$$

where $\mathcal{D}_{\text{HAM}}$ is the labeled training set. The training implementation uses Smooth L1 with unit transition, so $\ell_{\text{reg}}(e) = \frac{1}{2}e^2$ when $|e| \le 1$; this is the relevant range for predictions and normalized targets in $[0, 1]$. Observed future sequences are required only for target construction. During imagined rollouts, HAM predicts $\hat{h}_c$ from the generated chunk and its conditioning inputs for subsequent reward modulation.

3.2 Reward-Soft Mechanism for Hallucination-Aware RL

HAM predicts one hallucination score $H_c = \hat{h}_c$ for each generated chunk c; there is no additional average of frame-level HAM predictions. Reward-Soft broadcasts the same reliability weight to all action rewards within that chunk:

$$w_c = \text{clip}_{[0,1]}(1 - \alpha H_c), \qquad \widetilde{r}_{c,j} = w_c r_{c,j}, \quad j = 1, \ldots, K, \tag{3}$$

where $K = 8$ and $r_{c,j}$ is the raw reward signal associated with action j in chunk c. Thus, HAM aggregates visual and action information into one chunk score, whereas the reward sequence retains its action-level resolution. With $\alpha = 0.3$ and $H_c \in [0, 1]$, weights lie in $[0.7, 1]$: more hallucinated chunks receive smaller reward magnitudes without being hard-rejected. For nonnegative rewards this attenuates apparent success; for signed reward differences it attenuates both positive and negative signals. Reward scaling occurs before return and advantage computation. We use $\alpha = 0.3$ in all main comparisons and consider other values only in the sensitivity analysis.

3.3 Hallucination-Aware World Model-Based Policy Optimization

The action-conditioned world model acts as a frozen simulator. Each step uses a fixed reference image, the four most recent frames, and an eight-action chunk to generate eight future frames.

These update the recent visual context for the next policy decision. HAM uses the generated chunk, conditioning actions, the current image, and the fixed anchor to estimate its hallucination score; a separate reward model supplies action-level rewards. Appendix A.7 details the simulator, and Algorithm 1 summarizes the complete loop.

GRPO compares G rollouts with the same task and initial condition. Here, G is the number of rollouts per comparison group, not the total number of parallel environments. For rollout i, let S_i be the sum of its modulated rewards up to termination. The group-relative advantage is

$$\bar{S} = \frac{1}{G}\sum_{i=1}^{G} S_i, \qquad \sigma_S = \sqrt{\frac{1}{G-1}\sum_{i=1}^{G}(S_i - \bar{S})^2}, \qquad A_i = \frac{S_i - \bar{S}}{\sigma_S + 10^{-6}}. \tag{4}$$

The same rollout advantage is broadcast to its valid policy decisions. For $G > 1$, equal returns give zero advantages; the additive constant avoids division by zero. Invalid or padded decisions are masked during optimization.

Let $u_{i,t}$ be a sampled policy output token and $s_{i,t}$ its conditioning context, including the observation, task instruction, and action-position context. Let $\mathcal{T}_i$ index valid tokens in rollout i, and let $\pi_{\theta_{\text{old}}}$ be the policy that collected the rollouts. OpenVLA-OFT predicts the action-token positions in parallel. We define the probability ratio and clipped surrogate as

$$\begin{aligned} \rho_{i,t} &= \frac{\pi_\theta(u_{i,t} \mid s_{i,t})}{\pi_{\theta_{\text{old}}}(u_{i,t} \mid s_{i,t})}, \\ \ell_{i,t} &= \min\{\rho_{i,t}A_i, \text{clip}(\rho_{i,t}, 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})A_i\}. \end{aligned} \tag{5}$$

Let $m_i \in \{0, 1\}$ denote the rollout's reward-filter mask. The objective averages valid-token surrogates within each rollout and then across the group:

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau_{1:G}\sim\pi_{\theta_{\text{old}}}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{m_i}{|\mathcal{T}_i|}\sum_{t\in\mathcal{T}_i}\ell_{i,t}\right]. \tag{6}$$

Masked rollouts contribute zero. Reward-based group filtering uses unmodulated returns, as detailed in Appendix A.2; it is separate from HAM-based soft weighting. We minimize $-\mathcal{J}(\theta)$. The actor-only setup uses neither a learned value baseline nor an explicit reference-policy KL penalty. This separates the reward-model signal from the group-relative baseline used by GRPO.

4 Experiments

The main comparisons and HAM ablations use a fixed $\alpha = 0.3$ across all LIBERO task suites. Alternative values are evaluated only in the sensitivity analysis. Experimental settings are provided in Appendix A.2.

4.1 Simulation Experiments

We first conduct simulation experiments on the LIBERO benchmark to evaluate the effectiveness of the proposed hallucination-aware reinforcement learning pipeline. We compare HaWMPO with three representative baselines: the vanilla OpenVLA-OFT policy, WMPO, and WoVR*. All methods are evaluated on three LIBERO task suites (Spatial, Goal, and Object), and we report the success rate on each suite together with the average.

Table 1: Comparison of success rates across Spatial, Goal, and Object task categories. HaWMPO uses $\alpha = 0.3$ for every task suite and achieves the highest average performance.

