World Action Verifier: Self-Improving World Models via Forward-Inverse Asymmetry

Yuejiang Liu[†,*], Fan Feng[‡,*], Lingjing Kong[§,*], Weifeng Lu[*], Jinzhou Tang[‡], Kun Zhang[§], Kevin Murphy[¶], Chelsea Finn[†], Yilun Du[‖]

[†]Stanford University [‡]UC San Diego [¶]University of British Columbia
[§]Carnegie Mellon University [‖]Harvard University
Project website: world-action-verifier.github.io

Abstract

General-purpose world models promise scalable policy evaluation, optimization, and planning, yet achieving the required level of robustness remains challenging. Unlike policy learning, which primarily focuses on optimal actions, a world model needs to be reliable over a much broader range of suboptimal actions, which are often insufficiently covered by action-labeled interaction data. To address this challenge, we propose World Action Verifier (WAV), a framework that enables world models to identify their own prediction errors and self-improve. The key idea is to decompose action-conditioned state prediction into two factors—state plausibility and action reachability—and verify each separately. We show that these verification problems can be substantially easier than predicting future states due to two underlying asymmetries: the broader availability of action-free data and the lower dimensionality of action-relevant features. Leveraging these asymmetries, we augment a world model with (i) a diverse subgoal generator obtained from video corpora and (ii) a sparse inverse model that infers actions from a subset of state features. By enforcing cycle consistency among generated subgoals, inferred actions, and forward rollouts, WAV provides an effective verification mechanism in under-explored regimes, where existing methods typically fail. Empirically, WAV doubles sample efficiency across six robotic manipulation tasks in RoboMimic and ManiSkill and boosts downstream policy reward by 22%. WAV also scales to the diverse multitask benchmark MMBench2, raising verification correlation by 23% and improving closed-loop planning performance by 11% on unseen tasks.

1 Introduction

World models—action-conditioned forward dynamics models that predict future states given specific actions or action chunks—have come to play an increasingly important role in robot learning [4, 37, 53, 110, 121, 138]. Recent works have shown that, when trained on action-labeled robot interactions alongside action-free internet videos [52, 98, 127], world models have the potential to not only generate controllable future dynamics but also enable scalable policy evaluation [68, 94, 109, 141], policy optimization [34, 40, 124, 125], and test-time planning [39, 41, 54, 93, 139].

Despite remarkable progress, building a general-purpose world model that is robust enough for various downstream applications remains difficult. A central challenge is *action following*: predicting future states that faithfully reflect the effects of the given actions [102]. Unlike policy learning, which primarily focuses on modeling optimal actions, a world model must be reliable across a much broader action distribution, including suboptimal, exploratory, and even random actions encountered during policy learning or evaluation [54, 64, 136]. However, collecting robot interactions covering diverse

[*]Core contributors. Correspondence to: yuejiang.liu@stanford.edu, f2feng@ucsd.edu.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

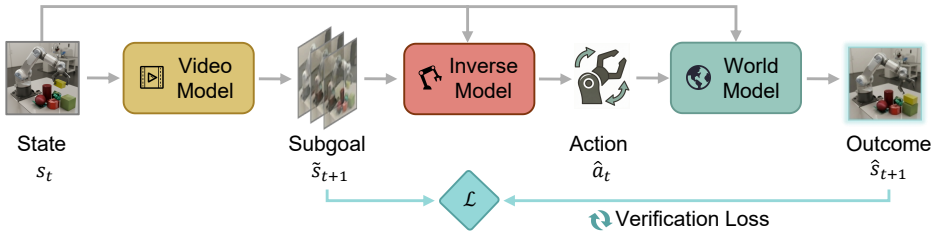

Figure 1: Overview of World Action Verifier, a framework that enables action-conditioned world models to verify their predictions and self-improve through an asymmetric forward–inverse cycle: (i) a *diverse* subgoal generator proposes plausible future states, (ii) a *sparse* inverse model infers actions from a relevant subset of state features, and (iii) the world model rolls forward and verifies consistency between its prediction and the proposed state.

actions is often slow, expensive, and sometimes even unsafe. Given a limited budget of robot data, deciding which specific interactions to collect remains a pressing challenge.

Previous work has sought to address this through two main approaches. One line of work relies on on-policy exploration, *i.e.*, gathering data by rolling out the policies of interest [35, 54, 76]. While effective for the considered policies, the learned model often degrades sharply beyond predefined policy sets, compromising its generality. Another line of work focuses on info-max exploration, actively seeking interactions that maximize information gain [58, 91, 101].

A common proxy for information gain is the prediction error of the world model, estimated before collecting the corresponding transition–a process we refer to as *world model verification*. This verification process, however, often suffers from a practical challenge: existing methods tend to be reliable in well-explored regions where additional data are largely redundant, but unreliable in under-explored regions where verification is critically needed. After all, the interactions that are most informative for exploration are precisely those where the least prior information exists for verification. This tension raises a central question:

How can we reliably verify the predictions of a world model in under-explored regimes?

To this end, we propose World Action Verifier (WAV), a framework that enables world models to verify their own predictions and self-improve through an asymmetric forward–inverse cycle. The core idea is to decompose action-conditioned forward predictions into two complementary factors: *state plausibility*, *i.e.*, whether a predicted state is visually realistic, and *action reachability*, *i.e.*, whether the predicted transition is physically achievable under the given actions.

This decomposition not only allows each factor to be verified separately, but also admits two crucial asymmetries: (i) the broader availability of action-free data: state plausibility can be verified using internet videos without action labels, which are far more abundant than the action-labeled robot interactions used to train the world model, and (ii) lower dimensionality of action-relevant features: action reachability can be verified based on a compact subset of state features relevant to the actions, which are much lower-dimensional than the full state the world model must predict.

Motivated by these asymmetries, we augment a world model with two additional components: a diverse subgoal generator obtained from video corpora, and a sparse inverse model trained to infer actions from a learned subset of state features. Together, these components induce a goal-oriented self-improvement cycle: the subgoal generator proposes plausible future states, the inverse model infers actions that could reach them, and the forward world model rolls out those actions to test whether the predicted states are consistent with the proposed subgoals (Figure 1).

Theoretically, we show that verification via a sparse inverse process is easier than dense forward generation, particularly in high-dimensional stochastic environments. Empirically, we evaluate WAV on MiniGrid [19], RoboMimic [142], ManiSkill [86], and MMBench-2 [43]. WAV doubles sample efficiency across the robotic manipulation environments, improves downstream reward by 22% under novel objects and interactions, and improves verification correlation by 23% relative to the strongest baseline. Our results suggest that the asymmetries between forward and inverse dynamics offer a promising ingredient for building self-improving world models.

2 Method: World Action Verifier for Self-Improving World Models

World models excel when grounded in action-labeled interaction data, yet collecting such data at scale is often prohibitively expensive. In this section, we present World Action Verifier (WAV), a self-improving framework that enables a world model to verify its own predictions and prioritize informative exploration. We first formalize the verification problem in a semi-supervised setting (Sec. 2.1), then decompose it into two more tractable subproblems (Sec. 2.2), and finally couple them into a goal-oriented exploration procedure for self-improvement (Sec. 2.3).

2.1 Preliminary: Semi-Supervised Verification of World Models

We consider a world model f_θ as an action-conditioned forward dynamics model, $\hat{s}^{t+1} = f_\theta(s^t, a^t)$, where s^t and a^t are the state and action, or action chunk, at time t, and $\hat{s}^{t+1}$ is the predicted successor state. Following recent training recipes [31, 52], we study a semi-supervised setting with two data sources: a small action-labeled robot interaction dataset $\mathcal{D}_{\text{act}} = \{(s^t, a^t, s^{t+1})\}$ and a large action-free video dataset $\mathcal{D}_{\text{vid}} = \{(s^t, s^{t+1}, \ldots)\}$. Typically, $\mathcal{D}_{\text{vid}}$ spans a much broader range of state transitions than $\mathcal{D}_{\text{act}}$.

Our goal is to improve f_θ not only on the narrow action distribution represented in $\mathcal{D}_{\text{act}}$, but also on the broader transition support reflected in $\mathcal{D}_{\text{vid}}$. However, the lack of action labels in online videos poses a key challenge for *action following*: rather than faithfully grounding predictions in the conditioning action, existing world models often hallucinate future states that may look visually plausible but are physically misaligned with the given action [83, 102]. A natural remedy is to collect additional action-labeled robot interactions [35, 54, 76]. Yet, since large-scale robot interaction data are costly to collect, a critical question arises: *which specific interactions should be prioritized to improve the world model most effectively?*

Intuitively, transitions that the model can already predict accurately yield little new knowledge. Instead, the data budget should be steered toward transitions that are likely to induce large prediction errors. More formally, for a transition (s^t, a^t, s^{t+1}), we define the true prediction error as

$$\varepsilon(s^t, a^t; s^{t+1}) := \ell\big(f_\theta(s^t, a^t), s^{t+1}\big)\,, \tag{1}$$

where $\ell(\cdot, \cdot)$ is a discrepancy measure in the state space. Since the true successor state s^{t+1} cannot be observed prior to execution, we aim to construct a *verifier* $\hat{\varepsilon}(s^t, a^t, \hat{s}^{t+1})$ that estimates this error. From an exploration standpoint, the verifier need not be perfectly calibrated, but it should preserve the relative ranking of prediction errors across candidate actions [47, 99]. That is, given two candidate actions a_i^t and a_j^t, with predicted successor states $\hat{s}_i^{t+1} = f_\theta(s^t, a_i^t)$ and $\hat{s}_j^{t+1} = f_\theta(s^t, a_j^t)$, an effective verifier for exploration should satisfy

$$\varepsilon(s^t, a_i^t; s_i^{t+1}) < \varepsilon(s^t, a_j^t; s_j^{t+1}) \implies \hat{\varepsilon}(s^t, a_i^t, \hat{s}_i^{t+1}) < \hat{\varepsilon}(s^t, a_j^t, \hat{s}_j^{t+1}). \tag{2}$$

2.2 Two Complementary Factors of Verification

A common approach to verifying forward predictions in Equation (2) is to leverage the internal knowledge of the world model itself, *e.g.*, extracting epistemic uncertainty from a single model [91] or measuring disagreement across multiple models [58, 101]. However, such verification methods often inherit the blind spots of the learned world model: they provide relatively reliable error estimates in well-explored regimes where the current world model is already accurate, but become much less reliable in under-explored regimes where accurate verification is most critical.

To overcome this issue, we take a different perspective: rather than directly verifying the overall correctness of a forward prediction, we decompose it into sub-conditions that are easier to verify. More specifically, motivated by the Bayes decomposition,

$$p(s^{t+1} \mid s^t, a^t) = \frac{p(a^t \mid s^t, s^{t+1})\, p(s^{t+1} \mid s^t)}{p(a^t \mid s^t)} \propto \underbrace{p(s^{t+1} \mid s^t)}_{\text{state}}\, \underbrace{p(a^t \mid s^t, s^{t+1})}_{\text{action}}, \tag{3}$$

we view a correct action-conditioned forward prediction as satisfying two complementary criteria:

- *State Plausibility*: whether the predicted next state is plausible under the environment dynamics.
- *Action Reachability*: whether the transition from s^t to s^{t+1} is consistent with the given action.

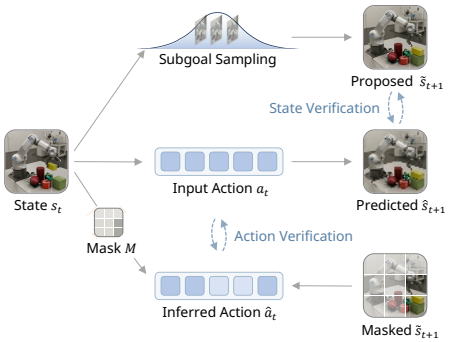

Figure 2: Decomposing model verification into state plausibility and action reachability.

Algorithm 1: WAV-Guided Exploration.

```
# f: world model, h: inverse model
# s: current state, g: subgoal generator
# D: current data, K: number of candidates

for each exploration iteration:
  s_g = g.sample(s, K)      # subgoals
  a = h.inverse(s, s_g)     # actions
  s_p = f.predict(s, a)     # outcomes
  scores = dist(s_g, s_p)   # disagreement
  idx = argmax(scores)      # max surprise
  s_n = env.step(a[idx])
  D.append((s, a[idx], s_n))
  f.update(D), h.update(D)
```

From this view, a correct prediction should both lie on the manifold of plausible futures and be reachable under the specified action. Crucially, each condition admits a verification strategy that is more tractable than predicting high-dimensional forward dynamics, which we will describe next.

State verification via distribution asymmetry. One common failure mode of high-dimensional forward prediction is poor visual plausibility. For example, a world model post-trained on limited robot interaction data may partially forget the general dynamics learned from video pretraining, resulting in blurry or inconsistent rollouts [134]. To detect such errors, we build a state verifier from the first factor in Equation (3), namely a state transition prior $g_\phi(s^{t+1} \mid s^t)$. Since this prior does not condition on actions, it can be trained not only on action-labeled robot interactions $\mathcal{D}_{\text{act}}$, but also on the much larger action-free video dataset $\mathcal{D}_{\text{vid}}$. Moreover, as the prediction error of a world model often compounds over longer horizons, we instantiate s^{t+1} as a future subgoal reached after an action chunk rather than as the immediate next frame. After training, g_ϕ allows us to sample a diverse set of K plausible future subgoals for state verification,

$$\{\tilde{s}_k^{t+1}\}_{k=1}^{K} \sim g_\phi(\cdot \mid s^t). \tag{4}$$

Action verification via dimensionality asymmetry. State plausibility alone does not imply correct forward prediction: for downstream policy use, predicted transitions must also be reachable under the specified action. To assess this condition, we build another verifier from the second factor in Equation (3), namely an inverse dynamics model $h_\psi(a^t \mid s^t, s^{t+1})$ that infers which action could connect a current state to a future state. While the inverse model h_ψ is action-dependent and cannot leverage more training data than the forward world model, it benefits from a lower effective dimensionality in both its outputs and its relevant inputs. For instance, in visually complex scenes, an action typically affects only one object at a time rather than the entire scene. Inspired by the observation that models attending to a compact set of causally relevant features often generalize better [77, 118], we explicitly impose a learnable sparsity mask M in the inverse dynamics model:

$$\hat{a}^t = h_\psi\big(M \odot s^t,\ M \odot s^{t+1}\big). \tag{5}$$

As illustrated in Figure 2, the subgoal generator g_ϕ and inverse model h_ψ provide two complementary components for verification: the former checks whether a candidate future is plausible, while the latter checks whether it is reachable through an inferred action.

2.3 Goal-Oriented Exploration

Given the two verification criteria above, we next couple them into a verification-driven exploration algorithm. A straightforward design is action-oriented exploration: sample candidate actions, roll out the forward world model f_θ, and then use the inverse model h_ψ to check whether the generated state transitions recover the original actions [128]. However, this ordering can be brittle in practice. Among the three components, the forward world model f_θ is often the least reliable when action-labeled data are scarce. As such, errors introduced by the initial forward rollout can produce off-manifold states, on which the subsequent inverse model also becomes unreliable.

We therefore pursue a goal-oriented alternative that places the forward world model as the final step in the verification cycle. At each time step t, we first sample plausible subgoals from the transition prior, infer actions that could reach those subgoals, and only then verify whether the action-conditioned world model can realize them:

$$s^t \xrightarrow{g_\phi} \tilde{s}_{1:K}^{t+1} \xrightarrow{h_\psi} \hat{a}_{1:K}^{t} \xrightarrow{f_\theta} \hat{s}_{1:K}^{t+1} \xrightarrow{\ell} \hat{\varepsilon}_{1:K} \xrightarrow{\max} a^\star. \tag{6}$$

For each candidate subgoal $\tilde{s}_k^{t+1}$, we measure how far the forward rollout $\hat{s}_k^{t+1}$ deviates from it. In practice, this discrepancy ℓ can be computed in the discrete state space, a continuous representation space, or a diffusion noise space, depending on the parameterization of the world model.

As summarized in Algorithm 1, we execute $a^\star$ associated with the largest discrepancy at each time step, add the resulting transition to $\mathcal{D}_{\text{act}}$, and iteratively update both the forward world model and the inverse dynamics model. By verifying candidate rollouts in parallel before acting, WAV uses additional computation to reduce unnecessary interactions and improve data efficiency.

3 Theory: When Is Inverse Verification Easier Than Forward Prediction?

Our method in Sec. 2 rests on a central premise: verifying actions through inverse dynamics can be substantially easier than predicting full future states with a forward world model. In this section, we provide a minimal theoretical analysis characterizing the conditions under which this forward–inverse asymmetry becomes most pronounced.

To isolate the key factors, we consider a linear–Gaussian setting with the observed state $s \in \mathbb{R}^{d_s}$ and action $a \in \mathbb{R}^{d_a}$. Assume one-step dynamics

$$s' = As + Ba + \xi, \qquad \xi \sim \mathcal{N}(0, \sigma_s^2 \mathbf{I}_{d_s}), \tag{7}$$

where σ_s captures transition stochasticity. We further assume that the action is recoverable from a low-dimensional action-relevant slice $z := Ms \in \mathbb{R}^{d_z}$ with $d_z \ll d_s$:

$$a = h(z, z') + \eta, \qquad h(z, z') := H \begin{bmatrix} z \\ z' \end{bmatrix}, \qquad \eta \sim \mathcal{N}(0, \sigma_a^2 \mathbf{I}_{d_a}), \tag{8}$$

where $z' := Ms'$ and σ_a measures irreducible ambiguity in recovering the action from (z, z').

We compare a dense forward model f_θ trained on $[s; a] \in \mathbb{R}^{d_s + d_a}$ against a sparse inverse model h_ψ trained on $[z; z'] \in \mathbb{R}^{2d_z}$, both fit by ordinary least squares (OLS) on n transitions from $\mathcal{D}_{\text{act}}$. To compare them in the same units, we evaluate in state space:

$$\mathcal{E}_F := \frac{1}{d_s} \mathbb{E}\big[\|f_\theta(s, a) - f(s, a)\|_2^2\big], \quad \mathcal{E}_I := \frac{1}{d_s} \mathbb{E}\big[\|f(s, h_\psi(z, z')) - f(s, h(z, z'))\|_2^2\big], \tag{9}$$

where $f(s, a)$ denotes the true dynamics. Since inverse errors enter the state prediction through the action channel, we define $\lambda := \|B\|_{\text{op}}$ as the worst-case amplification from action error to state error.

Proposition 3.1 (Informal)**.** *Under the stylized setup above, if both models are fit by OLS on n labeled transitions, then*

$$\frac{\mathbb{E}[\mathcal{E}_F]}{\mathbb{E}[\mathcal{E}_I]} \;\geq\; \underbrace{\Big(\frac{d_s + d_a}{2d_z} \cdot \frac{d_s}{d_a}\Big)}_{\textit{dimensionality}} \cdot \underbrace{\Big(\frac{\sigma_s}{\lambda\, \sigma_a}\Big)^2}_{\textit{stochasticity}} \cdot \underbrace{\Big(\frac{n - 2d_z - 1}{n - (d_s + d_a) - 1}\Big)}_{\textit{sample size}}, \tag{10}$$

provided $n > d_s + d_a + 1$ and $n > 2d_z + 1$. The exact statement and proof are in Appendix F.2.

Interpretation. The ratio in (10) factorizes into three terms. (1) *Dimensionality*: the forward model must estimate a map from $d_s + d_a$ inputs, whereas the sparse inverse uses only $2d_z$. (2) *Stochasticity*: forward prediction suffers from environment noise σ_s, while inverse verification suffers only from action-recovery ambiguity σ_a (scaled by λ). (3) *Sample size*: when n is only modestly larger than $d_s + d_a$, the forward estimator is far less stable. In practice, WAV helps most when (i) the verifier needs only a small agent-centric subset while the world model predicts a large scene (*large* d_s/d_z); (ii) uncontrolled dynamics inflate σ_s while the action imprint stays clean (*large* σ_s/σ_a); and (iii) action-labeled data are limited (*small* n). We validate each factor empirically in Sec. 4.1.1: varying the data budget isolates the sample-size term, increasing the number of objects raises the effective state dimension d_s, and adding noisy floors inflates σ_s while leaving σ_a unchanged.