Method	Spatial	Goal	Object	Avg
OpenVLA-OFT-base	61.5	48.2	36.3	48.7
WMPO	67.8	54.6	48.0	56.8
WoVR*	69.2	**64.0**	49.6	60.9
HaWMPO (Ours)	**77.2**	61.6	**52.2**	**63.7**

Our reproduced WoVR* baseline does not use the PACE co-evolution strategy from the WoVR paper. HaWMPO and WoVR* train the world model once, freeze it, and optimize the policy for 200

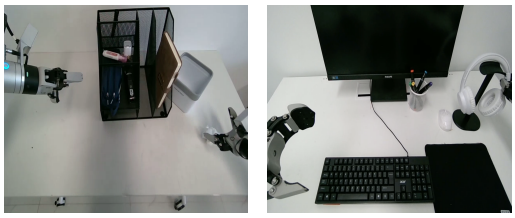
steps; HaWMPO uses $\alpha = 0.3$ on every suite. The OpenVLA-OFT-base row is the one-shot SFT checkpoint before these updates, and it matches the step-0 points in Fig. 4 (Spatial 61.5, Goal 48.2, Object 36.3). WMPO is compared at the same 200-step budget. WoVR* can therefore differ from the original WoVR paper, which includes PACE.

As shown in Table 1, the fixed $\alpha = 0.3$ setting reaches an average success rate of 63.7%. These three suite scores are exactly the $\alpha = 0.3$ curves in Fig. 4 at 200 steps, not the best α on each suite. The average improves over the SFT base by 15.0 points, over WMPO by 6.9 points, and over WoVR* by 2.8 points. HaWMPO is highest on Spatial (77.2%, +8.0 over WoVR*) and Object (52.2%, +2.6), and is 2.4 points below WoVR* on Goal (61.6% versus 64.0%).

4.2 Real-World Experiments

We evaluate HaWMPO on the G1 robot using two tasks (Fig. 3): *Tissue-to-Box*, placing a tissue inside a white storage box, and *Headphone-on-Stand*, placing headphones onto a stand. Success requires the tissue to remain inside the box or the headphones to remain supported by the stand after release.

We compare against the SFT-trained OpenVLA-OFT base policy and WoVR, with 20 trials per method per task under the same evaluation protocol. This WoVR entry is the same frozen-world-model, no-PACE baseline as WoVR*, and HaWMPO again uses $\alpha = 0.3$. Table 2 shows that HaWMPO achieves the highest observed success rates on both tasks: 85% and 75%. Its average success rate of 80.0% exceeds the base policy and WoVR by 12.5 and 7.5 percentage points, respectively. These results provide preliminary evidence of benefits in physical manipulation; broader evaluation remains necessary.



(a) Tissue-to-Box (b) Headphone

Figure 3: Real-world G1 setups for the 20-trial evaluation. Tissue-to-Box places a tissue into the white box; Headphone-on-Stand places headphones onto the stand.

Table 2: Success rates (%); successes out of 20 trials in parentheses.

Method	Tissue-to-Box	Headphone
Base VLA	75% (15/20)	60% (12/20)
WoVR	80% (16/20)	65% (13/20)
HaWMPO	**85% (17/20)**	**75% (15/20)**

4.3 Analysis of the Reward-Soft Mechanism

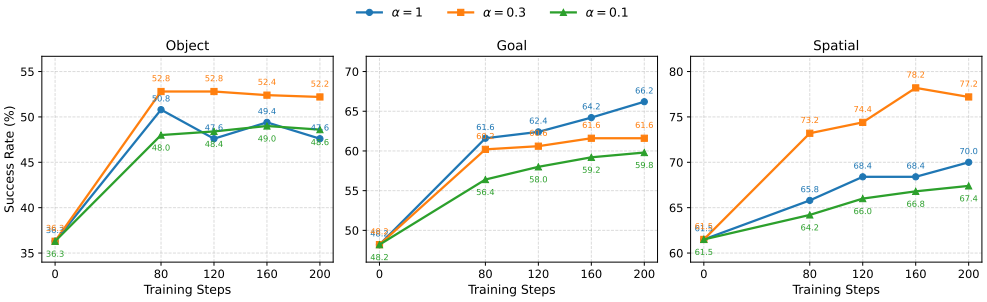


Figure 4: Sensitivity of success rate to α. Reported comparisons fix $\alpha = 0.3$, which is preferred on Object and Spatial. Goal is higher at $\alpha = 1$; that setting is not used in Table 1.

We further investigate the effect of the Reward-Soft mechanism, focusing on the hyperparameter α in Eq. (3), which controls how strongly the hallucination score modulates the final reward. Intuitively, an overly large α suppresses the reward of high-hallucination rollouts too aggressively, discarding virtual experience that remains useful despite visual imperfections; an overly small α instead lets unreliable trajectories bias policy updates. As shown in Fig. 4, the preferred penalty is task-dependent: $\alpha = 0.3$ gives the best and most stable results on Object and Spatial (52.2% and 77.2% at 200 steps), whereas Goal benefits from a stronger penalty ($\alpha = 1$, 66.2%). Across all three suites, $\alpha = 0.1$ remains below $\alpha = 0.3$, so some hallucination-aware modulation matters, but

a larger penalty is not uniformly better. We therefore keep $\alpha = 0.3$ fixed for every suite in Table 1 and the HAM ablation, rather than selecting a different value per suite. The 66.2% Goal result is only this sensitivity outcome and is not included in the reported average of 63.7%.

4.4 Analysis of the Hallucination-Aware Model

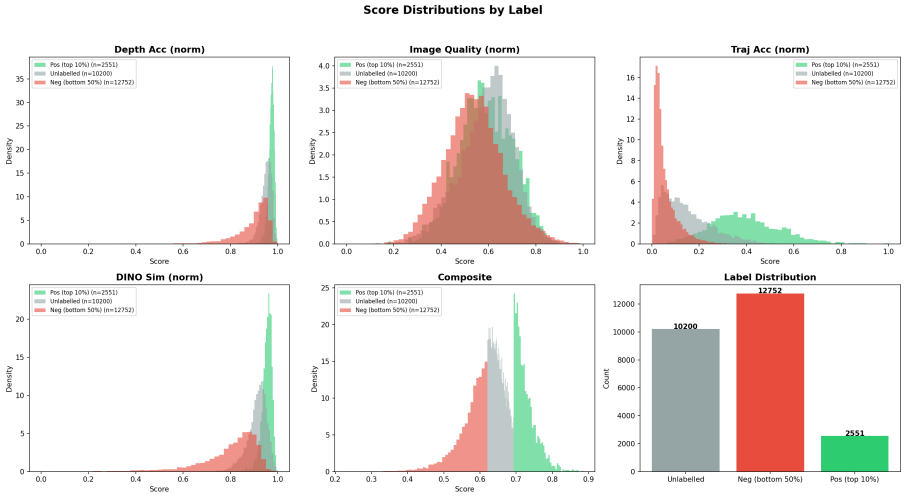


Figure 5: Score distributions of world-model chunks used to build HAM supervision. Pos, Unlabelled, and Neg are the top 10%, middle 40%, and bottom 50% of the composite, and are not the regression labels. Trajectory accuracy separates Pos from Neg; image quality overlaps.

We evaluate HAM independently of downstream policy performance to assess its reliability signal for Reward-Soft. The evaluation uses 50 video chunks from 50 trajectories in LIBERO, covering 10 tasks with five chunks per task. Human annotations identify 18 hallucinated and 32 non-hallucinated chunks.

We compare HAM with four single-metric baselines: MUSIQ image quality, DINO feature similarity, optical-flow trajectory consistency, and depth discrepancy. All scores are oriented so that higher values indicate greater predicted hallucination. Each method is evaluated on the same annotated chunks using AUROC and average precision (AP), with 95% confidence intervals estimated from 2,000 trajectory-level bootstrap resamples. DINO, trajectory, and depth baselines require corresponding observed future frames as references; HAM and MUSIQ do not.

Table 3: Hallucination detection against human annotations on 50 LIBERO chunks. Brackets show trajectory-bootstrap 95% confidence intervals. †Requires observed future frames.

Method	AUROC ↑	AP ↑
MUSIQ-only	0.3056 [0.1410, 0.4798]	0.3750 [0.2079, 0.5561]
DINO-only†	0.9062 [0.8182, 0.9733]	0.8299 [0.6638, 0.9563]
Trajectory-only†	0.8872 [0.7883, 0.9610]	0.8069 [0.6276, 0.9388]
Depth-only†	0.8941 [0.7912, 0.9745]	0.8644 [0.7258, 0.9649]
HAM	**0.9375** [0.8648, 0.9911]	**0.8952** [0.7660, 0.9885]

As shown in Table 3, HAM achieves the highest AUROC (0.9375) and AP (0.8952). The reference-based metrics also provide informative signals, whereas MUSIQ-only performs substantially worse, suggesting that image quality alone is insufficient for detecting hallucinations in this evaluation set. Fig. 5 shows the same pattern on the training chunks: trajectory accuracy separates the groups, while image quality overlaps. Those percentile groups only visualize the dataset; the regression target is the reversed, normalized composite in Eq. (1), with correlations and the PCA view in Appendix A.4. HAM does not need observed future frames at rollout time. These comparisons are descriptive; statistical significance has not been established. Separately, HAM achieves an MAE of 0.0174, an RMSE of 0.0250, and a Spearman correlation of 0.9310 against the composite proxy targets. These regression metrics measure agreement with the constructed supervision, whereas the human-label evaluation independently assesses hallucination detection. Fig. 6 provides complementary quali-

tative examples: higher HAM scores correspond to more evident discrepancies from the observed frames. Together, these findings support using HAM as a reliability signal for downweighting unreliable imagined chunks through Reward-Soft, complementing the downstream policy ablation.

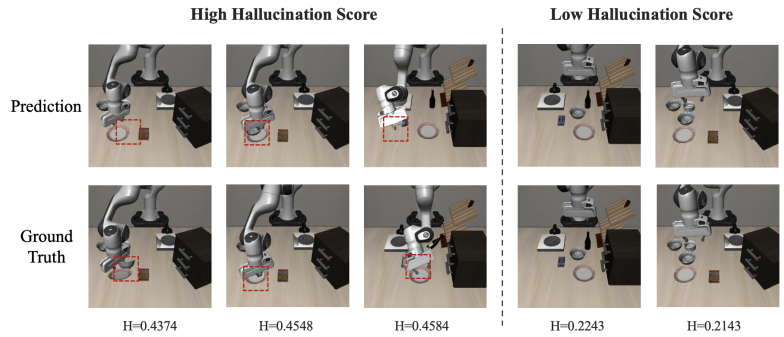


Figure 6: World-model predictions with higher and lower HAM scores. Higher scores correspond to more evident discrepancies from the observed frames.