4 Experiments

In this section, we empirically evaluate the effectiveness of WAV. We first assess the robustness of sparse inverse verification, then examine its benefits for the self-improvement of world models, and finally evaluate its scalability across diverse tasks. Concretely, we study the following research questions:

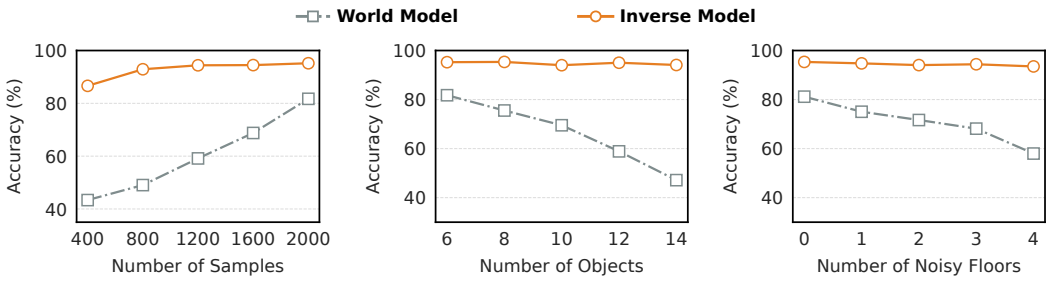

Figure 3: **Robustness of WAV verification on MiniGrid.** (Left) Sample efficiency comparison between Sparse IDM and the World Model with six objects. (Mid) Robustness to increasing state complexity. (Right) Robustness to growing environment stochasticity.

- *RQ1*: Are inverse dynamics models easier to learn than forward world models?
- *RQ2*: Does sparsity improve inverse-model generalization to unseen objects or interactions?
- *RQ3*: How effectively does forward–inverse asymmetry improve world-model learning?
- *RQ4*: Do these world-model gains translate into better downstream policy learning?
- *RQ5*: How effective is WAV for new environments with very limited interaction data?
- *RQ6*: Does WAV generalize to verification across diverse tasks at larger scale?

Baselines. We compare our method (Sec. 2) against the following exploration strategies:

- *Random:* randomly samples candidate interactions, serving as a lower-bound exploration strategy.
- *Uncertainty:* selects candidates with the highest predictive uncertainty [101].
- *Progress:* selects candidates with the largest learning progress, measured by the disagreement between two consecutive world models during training [58].
- *RLIR:* selects candidates with the largest disagreement between inverse-predicted and conditioning actions [128].
- *Vanilla IDM*: our method without the sparsity mask M in Equation (5).
- *Oracle:* selects candidates with the largest world-model prediction loss computed using ground-truth successor states, serving as an upper bound.

4.1 Experiments in Synthetic MiniGrid

Dataset. We collect approximately 50,000 transitions from three custom MiniGrid tasks: `Key Delivery`, `Ball Delivery`, and `Object Matching`, using a deterministic policy for each task. Half of these transitions are used to train the action-free subgoal generator as a video prior. The remaining data form an exploration pool, including an action-labeled seed set of 200 transitions and an unlabeled candidate set of 20,000 transitions for acquisition. To enable controlled evaluation, we construct additional random-play datasets by varying object counts and environmental stochasticity. Specifically, we vary the number of objects to study generalization under increasing scene complexity, and introduce noisy floor tiles whose colors change after each action to simulate stochastic observations. Data samples from these noisy environments are used exclusively for robustness evaluation. Additional details are provided in Appendix E.2.

4.1.1 Robustness of World Action Verification

To test whether WAV is a reliable verifier, we compare forward world models and inverse dynamics models under controlled distribution shifts, addressing ***RQ1*** and ***RQ2***.

Setup. We vary the amount of labeled training data $\{400, 800, 1200, 1600, 2000\}$ collected in environments with 6 objects, and evaluate on test transitions in environments with $\{6, 8, 10, 12, 14\}$ objects. In addition, to examine robustness against observation noise, we construct training datasets in 6-object environments with $\{0, 1, 2, 3, 4\}$ noisy floors. For direct comparison, we convert inverse model predictions into next state predictions: for each test pair, the IDM predicts an action, which we then execute in the simulator to obtain the induced next state. We report the same dynamics accuracy defined in Appendix E.2.3 for both the world model and the IDM-induced transition.

Results. For ***RQ1***, Figure 3 empirically validates the three factors identified in Proposition 3.1, each isolated by a separate controlled variable. Figure 3 (Left) isolates the *sample-size* factor: across data regimes, action inference using IDMs consistently outperforms learning action-conditioned world

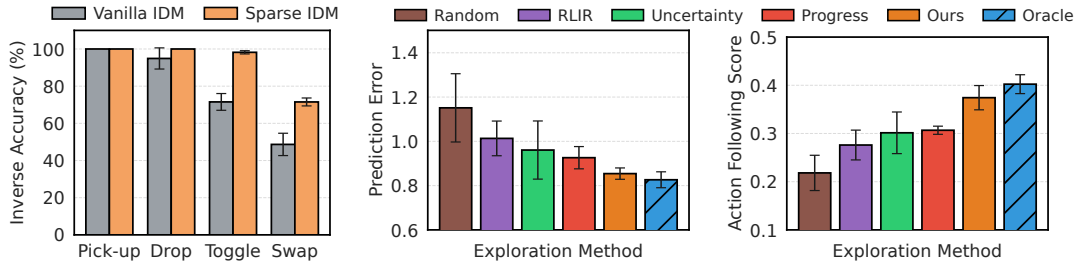

Figure 4: **Evaluation of world model learning with WAV on MiniGrid.** (Left) Prediction accuracy of Sparse IDM and Vanilla IDM. (Mid) Comparison of different exploration methods. (Right) Robustness of action following across different methods. Results are averaged across 5 seeds.

models, with the performance gap being most pronounced in the low-data regime, consistent with the diverging finite-sample term in (10) when n is only modestly larger than $d_s + d_a$. Figure 3 (Mid) isolates the *dimensionality* factor: increasing the number of objects raises the effective state dimension d_s while leaving the action-relevant subset d_z unchanged. The WM's performance degrades rapidly as state complexity increases, whereas the IDM remains stable, consistent with the growing ratio $(d_s + d_a)/2d_z$ in the dimensionality term. Figure 3 (Right) isolates the *stochasticity* factor: noisy floor tiles inflate observation noise σ_s while the action imprint on the agent-centric features remains clean (σ_a unchanged). The WM exhibits clear sensitivity to the induced observation noise, whereas the IDM maintains largely invariant performance, consistent with the $(\sigma_s/\lambda\sigma_a)^2$ ratio in the stochasticity term. These results confirm the predicted advantage of sparse inverse verification across all three factors, providing empirical justification for *using the IDM as a reliable verifier of the world model*.

For ***RQ2***, Figure 4 (Left) reveals a pronounced divergence in out-of-distribution generalization when data are limited. The vanilla IDM struggles on `toggle` and `swap`, whereas the sparse IDM maintains strong performance, showing that enforcing sparsity promotes more robust action inference.

4.1.2 Effectiveness for World Model Exploration

Setup. To test whether sparse inverse verification can guide exploration (***RQ3***), we first train a base world model on 200 uniformly sampled labeled transitions, then run five rounds of data acquisition, with a budget of 100 transitions per strategy per round. Figure 4 reports the prediction error averaged across all five rounds, providing an overall evaluation of performance without focusing on any particular round. We additionally report the Action Following Score, defined in Appendix E.2.3.

Results. Figure 4 (Mid, Right) shows that WAV approaches Oracle prediction performance and achieves the highest Action Following Score, indicating improvements in both prediction accuracy and action following. Across exploration rounds, WAV achieves the lowest mean prediction error among non-Oracle baselines and outperforms RLIR at all five acquisition budgets. Detailed comparisons and qualitative examples are provided in Appendices A.5 and E.2.5, respectively.

4.2 Experiments on Simulated Robot Manipulations

Datasets & Setups. We consider a set of challenging robotic manipulation tasks from two evaluation suites: RoboMimic [142] (`Lift`, `Can`, `Square`) and ManiSkill [86] (`PullCube`, `PokeCube`, `LiftPeg`). For both suites, we curate training data using expert demonstrations in a two-stage process. We first pretrain diffusion policies [20] for different numbers of training steps, yielding a diverse collection of behavior trajectories with varying levels of optimality. Based on these trajectories, we partition the data into *two subsets*: (1) *the warm-up dataset*, which includes the expert demonstrations together with on-policy trajectories collected from the best-performing diffusion policy checkpoint trained on those demonstrations; (2) *the exploration dataset*, which consists of trajectories generated by imperfect diffusion policy checkpoints, capturing diverse exploratory behaviors.

Model Choices. For the world model, we adopt Dreamer-v3 [41], which learns a latent recurrent state-space model (RSSM). For sparse IDM, we employ the model backbones from CLAM [69] and further impose sparsity on this latent action space. Details are given in Appendix E.4.

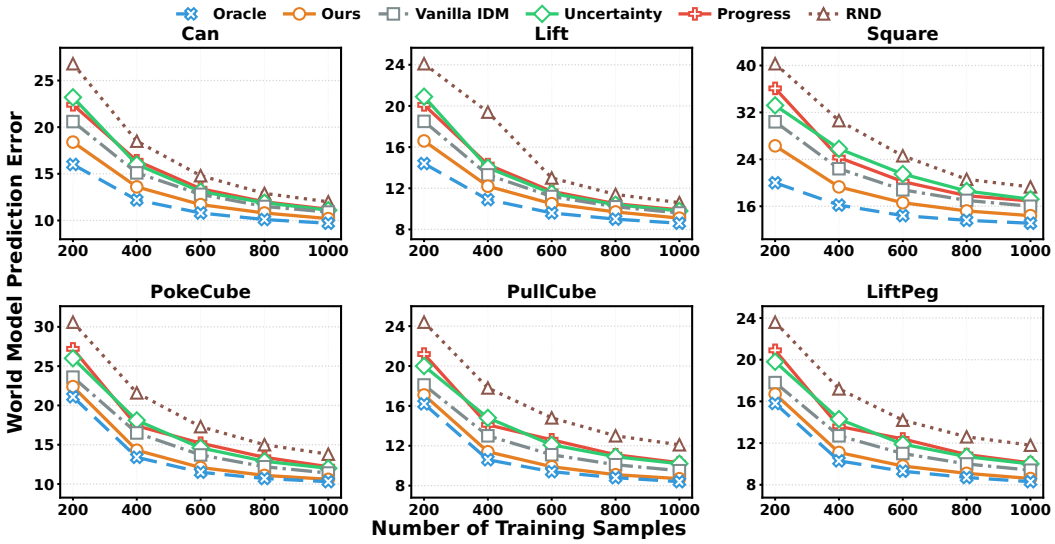

Figure 5: **World model exploration for robotic manipulation.** We report 32-frame prediction error (MSE) on RoboMimic and ManiSkill across training trajectory budgets, averaged over three seeds.

4.2.1 Effectiveness for World Model Exploration

To address ***RQ3***, we first evaluate whether WAV improves world-model learning under different data budgets in simulated robotic manipulation tasks.

Setup. We warm-start each world model using {200, 400, 600, 800, 1000} trajectories (each containing approximately 200 episodic samples) and then fine-tune it for another 200 epochs with the self-improving loop. We report the average observation MSE over 32 predicted frames after 2 exploration rounds, averaged over 3 seeds.

Results. Figure 5 shows the predictions of the world-model across different data budgets. Results under the 200-sample budget, together with standard errors, are reported in Appendix Table 1. WAV consistently outperforms all baselines, with especially large gains in the low-data regime. Across all six robotic manipulation environments, WAV requires fewer than 500 trajectories to match the prediction accuracy of the vanilla baseline trained on 1,000 trajectories, demonstrating at least a $2\times$ gain in sample efficiency. Sparse inverse dynamics models also consistently outperform dense variants on object manipulation tasks, supporting our hypothesis that sparsity improves inverse verification under limited data. The full downstream policy evaluation in Appendix A.3 and Figure 9 follow the same trend: world models improved by WAV support stronger imagination-based policy refinement than baseline world models, suggesting that the prediction gains correspond to more useful latent dynamics for control.

4.2.2 Adaptation under Distribution Shift

To address ***RQ4*** and ***RQ5***, we evaluate whether WAV improves adaptation to out-of-distribution robotic manipulation settings and whether it supports better downstream policy refinement.

Setup. We use RoboMimic `Can` as the base environment and construct two OOD variants (visualized in Appendix Figure 10). The first introduces *visual shifts*, changing nuisance factors such as background and embodiment color while keeping the task dynamics unchanged. The second introduces *object and interaction shifts*, increasing task complexity with multiple objects and demonstrations of mixed optimality. These demonstrations are collected from diffusion policy checkpoints trained for different numbers of steps, producing expert-like, medium-quality, and suboptimal behaviors.

For each OOD variant, all methods start from the same model pretrained on RoboMimic `Can`, with only action-free target-domain observations available before exploration. Each method acquires 200 labeled target-domain trajectories using its exploration criterion and fine-tunes the model under the same optimization protocol. We evaluate both world-model quality, measured by MSE on held-out OOD trajectories (with 256×256 resolutions), and downstream policy performance, measured by reward after imagination-based policy refinement under the SAILOR-based protocol [54].

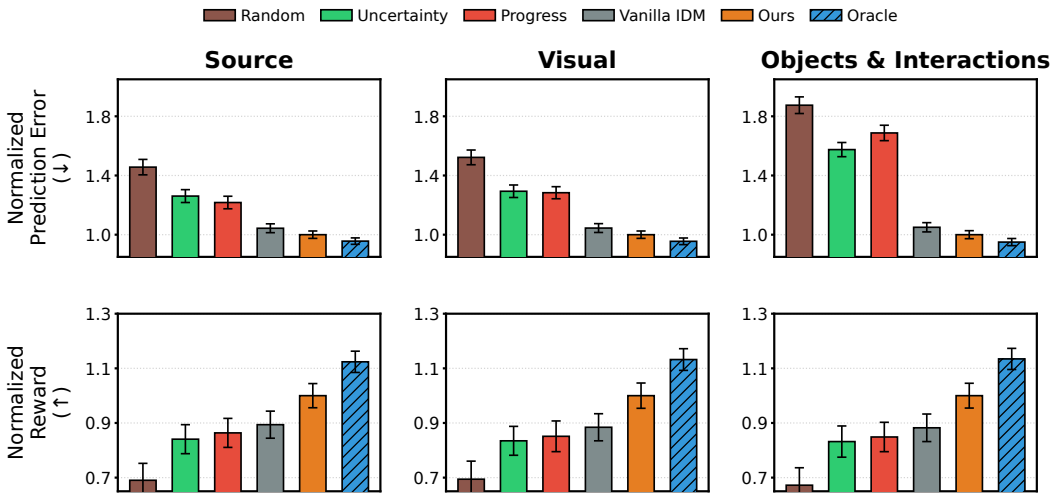

Figure 6: **Adaptation under distribution shift on RoboMimic.** We report normalized prediction error and downstream policy reward after adapting with 200 labeled trajectories per target environment. The two settings introduce visual shifts or changes in objects and interactions. Error bars show standard errors across three seeds.

Results. Figure 6 shows that WAV achieves stronger OOD adaptation than the baselines under both types of shift. Under visual shifts, where the dynamics are unchanged, but nuisance appearance factors vary, WAV better preserves action-conditioned dynamics after adaptation and achieves lower prediction error with the same 200 target trajectories. The corresponding reward results show that this improvement is not only a prediction-level gain: policies refined with WAV-adapted world models also achieve stronger downstream performance. Full in-distribution reward results across all environments are provided in Appendix A.3 and Figure 9.

The gains become larger under object and interaction shifts, where multiple objects and mixed-optimality demonstrations introduce diverse contacts, ambiguous interactions, and under-explored transition regions. In this setting, heuristic acquisition strategies may either overemphasize visually novel but less informative transitions or miss rare interaction-rich cases that are critical for learning accurate dynamics. By focusing adaptation on transitions that are difficult for the current world model but still action-reachable, WAV provides a more data-efficient mechanism for adapting world models to under-explored target regions. Overall, the combined improvements in MSE and reward suggest that WAV learns OOD dynamics that are not only more predictive, but also more useful for downstream imagination-based policy refinement, achieving 22% improvement on novel environments with new objects and interactions.

4.3 Scalability to Multitask Environments

Setup. To address ***RQ6***, we evaluate WAV using the pretrained Dreamer-v4 [42] model released with MMBench2 [43], a 427-hour, 210-task benchmark spanning 10 domains. We measure verification correlation on 1,000 training-split trajectories, sampling 100 per domain before adaptation so that all verifiers are compared on the same data. The motion-normalized baselines are inter-seed variance (u_s), measuring disagreement across denoising seeds; tokenizer residual (u_r), measuring latent inconsistency after a decode–encode round trip; and flow instability (u_f), measuring changes in clean-target predictions during late denoising steps.

We then evaluate adaptation on 10 seen and 10 unseen tasks, where seen/unseen refers to pretraining exposure. Selection methods use 50 adaptation trajectories per task; Combined uses 250 pooled from the benchmark's collection strategies. We report the action-shuffle ratio, a proxy for action following defined as one-step teacher-forced flow-prediction MSE with batch-shuffled actions divided by that with correct actions. Higher ratios indicate stronger sensitivity to action conditioning. For downstream control, closed-loop MPC optimizes 32-step action sequences with CEM and replans every 16 environment steps. Task-normalized MPC scores in $[0, 1]$ are macro-averaged within each split and normalized to Base for visualization. Appendix A.7 provides the protocols and task lists.

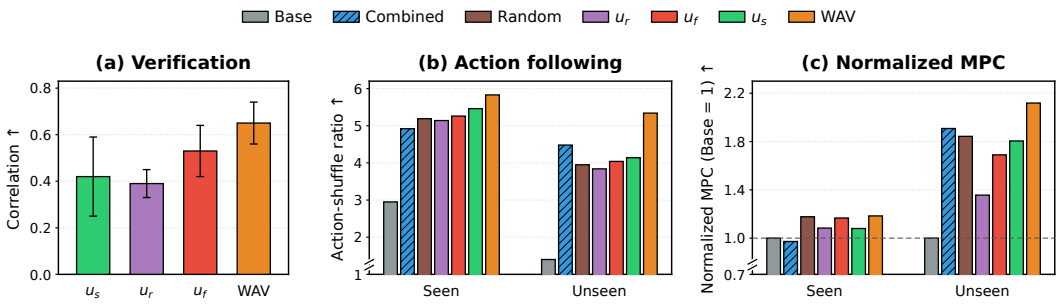

Figure 7: **Verification and data-efficient adaptation on MMBench2.** We report (a) verification correlation, (b) action following measured by the action-shuffle ratio, and (c) normalized MPC performance. Adaptation is evaluated on 10 seen and 10 unseen tasks, using 50 trajectories per task for the selection methods and 250 for Combined. MPC scores are normalized to the Base checkpoint separately within each split (Base = 1.0). Higher is better for all metrics.

Results. Figure 7(a) shows that WAV attains a verification correlation of 0.65, versus 0.53 for Flow Instability, a 22.6% relative gain. Panels (b,c) show improved action following and MPC performance on unseen tasks, while seen-task MPC is comparable to the strongest baseline. WAV outperforms Combined on both displayed metrics in both splits using only 20% of its adaptation data. Absolute scores and reconstruction and rollout PSNR are reported in Appendix Table 5.