4.5 Ablation Study

Removing HAM also disables score-based Reward-Soft, so the two are ablated together. The comparison therefore measures the contribution of the combined reliability estimation and reward modulation pathway; it does not isolate the contribution of either component on its own. We compare the same training checkpoints across the three suites and also report their arithmetic mean.

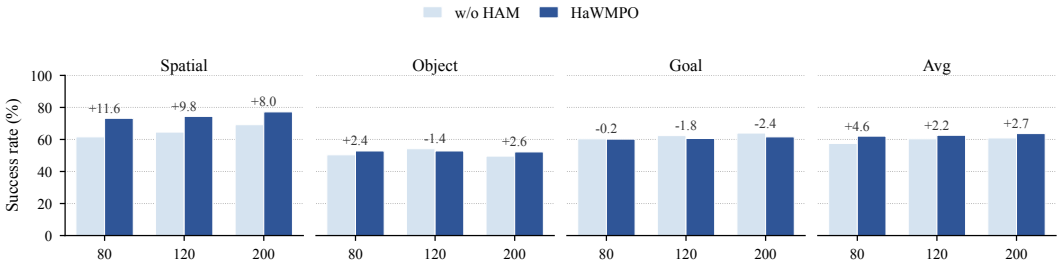


Figure 7: Ablation of HAM and Reward-Soft at $\alpha = 0.3$. Avg is the mean of the three suites. Labels are HaWMPO minus w/o HAM, in percentage points.

Fig. 7 is suite-dependent: Spatial rises by 8.0–11.6 points at every checkpoint, Object is 1.4 points lower at step 120, and Goal stays 0.2–2.4 points lower. The 66.2% Goal result at $\alpha = 1$ is not included here or in Table 1.

At 200 steps, the joint mechanism improves Spatial by 8.0 points and Object by 2.6 points, while Goal decreases by 2.4 points. The positive average gain therefore reflects a balance across suites rather than a uniform improvement on every task category. In particular, the persistent Spatial gains contrast with the smaller and checkpoint-dependent Object differences. This pattern is consistent with the sensitivity analysis: the usefulness of a fixed reliability penalty depends on the suite, even when the world model and HAM remain frozen throughout policy optimization.

5 Conclusion and Limitations

HaWMPO brings action-conditioned hallucination estimates into world-model-based VLA policy optimization through Reward-Soft. With $\alpha = 0.3$ fixed across the LIBERO suites, it achieves the highest average success rate, improving Spatial and Object while remaining below WoVR* on Goal, and the two physical tasks provide preliminary support. These results rest on proxy supervision that cannot guarantee physical validity, and the reliability penalty is not equally useful on every suite. The study covers three LIBERO suites and two manipulation tasks; HAM and the world model are trained separately and held fixed during policy optimization, so joint adaptation and broader evaluation remain open. Reward-Soft downweights unreliable chunks instead of discarding them.

AI USE STATEMENT

The action-conditioned Wan2.2 world model is a generative AI component of the research method. It generates synthetic visual rollouts for policy optimization and predicted video chunks for hallucination-model supervision, as described in Section 3 and Appendix A. OpenAI Codex was used to assist with the conversion of the manuscript to the ICLR 2027 template, anonymization, LaTeX formatting and cross-reference checks, drafting the submission statements, and revising mathematical notation and experimental descriptions for consistency with the reported results and available implementation. The original figure assets and numerical table entries were retained during this conversion.

ETHICS STATEMENT

This work studies robot policy optimization using generated visual experience. Inaccurate imagined transitions can yield unreliable policies, and hallucination scores are proxies for rollout reliability rather than safety guarantees. The physical experiments cover two manipulation tasks; deployment beyond these settings requires further validation, appropriate supervision, and safeguards against unintended robot actions. Human annotations used to evaluate hallucination detection should be interpreted within the scope of the reported evaluation.

REPRODUCIBILITY STATEMENT

Section 3 describes the hallucination targets, model architecture, reward modulation, and optimization procedure. Section 4 reports the baseline comparisons, evaluation metrics, and ablations. Appendix A.2 describes the experimental settings, and Appendix A.1 provides pseudocode. The remaining implementation appendix describes data construction and analysis, the baseline adaptation, real-world evaluation, and the world-model simulator.

REFERENCES

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Hao Chen, Jiaming Liu, Zhonghao Yan, Nuowei Han, Renrui Zhang, Chenyang Gu, Jialin Gao, Ziyu Guo, Siyuan Qian, Yinxi Wang, et al. Last-r1: Reinforcing action via adaptive physical latent reasoning for vla models. *arXiv preprint arXiv:2604.28192*, 2026.

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*, 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. *arXiv preprint arXiv:2502.05450*, 2025b.

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. *arXiv preprint arXiv:2501.16664*, 2025.

Zhennan Jiang, Shangqing Zhou, Yutong Jiang, Zefang Huang, Mingjie Wei, Yuhui Chen, Tianxing Zhou, Zhen Guo, Hao Lin, Quanlu Zhang, et al. Wovr: World models as reliable simulators for post-training vla policies with rl. *arXiv preprint arXiv:2602.13977*, 2026.

Joy Future Academy. Joyai-ra 0.1: A foundation model for robotic autonomy. *ArXiv*, abs/2604.20100, 2026. URL https://api.semanticscholar.org/CorpusID:287668520.

JoyAI-RA Team. Joyai-ra 0.5: Scaling robot manipulation learning via dual action alignment, 2026. URL https://api.semanticscholar.org/CorpusID:290852585.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

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

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *ArXiv*, abs/2601.16163, 2026. URL https://api.semanticscholar.org/CorpusID:284935587.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

Pei-Dong Liu, Yongce Liu, Song Guo, Fu-Yu Ma, Zhihao Yuan, Aosheng Li, Zeng Kai Chen, Wenhao Li, Tianle Zhang, Mingyan Li, Jiale Zhang, Junlin Xiong, Zhiyuan Xiang, Dafeng Chi, Yu-Hsiang Zhuang, Ruodai Li, Li qiong Luo, Wei Tan, Dongjiang Li, Yihang Li, Qin He, Jiaming Liang, Mingxia Luo, Chengman Cai, Hui Zhang, Peng Hao, Song Wang, Ning Qiao, Yince Gao, Lei Kang, Junwu Xiong, Jiawei Li, Hui Shen, Yichen Gong, Nan Duan, and Liang Lin. Joyai-sim: A simulation-enabled interconversion toolchain for the embodied data pyramid. *ArXiv*, abs/2606.16776, 2026. URL https://api.semanticscholar.org/CorpusID:289300675.

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. *arXiv preprint arXiv:2505.18719*, 2025.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqin Wang, Yicheng Feng, and Zongqing Lu. Being-h0.5: Scaling human-centric robot learning for cross-embodiment generalization. *ArXiv*, abs/2601.12993, 2026. URL https://api.semanticscholar.org/CorpusID:284909770.

Nvidia, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, LinxiJimFan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyuan 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, Zhen-Teng 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. *ArXiv*, abs/2503.14734, 2025. URL https://api.semanticscholar.org/CorpusID:277113335.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Yi Wang, Xinchen Li, Pengwei Xie, Pu Yang, Buqing Nie, Yunuo Cai, Qinglin Zhang, Chendi Qu, Jeffrey Wu, Jianheng Song, et al. Learning while deploying: Fleet-scale reinforcement learning for generalist robot policies. *arXiv preprint arXiv:2605.00416*, 2026.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026.

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. *arXiv preprint arXiv:2509.24948*, 2025.

Sheng Xu, Ruixing Jin, Huayi Zhou, Bo Yue, Guanren Qiao, Yunxin Tai, Yueci Deng, Kui Jia, and Guiliang Liu. From reaction to anticipation: Proactive failure recovery through agentic task graph for robotic manipulation. *arXiv preprint arXiv:2605.11951*, 2026.

Chao Yu, Yuanqing Wang, Zhen Guo, Hao Lin, Si Xu, Hongzhi Zang, Quanlu Zhang, Yongji Wu, Chunyang Zhu, Junhao Hu, Zixiao Huang, Mingjie Wei, Yuqing Xie, Ke Yang, Bo Dai, Zhexuan Xu, Xiang-Yun Wang, Xu Fu, Zhihao Liu, Kang Chen, Weilin Liu, Gang Liu, Boxun Li, Jianlei Yang, Zhi Yang, Guohao Dai, and Yu Wang. Rlinf: Flexible and efficient large-scale reinforcement learning via macro-to-micro flow transformation. *ArXiv*, abs/2509.15965, 2025. URL `https://api.semanticscholar.org/CorpusID:281410732`.

Zhihao Yuan, Shuyi Jiang, Chun-Mei Feng, Yao Zhang, Shuguang Cui, Zhen Li, and Na Zhao. Scene-r1: Video-grounded large language models for 3d scene reasoning without 3d annotations. *ArXiv*, abs/2506.17545, 2025. URL `https://api.semanticscholar.org/CorpusID:279999995`.

Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Siwei Han, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. Grape: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. *arXiv preprint arXiv:2511.09515*, 2025.

Zheng Zhu, Xiaofeng Wang, Wangbo Zhao, Chen Min, Bohan Li, Nianchen Deng, Min Dou, Yuqi Wang, Botian Shi, Kai Wang, et al. Is sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*, 2024.

APPENDIX

A IMPLEMENTATION DETAILS

A.1 OPTIMIZATION PROCEDURE

Algorithm 1 Hallucination-Aware World Model-Based Policy Optimization (HaWMPO)

Require: Initial policy π_θ; frozen world model M_ϕ, HAM f_ψ, reward model; group size $G > 1$; optimization budget K_{iter}; chunk length $K = 8$; $\alpha = 0.3$ for the main comparisons.

for $k = 1, \ldots, K_{\text{iter}}$ **do**
  Set behavior policy $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$.
  Sample groups of G rollouts sharing a task and initial condition.
  for each group and rollout i **do**
    Initialize reference image and four-frame recent context.
    while the rollout has not terminated or reached its horizon **do**
      Sample an eight-action chunk from $\pi_{\theta_{\text{old}}}$.
      Generate eight future frames with M_ϕ.
      Predict one chunk score H_c with f_ψ.
      Obtain and retain the raw action rewards $r_{c,1:K}$.
      Scale each reward: $\widetilde{r}_{c,j} \leftarrow \text{clip}_{[0,1]}(1 - \alpha H_c) r_{c,j}$.
      Record the rollout and update the four-frame recent context.
    end while
    Sum modulated rewards up to termination to obtain S_i.
  end for
  Compute group masks from mean unmodulated returns in $[0.5, 4.5]$.
  Compute group-relative advantages using Eq. (4).
  Broadcast advantages to valid tokens; mask filtered groups and invalid tokens.
  Update π_θ by minimizing $-\mathcal{J}(\theta)$ in Eq. (6).
end for
return Post-trained policy π_θ.