5 Related Work

Exploration for World Models. Exploration for world models asks which interactions most improve a learned dynamics model. Classical coverage-based objectives use counts or density surrogates [7, 13, 36, 89], while model-based variants estimate value through uncertainty or disagreement [47, 101, 104], learning progress [1, 33, 58], prediction error and curiosity [27, 38, 91, 107], or goal discovery with learned world models [49, 75, 84, 137]. These signals are useful but often come from the current action-conditioned world model, making them unreliable in under-explored regimes where verification is most needed. Related work also exploits unlabeled prior data for optimistic exploration [66, 97], unsupervised skill discovery [117, 119], and mutual-information objectives [29, 137], but typically aims to learn exploration behaviors. We instead verify informative transitions through plausible future states and sparse inverse-dynamics reachability to directly enhance action-conditioned world models.

World Action Models. World action models (WAMs) have recently made substantial progress in policy learning by jointly leveraging action-free internet video and action-labeled robot data. One line jointly predicts future video and actions within a unified model, allowing large-scale action-free data to improve representations without changing the downstream policy interface [9, 15, 52, 67, 98, 120, 140]. A second line performs visual planning or foresight generation and conditions action prediction on future frames, shifting more of the planning burden into semantic image or video space while keeping low-level control at the action level [10, 14, 17, 28, 50, 55, 65, 80, 114, 126, 127, 130]. Our method is formally closer to the second family: like these approaches, it combines future-state generation with inverse action prediction. The difference is that we use this structure not as a policy model, but to expose an accuracy advantage of inverse verification over world models and to repurpose a WAM-like pipeline as a verifier for action-conditioned world models.

6 Conclusion

In this work, we identified an essential asymmetry between forward and inverse dynamics: inferring which action caused a plausible transition can be substantially easier than predicting its full outcome. Building on this insight, we introduced World Action Verifier, a self-improving framework that exploits cycle consistency among a diverse subgoal generator, a sparse inverse model, and a forward world model to gather informative interactions. Across diverse world models and robotic manipulation settings, World Action Verifier improves verification accuracy and sample efficiency, while enabling stronger downstream policy learning and closed-loop planning.

Acknowledgments

We thank the members of the IRIS Lab for valuable feedback and discussions. We also thank Xiangcheng Zhang for help with the simulation setup. This work was supported in part by the Robotics and AI Institute, DARPA, ONR, CIFAR, SNSF, Schmidt Science, NSF, NIH, Volkswagen, and the AI Institute for Societal Decision Making.

References

[1] Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning, 2017. URL `https://arxiv.org/abs/1703.01732`. 10

[2] Christopher Agia, Rohan Sinha, Jingyun Yang, Rika Antonova, Marco Pavone, Haruki Nishimura, Masha Itkina, and Jeannette Bohg. Cupid: Curating data your robot loves with influence functions. *arXiv preprint arXiv:2506.19121*, 2025. 27

[3] Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016. 26

[4] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. 1

[5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022. 26

[6] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35: 24639–24654, 2022. 26

[7] Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation, 2016. URL `https://arxiv.org/abs/1606.01868`. 10

[8] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, volume 2, page 4, 2021. 32

[9] Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. *arXiv preprint arXiv:2512.13030*, 2025. 10

[10] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023. 10, 26

[11] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 27

[12] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in neural information processing systems*, 31, 2018. 26

[13] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018. URL `https://arxiv.org/abs/1810.12894`. 10

[14] Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026. 10, 26

[15] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 10

[16] Annie S Chen, Alec M Lessing, Yuejiang Liu, and Chelsea Finn. Curating demonstrations using online experience. *arXiv preprint arXiv:2503.03707*, 2025. 27

[17] Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, Caiyi Zhang, Peihao Li, William T Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, et al. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025. 10, 26, 27

[18] Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. IGOR: Image-GOal representations are the atomic building blocks for next-level generalization in embodied AI, 2025. URL `https://openreview.net/forum?id=bpdIZTIVq8`. 26

[19] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems*, 36:73383–73394, 2023. 2

[20] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. 7

[21] Xiaowei Chi, Peidong Jia, Chun-Kai Fan, Xiaozhu Ju, Weishi Mi, Kevin Zhang, Zhiyuan Qin, Wanxin Tian, Kuangzhi Ge, Hao Li, et al. Wow: Towards a world omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025. 26

[22] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018. 26

[23] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 26

[24] William Jay Conover. *Practical nonparametric statistics*. john wiley & sons, 1999. 21

[25] Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. Robomme: Benchmarking and understanding memory for robotic generalist policies. *arXiv preprint arXiv:2603.04639*, 2026. 27

[26] Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 465–472. Omnipress, 2011. URL `https://icml.cc/2011/papers/323_icmlpaper.pdf`. 26

[27] Yilun Du, Chuang Gan, and Phillip Isola. Curious representation learning for embodied intelligence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10408–10417, 2021. 10

[28] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023. 10, 26

[29] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018. 10

[30] Yunhai Feng, Nicklas Hansen, Ziyan Xiong, Chandramouli Rajagopalan, and Xiaolong Wang. Finetuning offline world models in the real world. *arXiv preprint arXiv:2310.16029*, 2023. 26

[31] Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026. 3, 26, 27

[32] Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning. *arXiv preprint arXiv:2403.00504*, 2024. 26

[33] Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks, 2017. URL `https://arxiv.org/abs/1704.03003`. 10

[34] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation, 2025. URL `https://arxiv.org/abs/2510.10125`. 1, 26

[35] 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. *arXiv preprint arXiv:2602.12063*, 2026. 2, 3

[36] Anthony GX-Chen, Kenneth Marino, and Rob Fergus. Efficient exploration and discriminative world model learning with an object-centric abstraction, 2025. URL `https://arxiv.org/abs/2408.11816`. 10

[37] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018. 1, 26

[38] Nick Haber, Damian Mrowca, Li Fei-Fei, and Daniel L. K. Yamins. Learning to play with intrinsically-motivated self-aware agents, 2018. URL `https://arxiv.org/abs/1802.07442`. 10

[39] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019. 1, 26

[40] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019. 1

[41] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 1, 7, 26, 31

[42] Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025. 9, 26

[43] Nicklas Hansen and Xiaolong Wang. Hallucination in world models is predictable and preventable. *arXiv preprint arXiv:2606.27326*, 2026. 2, 9, 24

[44] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023. 26

[45] Nicklas A Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, pages 8387–8406. PMLR, 2022. 26

[46] Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, Pannag Sanketi, Dhruv Shah, Coline Devin, and Dorsa Sadigh. Robot data curation with mutual information estimators. *arXiv preprint arXiv:2502.08623*, 2025. 27

[47] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration, 2017. URL `https://arxiv.org/abs/1605.09674`. 3, 10

[48] Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014. 37

[49] Edward S Hu, Richard Chang, Oleh Rybkin, and Dinesh Jayaraman. Planning goals for exploration. *arXiv preprint arXiv:2303.13002*, 2023. 10

[50] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *International Conference on Machine Learning*, pages 24328–24346. PMLR, 2025. 10, 26

[51] Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025. 26

[52] Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2world: Crafting video diffusion models to interactive world models. *arXiv preprint arXiv:2505.14357*, 2025. 1, 3, 10

[53] Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation, 2026. URL `https://arxiv.org/abs/2601.03782`. 1

[54] Arnav Kumar Jain, Vibhakar Mohta, Subin Kim, Atiksh Bhardwaj, Juntao Ren, Yunhai Feng, Sanjiban Choudhury, and Gokul Swamy. A smooth sea never made a skilled sailor: Robust imitation via learning to search, 2025. URL `https://arxiv.org/abs/2506.05294`. 1, 2, 3, 8, 22, 31

[55] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025. 10, 26

[56] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019. 26

[57] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938. 29

[58] Kuno Kim, Megumi Sano, Julian De Freitas, Nick Haber, and Daniel Yamins. Active world model learning with progress curiosity, 2020. URL `https://arxiv.org/abs/2007.07853`. 2, 3, 6, 10

[59] Victor Kolev, Rafael Rafailov, Kyle Hatch, Jiajun Wu, and Chelsea Finn. Efficient imitation learning with conservative world models. In *6th Annual Learning for Dynamics & Control Conference*, pages 1777–1790. PMLR, 2024. 26

[60] Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. Robomonkey: Scaling test-time sampling and verification for vision-language-action models. *arXiv preprint arXiv:2506.17811*, 2025. 27

[61] Jacky Kwok, Xilun Zhang, Mengdi Xu, Yuejiang Liu, Azalia Mirhoseini, Chelsea Finn, and Marco Pavone. Scaling verification can be more effective than scaling policy learning for vision-language-action alignment. *arXiv preprint arXiv:2602.12281*, 2026. 27

[62] Sébastien Lachapelle. On the identifiability of latent action policies. *arXiv preprint arXiv:2510.01337*, 2025. 34, 35

[63] Sébastien Lachapelle, Pau Rodríguez López, Yash Sharma, Katie Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Nonparametric partial disentanglement via mechanism sparsity: Sparse actions, interventions and sparse temporal dependencies. *arXiv preprint arXiv:2401.04890*, 2024. 34, 35

[64] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022. 1

[65] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. 10

[66] Qiyang Li, Jason Zhang, Dibya Ghosh, Amy Zhang, and Sergey Levine. Accelerating exploration with unlabeled prior data. *Advances in Neural Information Processing Systems*, 36:67434–67458, 2023. 10

[67] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025. 10, 27

[68] Yaxuan Li, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Worldeval: World model as real-world robot policies evaluator, 2025. URL `https://arxiv.org/abs/2505.19017`. 1

[69] Anthony Liang, Pavel Czempin, Matthew Hong, Yutai Zhou, Erdem Biyik, and Stephen Tu. Clam: Continuous latent action models for robot learning from unlabeled demonstrations. *arXiv preprint arXiv:2505.04999*, 2025. 7, 32

[70] Shalev Lifshitz, Sheila A McIlraith, and Yilun Du. Multi-agent verification: Scaling test-time compute with multiple verifiers. *arXiv preprint arXiv:2502.20379*, 2025. 27

[71] Junhong Lin, Xinyue Zeng, Jie Zhu, Song Wang, Julian Shun, Jun Wu, and Dawei Zhou. Plan and budget: Effective and efficient test-time scaling on reasoning large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=ctspw4CqbS`. 27

[72] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL `https://arxiv.org/abs/2306.03310`. 27

[73] Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, et al. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025. 26

[74] Bo Liu, Chuanyang Jin, Seungone Kim, Weizhe Yuan, Wenting Zhao, Ilia Kulikov, Xian Li, Sainbayar Sukhbaatar, Jack Lanchantin, and Jason Weston. Spice: Self-play in corpus environments improves reasoning. *arXiv preprint arXiv:2510.24684*, 2025. 26

[75] Grace Liu, Michael Tang, and Benjamin Eysenbach. A single goal is all you need: Skills and exploration emerge from contrastive rl without rewards, demonstrations, or subgoals. *arXiv preprint arXiv:2408.05804*, 2024. 10

[76] Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-vla-loop: Closed-loop learning of video world model and vla policy. *arXiv preprint arXiv:2602.06508*, 2026. 2, 3

[77] Yuejiang Liu, Alexandre Alahi, Chris Russell, Max Horn, Dominik Zietlow, Bernhard Schölkopf, and Francesco Locatello. Causal triplet: An open challenge for intervention-centric causal representation learning. In *Conference on Causal Learning and Reasoning*, pages 553–573. PMLR, 2023. 4

[78] Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. *International Conference on Learning Representations (ICLR)*, 2025. URL `https://arxiv.org/abs/2408.17355`. 27

[79] Calvin Luo, Zilai Zeng, Mingxi Jia, Yilun Du, and Chen Sun. Self-adapting improvement loops for robotic learning. *arXiv preprint arXiv:2506.06658*, 2025. 26

[80] Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A vision-language-action model bridging understanding and generation to actions. *arXiv preprint arXiv:2509.06951*, 2025. 10, 26

[81] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023. 26

[82] Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. 26

[83] Zhiting Mei, Tenny Yin, Ola Shorinwa, Apurva Badithela, Zhonghe Zheng, Joseph Bruno, Madison Bland, Lihan Zha, Asher Hancock, Jaime Fernández Fisac, et al. Video generation models in robotics-applications, research challenges, future directions. *arXiv preprint arXiv:2601.07823*, 2026. 3, 26

[84] Russell Mendonca, Oleh Rybkin, Kostas Daniilidis, Danijar Hafner, and Deepak Pathak. Discovering and achieving goals via world models. *Advances in Neural Information Processing Systems*, 34:24379–24391, 2021. 10

[85] Jaouad Mourtada. Exact minimax risk for linear least squares, and the lower tail of sample covariance matrices. *The Annals of Statistics*, 50(4):2157–2178, 2022. 37

[86] Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Cathera Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 2, 7, 30

[87] Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. *arXiv preprint arXiv:2410.13816*, 2024. 27

[88] 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. *arXiv preprint arXiv:2406.02523*, 2024. 27

[89] Georg Ostrovski, Marc G. Bellemare, Aaron van den Oord, and Remi Munos. Count-based exploration with neural density models, 2017. URL `https://arxiv.org/abs/1703.01310`. 10

[90] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 26

[91] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction, 2017. URL `https://arxiv.org/abs/1705.05363`. 2, 3, 10, 26

[92] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017. URL `https://arxiv.org/abs/1709.07871`. 29

[93] Han Qi, Haocheng Yin, Aris Zhu, Yilun Du, and Heng Yang. Strengthening generative robot policies through predictive world modeling. *arXiv preprint arXiv:2502.00622*, 2025. 1

[94] Julian Quevedo, Ansh Kumar Sharma, Yixiang Sun, Varad Suryavanshi, Percy Liang, and Sherry Yang. Worldgym: World model as an environment for policy evaluation, 2025. URL `https://arxiv.org/abs/2506.00613`. 1

[95] Rafael Rafailov, Kyle Beltran Hatch, Victor Kolev, John D Martin, Mariano Phielipp, and Chelsea Finn. Moto: Offline pre-training to online fine-tuning for model-based robot learning. In *Conference on Robot Learning*, pages 3654–3671. PMLR, 2023. 26

[96] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9. 26

[97] Tabish Rashid, Bei Peng, Wendelin Boehmer, and Shimon Whiteson. Optimistic exploration even with a pessimistic initialisation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1xGP6VYwH. 10

[98] Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. Avid: Adapting video diffusion models to world models. In *Reinforcement Learning Conference*, 2025. 1, 10, 26

[99] Adhi Saravanan, Rik Knowles, Gavin Kerrigan, and Tom Rainforth. Diffbed: Scaling bayesian experimental design to high-dimensions. In *The Fourteenth International Conference on Learning Representations*, 2026. 3

[100] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. 26

[101] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR, 2020. 2, 3, 6, 10

[102] Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, et al. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026. 1, 3, 26

[103] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023. 26

[104] Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration, 2019. URL https://arxiv.org/abs/1810.12162. 10

[105] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017. 26

[106] Charles Spearman. The proof and measurement of association between two things. 1961. 21, 28

[107] Bradly C. Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models, 2015. URL https://arxiv.org/abs/1507.00814. 10

[108] Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In Bruce W. Porter and Raymond J. Mooney, editors, *Machine Learning, Proceedings of the Seventh International Conference on Machine Learning, Austin, Texas, USA, June 21-23, 1990*, pages 216–224. Morgan Kaufmann, 1990. doi: 10.1016/B978-1-55860-141-3.50030-4. URL https://doi.org/10.1016/b978-1-55860-141-3.50030-4. 26

[109] Gemini Robotics Team, Krzysztof Choromanski, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Isabel Leal, Fangchen Liu, Anirudha Majumdar, Andrew Marmon, Carolina Parada, Yulia Rubanova, Dhruv Shah, Vikas Sindhwani, Jie Tan, Fei Xia, Ted Xiao, Sherry Yang, Wenhao Yu, and Allan Zhou. Evaluating gemini robotics policies in a veo world simulator, 2026. URL https://arxiv.org/abs/2512.10675. 1

[110] PAN Team, Jiannan Xiang, Yi Gu, Zihan Liu, Zeyu Feng, Qiyue Gao, Yiyan Hu, Benhao Huang, Guangyi Liu, Yichi Yang, Kun Zhou, Davit Abrahamyan, Arif Ahmad, Ganesh Bannur, Junrong Chen, Kimi Chen, Mingkai Deng, Ruobing Han, Xinqi Huang, Haoqiang Kang, Zheqi Liu, Enze Ma, Hector Ren, Yashowardhan Shinde, Rohan Shingre, Ramsundar Tanikella, Kaiming Tao, Dequan Yang, Xinle Yu, Cong Zeng, Binglin Zhou, Zhengzhong Liu, Zhiting Hu, and Eric P. Xing. Pan: A world model for general, interactable, and long-horizon world simulation, 2025. URL https://arxiv.org/abs/2511.09057. 1

[111] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=meRCKuUpmc. 26, 27

[112] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4950–4957. International Joint Conferences on Artificial Intelligence Organization, 2018. 26

[113] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL https://arxiv.org/abs/1711.00937. 29

[114] 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 via skill-compositional world models. *arXiv preprint arXiv:2603.10422*, 2026. 10

[115] Pinzheng Wang, Juntao Li, Zecheng Tang, Haijia Gui, et al. Improving rationality in the reasoning process of language models through self-playing game. *arXiv preprint arXiv:2506.22920*, 2025. 26

[116] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508, 2023. 26

[117] Zizhao Wang, Jiaheng Hu, Caleb Chuck, Stephen Chen, Roberto Martín-Martín, Amy Zhang, Scott Niekum, and Peter Stone. Skild: Unsupervised skill discovery guided by factor interactions. *Advances in Neural Information Processing Systems*, 37:77748–77776, 2024. 10

[118] Zizhao Wang, Kaixin Wang, Li Zhao, Peter Stone, and Jiang Bian. Dyn-o: Building structured world models with object-centric representations. *arXiv preprint arXiv:2507.03298*, 2025. 4

[119] Max Wilcoxson, Qiyang Li, Kevin Frans, and Sergey Levine. Leveraging skills from unlabeled prior data for efficient online exploration. *arXiv preprint arXiv:2410.18076*, 2024. 10

[120] Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2024. 10, 26

[121] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning, 2022. URL https://arxiv.org/abs/2206.14176. 1

[122] Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. *Advances in neural information processing systems*, 32, 2019. 26

[123] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=NpU7ZXafRi. 27

[124] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1 (2):6, 2023. 1

[125] Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators, 2024. URL `https://arxiv.org/abs/2310.06114`. 1

[126] Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026. 10

[127] Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. 1, 10, 26, 27

[128] Yang Ye, Tianyu He, Shuo Yang, and Jiang Bian. Reinforcement learning with inverse rewards for world model post-training. *arXiv preprint arXiv:2509.23958*, 2025. 4, 6, 23, 26

[129] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021. 26

[130] Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. 10

[131] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022. 26

[132] Hangfan Zhang, Siyuan Xu, Zhimeng Guo, Huaisheng Zhu, Shicheng Liu, Xinrun Wang, Qiaosheng Zhang, Yang Chen, Peng Ye, Lei Bai, et al. The path of self-evolving large language models: Achieving data-efficient learning via intrinsic feedback. *arXiv preprint arXiv:2510.02752*, 2025. 26

[133] Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*, 2024. 26

[134] Qin Zhang, Peiyu Jing, Hong-Xing Yu, Fangqiang Ding, Fan Nie, Weimin Wang, Yilun Du, James Zou, Jiajun Wu, and Bing Shuai. Physion-eval: Evaluating physical realism in generated video via human reasoning. *arXiv preprint arXiv:2603.19607*, 2026. 4

[135] Xiangcheng Zhang, Haowei Lin, Haotian Ye, James Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. *arXiv preprint arXiv:2505.23614*, 2025. 27

[136] Zhilong Zhang, Ruifeng Chen, Junyin Ye, Yihao Sun, Pengyuan Wang, Jingcheng Pang, Kaiyuan Li, Tianshuo Liu, Haoxin Lin, Yang Yu, et al. Whale: Towards generalizable and scalable world models for embodied decision-making. *arXiv preprint arXiv:2411.05619*, 2024. 1

[137] Chongyi Zheng, Jens Tuyls, Joanne Peng, and Benjamin Eysenbach. Can a misl fly? analysis and ingredients for mutual information skill learning. *arXiv preprint arXiv:2412.08021*, 2024. 10

[138] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. Flare: Robot learning with implicit world modeling, 2025. URL `https://arxiv.org/abs/2505.15659`. 1

[139] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning, 2025. URL https://arxiv.org/abs/2411.04983. 1, 26

[140] 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. 10, 27

[141] Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation, 2025. URL https://arxiv.org/abs/2406.14540. 1

[142] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020. 2, 7, 30

[143] Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB. 26

Appendix Contents

A Additional Experiments **21**

B Additional Related Work **26**

C Additional Theoretical Analysis **26**

D Additional Discussions **27**

E Additional Implementation Details **27**

F Additional Theoretical Derivations **33**

G Broader Impact **38**

H NeurIPS Paper Checklist **43**

A Additional Experiments

A.1 Robustness of World Action Verification

Setup. Following the MiniGrid evaluation, we assess the robustness of WAV by computing Spearman's rank correlation [106] between each method's data-selection scores and those of the Oracle. We use 100 samples held out from world-model training.