A.2 EXPERIMENTAL SETTINGS

Policy and benchmark. We use OpenVLA-OFT as the base policy and initialize it by one-shot supervised fine-tuning, using one expert demonstration per task. Simulation experiments cover the LIBERO Spatial, Object, and Goal suites. Reinforcement-learning methods are compared after 200 policy optimization steps; the base policy is the SFT checkpoint and does not receive these RL updates. The world model is trained before policy optimization and then held fixed. The main comparison and joint HAM/Reward-Soft ablation use $\alpha = 0.3$ on every suite, including the physical G1 policies. The $\alpha = 0.1$ and $\alpha = 1$ runs in Fig. 4 are separate sensitivity experiments. At 200 steps they produce Goal 59.8% and 66.2%, respectively, and neither value replaces the Goal entry 61.6% or the average 63.7% in Table 1.

Optimization. The policy learning rate is 2.0×10^{-5}, with Adam-family optimization, weight decay 0.01, and gradient-norm clipping at 1.0. Experiments use 8 NVIDIA H200 GPUs. The GRPO setup is actor-only: no value network is trained, and the generic configuration's value-network learning-rate field is inactive. The available LIBERO configurations specify group size $G = 8$ and 64 parallel training environments; eight rollouts therefore form each comparison group. Clipping bounds are $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$. The reference-KL coefficient and entropy bonus are zero; the embodied actor uses the standard clipped surrogate without an additional dual-clipping term. Group scores are undiscounted sums of modulated rewards, normalized with sample standard deviation plus 10^{-6}. Before HAM-based modulation, reward filtering retains groups whose mean unmodulated rollout return lies in $[0.5, 4.5]$. Other groups are masked from the policy loss. Valid-token losses are averaged within trajectories, then across trajectories, with filtered trajectories contributing zero.

Reward implementation. A separate ResNet-based visual classifier evaluates generated frames. Spatial and Object use an image-based model, while Goal additionally conditions on a learned task embedding. The implementation rounds sigmoid outputs to binary success indicators. With relative rewards enabled, each action reward is the difference of consecutive indicators multiplied by the reward coefficient 5. These differences telescope, so an unmodulated rollout return equals 5 times the success indicator at the last counted step when the running value starts at zero. The group filter $[0.5, 4.5]$ then keeps groups whose mean return corresponds to a terminal-success rate between 0.1 and 0.9. HAM supplies one scalar per generated eight-frame chunk. Reward-Soft then broadcasts $\mathrm{clip}_{[0,1]}(1 - 0.3H_c)$ across that chunk's action rewards. This scaling is applied after the group filter and before the return used for advantages. It does not average the reward sequence, hard-reject a chunk, or introduce a critic. The code path used for the reported runs fixes the coefficient at 0.3; the other curves in Fig. 4 require changing that coefficient and are not mixed into one policy.

Physical evaluation. The G1 experiments use 20 trials per method per task. Success is assessed from actual robot execution, as specified in Appendix A.6, rather than from the world model or its learned reward classifier.

A.3 Data Collection and HAM Supervision

When training the hallucination-aware model, we first adopt the world model to perform chunk-wise inference and generation on the collected real dataset, so as to construct the predicted dataset. Next, we revise several metrics in WorldArena. Leveraging models including DINOv3 and Depth-Anything-v3 together with trajectory optical flow methods, we extract features from both real and predicted datasets. Four hallucination metrics corresponding to each chunk are calculated, namely depth information, DINO feature similarity, image quality and trajectory optical flow similarity. The resulting data is finally used as the training set for our hallucination-aware model.

Score direction and implementation. The preprocessing script normalizes depth discrepancy in the inverse direction and the other three metrics in the quality-increasing direction. It combines them through a weighted average C_c, then the HAM trainer converts composite quality to hallucination severity using Eq. (1). The metric weights are configurable; the current preprocessing defaults are $(2, 1, 3.5, 3.5)$ for depth, MUSIQ, trajectory, and DINO, respectively. Equal weighting is a different configuration, not a property of the composite formula itself. The trainer estimates $C_{\min}$ and $C_{\max}$ from its input manifest, retains those bounds with the checkpoint, and uses them during inference-side target comparisons. It returns one sigmoid score per video chunk. Smooth L1 regression with unit transition is quadratic for errors between normalized predictions and targets in $[0, 1]$.

A.4 Training Data Analysis

In robotic settings, world model hallucinations consist of intertwined failures in geometry, visual appearance, motion coherence and semantic alignment. We adopt four complementary metrics: Depth Accuracy for geometric consistency against ground truth, Image Quality scored by MUSIQ, Trajectory Accuracy for optical flow motion consistency via NDTW, and DINO Similarity for frame-level semantic and appearance matching using DINOv3. Fig. 5 shows the resulting score distributions. All metrics are oriented and min–max normalized before combination into the quality-oriented Composite score C_c. We split samples by scores: the top 10% as high-quality anchors, the bottom 50% as typical hallucination cases, and the remaining 40% as unlabeled transitions. Statistics show Composite is strongly correlated with Trajectory Accuracy (0.72) and DINO Similarity (0.80), while Depth Accuracy provides auxiliary geometric constraints. Image Quality has nearly no correlation and even weak negative correlation with other metrics, suggesting that visual quality alone does not establish the absence of hallucinations. The PCA visualization describes variation in these metrics and separation of the score-defined groups; it is not an independent hallucination-detection test.