Results. As shown in Figure 8, WAV's verification scores more faithfully recover the Oracle ranking of sample difficulty, demonstrating robust verification across the robotic tasks.

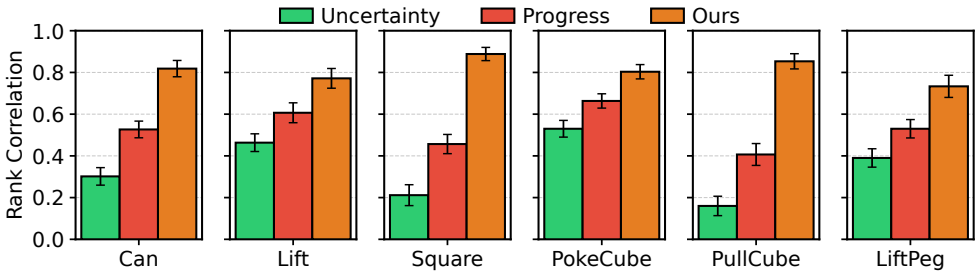

Figure 8: **Robustness of WAV verification on RoboMimic and ManiSkill.** We evaluate how well each method ranks informative samples by computing the Spearman correlation between its scores and Oracle scores. Higher correlation indicates closer agreement with the Oracle ranking.

A.2 Full Results on World Model Prediction

Table 1 reports the prediction error under the 200-sample budget. WAV consistently achieves the lowest error among all non-Oracle methods across the six tasks, suggesting that inverse-verification-based data selection provides a more effective adaptation signal than uncertainty-, progress-, or novelty-based acquisition. The improvement is especially clear relative to non-IDM baselines: WAV performs significantly better across the tasks under a Wilcoxon signed-rank test [24] at the 5% level. Compared with Vanilla IDM, Ours also further reduces prediction error, indicating that the verifier is not only useful as an inverse-dynamics filter, but also helps select transitions that are more informative for action-conditioned world-model adaptation.

Table 1: World model prediction error with 200 training samples. Results are reported as mean $\pm$ standard error. The best non-Oracle result for each task is highlighted in light yellow. † denotes a significant difference at the 5% level between the best non-Oracle method and the second-best method excluding other IDM variants.

Method	Can	Lift	Square	PokeCube	PullCube	LiftPeg
Random	26.5 ± 0.7	23.3 ± 0.8	39.7 ± 0.8	30.1 ± 0.9	24.6 ± 0.8	23.4 ± 0.7
Uncertainty	22.9 ± 0.6	20.7 ± 0.6	33.0 ± 0.7	25.2 ± 0.7	19.9 ± 0.5	19.4 ± 0.6
Progress	22.0 ± 0.6	19.6 ± 0.6	35.1 ± 0.7	27.3 ± 0.7	21.1 ± 0.6	20.5 ± 0.8
Vanilla IDM	20.2 ± 0.2	18.3 ± 0.5	29.8 ± 0.7	23.4 ± 0.6	18.2 ± 0.5	17.5 ± 0.5
Ours	$18.2 \pm 0.5^\dagger$	$16.4 \pm 0.4^\dagger$	$26.0 \pm 0.3^\dagger$	$22.4 \pm 0.5^\dagger$	$17.1 \pm 0.4^\dagger$	$16.8 \pm 0.5^\dagger$

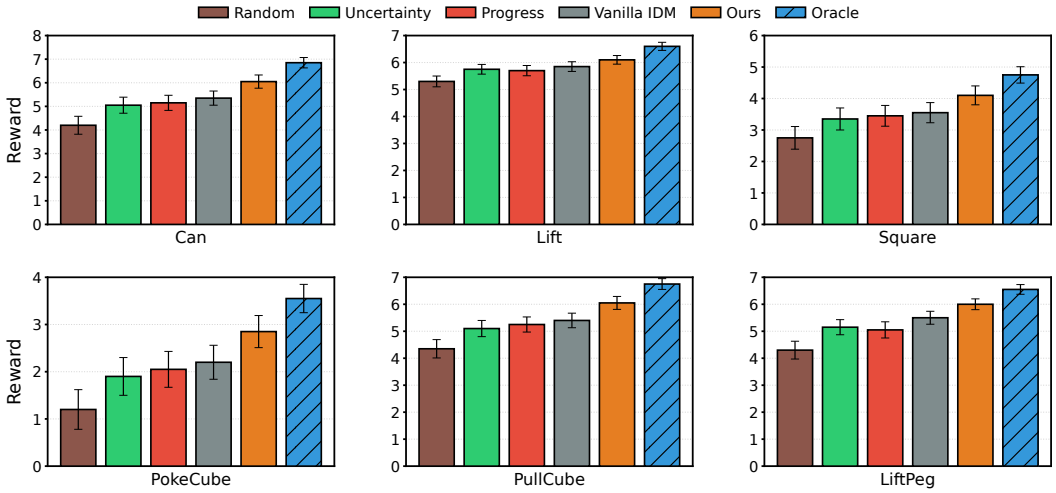

Figure 9: **Downstream policy performance on RoboMimic and ManiSkill using learned world models.** Error bars denote the standard error over three seeds.

A.3 Full Results on Policy Learning

We provide the full downstream policy learning results in Figure 9. For each learned world model, we refine the same base diffusion policy through imagination-based search following the SAILOR protocol [54], and report the resulting task reward after fine-tuning with a fixed data budget of 1,000 trajectories. Across RoboMimic and ManiSkill tasks, policies refined with WAV-based world models achieve higher rewards than those refined with baseline world models, and are second only to the oracle model that uses privileged ground-truth actions for sample selection. The improvement is most pronounced on tasks with ambiguous or contact-rich dynamics, such as `Can`, `Square`, and `PokeCube`, where accurate action-conditioned latent dynamics are important for effective imagination-based policy refinement. In contrast, simpler tasks such as `Lift` show smaller gaps across methods, suggesting that downstream policy gains are most sensitive to world-model quality when the task requires resolving more complex object interactions.

A.4 OOD Setups

We construct two out-of-distribution variants of the RoboMimic `Can` task, visualized in Figure 10. The first variant introduces visual appearance shifts by changing nuisance rendering factors, including the background and embodiment color, while keeping the underlying task dynamics unchanged. The second variant introduces object and interaction shifts by adding multiple objects and collecting demonstrations from diffusion policy checkpoints with different training progress, resulting in a mixture of expert-like, medium-quality, and suboptimal behaviors. For each OOD variant, all methods initialize from the same world model trained on the original `Can` environment. Target-domain action-free observations are available before exploration, but target-domain action labels are not. Each method receives the same budget of 200 labeled target-domain trajectories, collects them according to its exploration criterion, and then fine-tunes the world model using the same optimization schedule. We evaluate both 32-step next-observation prediction error on held-out OOD trajectories

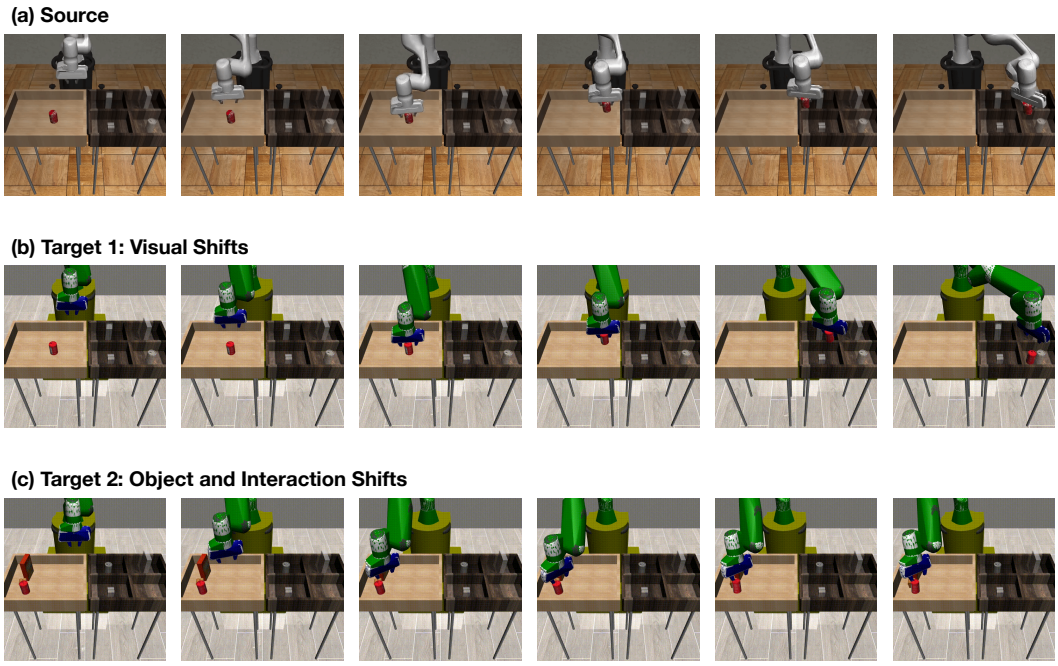

Figure 10: **Visualization of the OOD adaptation settings.** Starting from the original RoboMimic `Can` environment, we construct two categories of target-domain shifts. The first introduces visual shifts, including modified backgrounds and embodiment colors, while preserving the same task dynamics. The second introduces more challenging object and interaction shifts, including scenes with multiple objects and demonstrations collected from policies with different levels of optimality. These settings test whether the learned world model can adapt to both visual nuisance changes and harder dynamics-relevant distribution shifts with limited target-domain data.