The percentile groups describe the dataset for analysis; they are not the regression labels. HAM receives continuous hallucination targets obtained by reversing the normalized Composite score, as in Eq. (1). It takes an eight-frame predicted chunk, conditioning actions, the current image, and an anchor image, and outputs one scalar for the whole chunk. The supervision combines motion, geometry, appearance, and image quality rather than relying on sharpness alone. Given the heavily left-skewed Trajectory distribution, limited positive samples and score saturation of Depth and

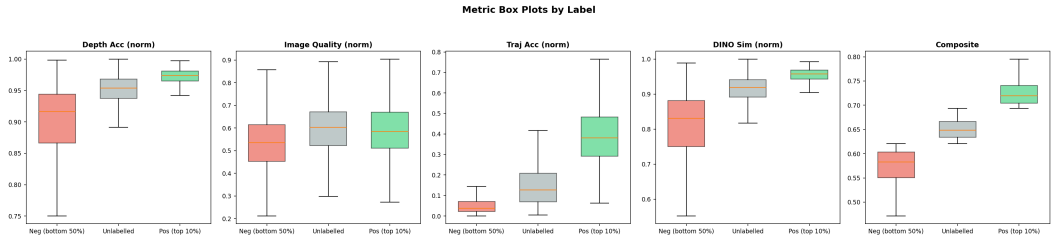


Figure 8: Median and interquartile ranges of each metric across the three label groups. Trajectory accuracy, DINO similarity, depth accuracy, and composite score increase monotonically from Neg to Pos, with composite showing near-complete separation of IQRs; image quality differs little across groups and is a poor standalone discriminator between faithful predictions and hallucinations.

DINO metrics, we use the reversed, normalized Composite as a multi-metric proxy target for training to align hallucination detection with robot decision-making. Motion-related hallucinations should be prioritized, and image quality should not be used to judge hallucinations.

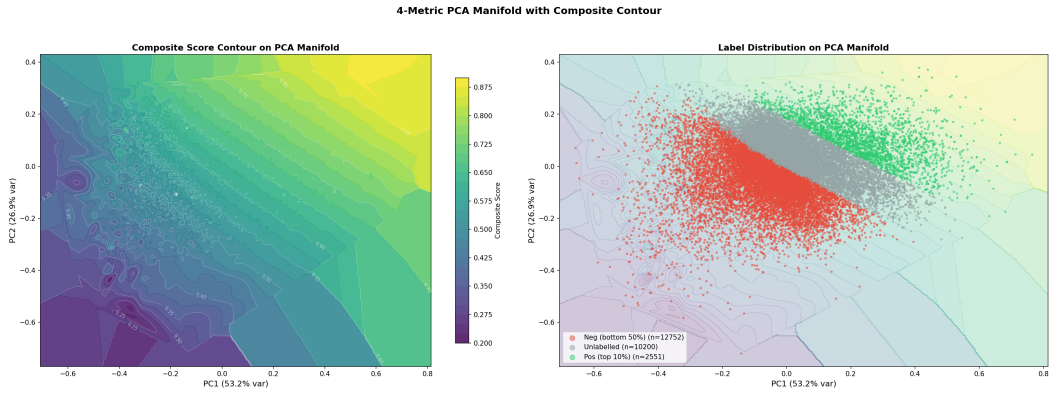


Figure 9: Two-dimensional PCA manifold of the four metrics: left, composite score contours; right, label scatter overlaid on faded contours. PC1 (53.2% variance) is driven mainly by trajectory accuracy and DINO similarity; PC2 (26.9%) by image quality and depth accuracy. Composite increases diagonally, with Pos samples concentrated in the upper-right and Neg in the lower-left, showing that overall quality is geometrically separable in metric space and suitable as continuous supervision for the hallucination-aware model.

A.5 Baseline

Our method is an extension of WoVR, which serves as our baseline. We retain its action-conditioned world-model architecture and add HAM-based reward modulation to policy optimization. We inherit the Action-Conditioned World Model from WoVR, using Wan2.2 as the backbone and injecting an action MLP to generate future image chunks. We do not include WoVR's PACE strategy in our setup. The world model remains frozen during policy optimization, so the comparisons isolate the effect of hallucination-aware reward modulation. We denote this reproduction by WoVR* in simulation; the WoVR label in the physical-task table refers to the same no-PACE baseline protocol. The effect of combining HaWMPO with PACE is not evaluated here.

A.6 Real-World Experiments

We evaluate HaWMPO on the G1 robot using two physical manipulation tasks: *Tissue-to-Box*, in which the robot places a tissue into a white storage box, and *Headphone-on-Stand*, in which it places headphones onto a stand. Fig. 10 shows actual robot executions recorded during physical testing, rather than world-model-generated predictions.

We compare HaWMPO ($\alpha = 0.3$) with the SFT-trained OpenVLA-OFT base policy and WoVR under the same evaluation protocol, with 20 trials per method per task. A trial is successful if the tissue remains inside the box or the headphones remain supported by the stand after release. On

Tissue-to-Box, the base policy, WoVR, and HaWMPO achieve success rates of 75% (15/20), 80% (16/20), and 85% (17/20), respectively. On Headphone-on-Stand, the corresponding success rates are 60% (12/20), 65% (13/20), and 75% (15/20).

Across the two tasks, HaWMPO achieves an average success rate of 80.0%, compared with 67.5% for the SFT baseline and 72.5% for WoVR, yielding improvements of 12.5 and 7.5 percentage points, respectively. These results provide preliminary evidence of benefits in physical manipulation, although evaluation on additional tasks and trials is needed to establish broader generalization.



Figure 10: Real-world robot executions recorded during physical testing on the G1 robot. The upper row shows the Tissue-to-Box task, and the lower row shows the Headphone-on-Stand task. These images depict actual robot executions, not world-model-generated rollouts; aggregate success rates are reported in Table 2.

A.7 World Model as Simulator

Model architecture. We use an action-conditioned adaptation of Wan2.2-TI2V-5B as the world-model simulator. The model consists of a video variational autoencoder (VAE) and a latent-space diffusion Transformer (DiT). The VAE encodes visual observations into spatiotemporal latents and decodes predicted latents into RGB frames. The DiT contains 30 Transformer blocks with a hidden dimension of 3072 and 24 attention heads. Each block combines self-attention over video tokens, cross-attention to conditioning tokens, and a feed-forward network.

Robot actions are incorporated through two conditioning paths. First, an MLP projects individual actions into tokens that serve as keys and values for cross-attention. Second, temporally grouped actions are projected into embeddings that are added to the diffusion-timestep embeddings, thereby conditioning the modulation of the Transformer blocks. This design conditions visual predictions on the commanded motion rather than relying on visual context alone.

Inputs and outputs. Each simulator step receives five conditioning RGB frames: a fixed reference frame and the four most recent frames. It additionally receives an eight-step action chunk, together with the action context associated with the conditioning frames. In LIBERO, each action is seven-dimensional, comprising three translational components, three rotational components, and one gripper command. All frames have a resolution of 256×256.

Let $\mathcal{C}_t$ denote the five-frame visual context, $\mathcal{U}_t$ the associated action context, and $A_t = (a_t, \ldots, a_{t+K-1})$ the proposed action chunk. The world model predicts

$$\widehat{O}_{t+1:t+K} \sim p_\phi(\cdot \mid \mathcal{C}_t, \mathcal{U}_t, A_t), \qquad K = 8, \tag{7}$$

where ϕ denotes the world-model parameters and $\widehat{O}_{t+1:t+K}$ contains eight predicted RGB frames. The generation pipeline decodes a 13-frame sequence, including the five conditioning frames; only the eight future frames are returned as the simulated transition. The conditioning latents are held fixed during iterative denoising. The simulator is conditioned on images and actions; task instructions are supplied to the VLA policy rather than directly to the video-generation model in this implementation.

Closed-loop imagined rollouts. A rollout is initialized from recorded visual context. The VLA policy predicts an action chunk from the current observation and task instruction, and the world model generates the corresponding future observations. The last four generated frames then update the visual context, while the reference frame remains fixed. Repeating this process produces a closed-loop imagined trajectory. Thus, generation is autoregressive across chunks, whereas each future chunk is generated jointly through latent-space denoising.

The world model is trained before policy optimization and remains frozen during reinforcement learning. It predicts visual transitions, not rewards or hallucination scores: a separate reward model evaluates the generated observations, and HAM estimates their reliability. Reward-Soft uses the HAM scores to modulate the rewards used for policy optimization.

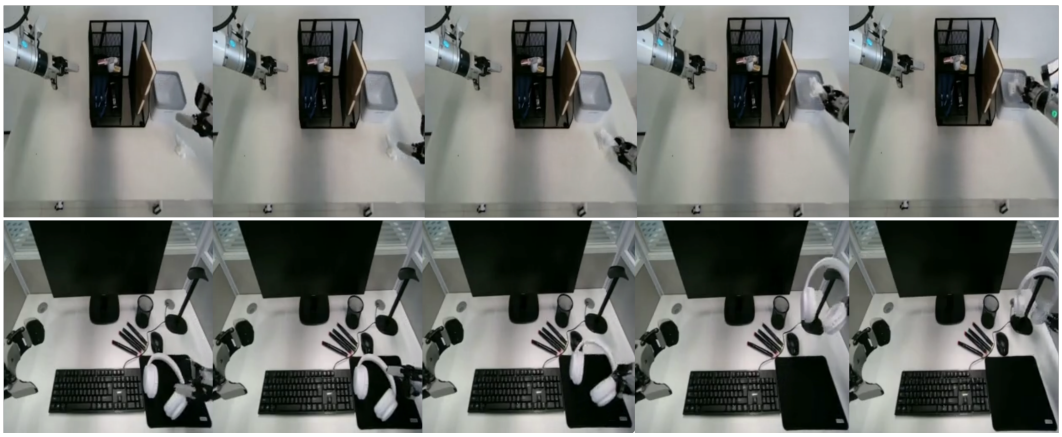


Figure 11: Selected world-model rollout frames for Tissue-to-Box (top), placing a tissue into a white storage box, and Headphone-on-Stand (bottom), placing headphones onto a stand. Frames are ordered chronologically from left to right. These examples illustrate action-conditioned visual predictions in physical manipulation scenes.

Fig. 11 presents qualitative examples of the generated rollouts. Although the predictions depict task-relevant changes in the scene, visual plausibility alone does not guarantee accurate contact dynamics or successful physical execution. Errors can accumulate as generated observations are reused as context, motivating the hallucination-aware reward modulation in HaWMPO.