Table 2: World-model prediction error averaged over all MiniGrid exploration rounds. Lower is better.

Method	**Average error** $\downarrow$
Random	1.504
Uncertainty	1.299
RLIR	1.258
Progress	1.172
WAV	**1.103**

and downstream reward after imagination-based policy refinement under the same SAILOR-style protocol used in the in-distribution robotic experiments.

A.5 Complete Exploration-Round Results and the RLIR Baseline

The main MiniGrid analysis uses an early exploration round as a diagnostic of low-budget behavior. To verify that the conclusion does not depend on that choice, we additionally average the world-model prediction error over all exploration rounds. Table 2 shows that the ordering remains unchanged: WAV obtains the lowest average error, followed by Progress, RLIR [128], Uncertainty, and Random.

We further compare RLIR and WAV at acquisition budgets of 50, 100, and 150 interactions per round. As shown in Table 3, RLIR is nearly indistinguishable from Random at the smallest budget and improves as the forward model becomes more accurate. WAV consistently performs best, supporting our motivation that methods initialized by a forward rollout are less reliable in under-explored regimes.

Table 3: MiniGrid prediction error under different per-round acquisition budgets. Lower is better.

Method	50	100	150
Random	2.355	1.890	1.467
RLIR	2.351	1.718	1.140
WAV	**1.992**	**1.241**	**0.881**

Table 4: Verification correlation on RoboMimic `Square` under different subgoal qualities. Higher is better.

Method	Verification correlation ↑
Uncertainty	0.21
Progress	0.44
WAV + early-stage generator	0.54
WAV + mid-stage generator	0.65
WAV + clean subgoals	**0.82**

A.6 Sensitivity to Subgoal Quality

An infeasible subgoal can cause WAV to overestimate forward-model error because even an accurate world model cannot realize it. We test this failure mode on RoboMimic `Square` using weaker DreamZero-5B subgoal generators obtained after 2,500 and 5,000 of the full 10,000 MimicGen fine-tuning steps. Although verification correlation decreases with generator quality, both learned-generator variants continue to outperform the uncertainty and progress baselines (Table 4).

A.7 MMBench-2 Evaluation Protocol

This section provides implementation and evaluation details for the MMBench-2 [43] experiments in Section 4.3. The aggregate comparison is reported in Figure 7 in the main text.

Benchmark and models. We evaluate on 20 MMBench2 tasks, comprising 10 seen and 10 unseen tasks under the benchmark's pretraining split.[2] *Unseen* refers to pretraining exposure; task-specific adaptation trajectories are provided during our experiments. We use the released Base, Coverage-aware, and Combined checkpoints as reference models. All released checkpoints are reevaluated using our common evaluation protocol.

Self-Improving setup and evaluation. For controlled comparisons, Random and the normalized u_r, u_f, and u_s predictors select 50 trajectories per task from the same 250-trajectory pool. These are *offline selection baselines*, distinct from the original online curiosity policies. Local Combined uses all 250 trajectories per task and therefore has a larger data budget.

Following the original adaptation setting in MMBench2, we use the following tasks for evaluation:
Seen tasks

- **DMControl:** `cup-catch`, `finger-turn-easy`.
- **Meta-World:** `mw-push`.
- **ManiSkill:** `ms-push-cube`.
- **Box2D:** `lunarlander-hover`.
- **OGBench:** `og-point-maze`, `og-point-bottleneck`.
- **MiniArcade:** `pygame-point-maze-var1`, `pygame-pong`, `pygame-bird-attack`.

Unseen tasks

- **DMControl:** `cup-catch-var1`, `finger-turn-easy-var1`.
- **ManiSkill:** `ms-push-banana`.
- **OGBench:** `og-point-var1`, `og-point-var2`.
- **MiniArcade:** `pygame-point-maze-var4`, `pygame-reacher-easy`, `pygame-dungeon-explorer1`, `pygame-foraging`, `pygame-whirlpool`.

Local full-budget adaptation starts from the released Coverage-aware checkpoint and uses 50,000 tokenizer updates followed by 30,000 dynamics updates, with learning rate 10^{-4} and global batch

[2] MMBench2 code and released models: `https://github.com/nicklashansen/mmbench2`.

Table 5: Results on the Seen (10 tasks) and Unseen (10 tasks) splits of MMBench2. All metrics are reported as absolute values.

Method	Adaptation traj./task	Seen (10 tasks)				Unseen (10 tasks)			
		Recon PSNR ↑	Rollout PSNR ↑	Action shuf. ↑	Task perf. (MPC) ↑	Recon PSNR ↑	Rollout PSNR ↑	Action shuf. ↑	Task perf. (MPC) ↑
Base	0	37.04	28.54	2.95	0.288	17.38	15.50	1.40	0.261
Combined	250	**38.68**	29.85	4.92	0.280	37.90	29.94	4.48	0.498
Random	50	38.15	30.05	5.19	0.339	37.59	29.70	3.95	0.481
u_r	50	38.16	30.00	5.14	0.312	37.41	29.47	3.84	0.354
u_f	50	38.43	30.13	5.26	0.336	37.44	29.63	4.04	0.441
u_s	50	38.23	30.38	5.46	0.311	37.19	30.00	4.14	0.471
WAV	50	38.43	**30.44**	**5.83**	**0.341**	**38.15**	**30.59**	**5.34**	**0.553**

Table 6: Closed-loop MiniGrid planning success rate. Relative gains are measured against Random.

Method	Success rate ↑	Relative gain
Random	56.25%	—
Progress	57.50%	$+2.2\%$
Uncertainty	63.75%	$+13.3\%$
WAV	$\mathbf{72.50}\%$	$+\mathbf{28.9}\%$

sizes of 512. Background replay is retained, excluding three unavailable RGB partitions. For WAV, we use target-task sampling probabilities $\alpha = 0.75$, where α controls target-task sampling probability rather than directly specifying the fraction of newly collected transitions.

We measure reconstruction quality (PSNR), future-frame prediction error (PSNR, LPIPS), action sensitivity (Action Shuffle Score), and downstream model predictive control (MPC) performance. Specifically, prediction evaluation uses 120 windows from 40 held-out episodes per task, with eight context frames, a 16-step prediction horizon, and eight denoising steps. Action sensitivity measures the increase in latent prediction error after shuffling actions within a task. MPC scores are averaged over three full evaluation episodes per task. Results are macro-averaged across tasks.

Details on WAV. WAV searches for action sequences whose imagined consequences are inconsistent with the IDM outpus. Given a candidate action a_t and the corresponding imagined transition, we define e_t as the WAV score at time t. We optimize the sequence-level objective

$$J(a_{t:t+H-1}) = \frac{1}{2H} \sum_{k=0}^{H-1} e_{t+k} + \frac{1}{2} \max_{0 \le k < H} e_{t+k}. \tag{11}$$

The cross-entropy method (CEM) uses 128 candidate sequences, 32 elites, three iterations, and a planning horizon of 32 steps. We execute the first 16 actions before replanning. Both the world model and IDM remain frozen during collection. The collected dataset contains *actual environment observations, executed actions, and rewards* and IDM predictions are used only for exploration scoring.

A.8 Full Results on MMBench2 Adaptation Experiments

Table 5 gives the full quantitative results on adaptation experiments discussed in Section 4.3, including reconstruction PSNR, rollout PSNR, action-shuffle ratio, and normalized MPC task performance.

A.9 Closed-Loop Planning

We additionally test whether improvements in world-model accuracy transfer to planning rather than only policy learning. Following the MiniGrid setup, a closed-loop planner evaluates candidate action sequences over a horizon of three steps, executes the first action of the sequence whose predicted terminal state is closest to the target, observes the resulting state, and replans. WAV improves the success rate from 56.25% under random exploration to 72.50%, a 28.9% relative gain (Table 6).

B Additional Related Work

World Models for Robotics. Model-based reinforcement learning (MBRL) learns predictive environment dynamics for planning and policy improvement. Early approaches focused on probabilistic, data-efficient control [26, 108], while modern deep MBRL combines expressive dynamics models with planning and imagined rollouts in off-policy learning [22, 56]. By reasoning over predicted futures, these methods can be highly sample-efficient and effective for control. However, they are often tied to specific tasks or policy distributions, which limits their robustness under distribution shifts. To mitigate this, prior work has explored model ensembles [12, 56], conservative optimization [59, 129], and online fine-tuning initialized from offline priors [30, 95]. Nevertheless, these approaches still remain limited in their scalability to high-dimensional perception, diverse interaction regimes, and broad generalization.

More recently, general-purpose world models that learn predictive representations from large and diverse sequential data have been developed. One line of work focuses on *latent world models*, which learn compact action-conditioned dynamics from high-dimensional observations and perform prediction, imagination, and control in a learned latent space [32, 37, 39–41, 82, 139, 143]. A second line is more *planning- and control-centric*, with objectives that are directly for decision making and policy improvement [44, 45, 100]. A third line studies *pixel-based* world models trained on large-scale robotics or even internet video data [17, 98, 120], either by combining video generation with inverse dynamics models [21] or by directly learning action-conditioned dynamics [31, 34, 42], with the goal of producing visually realistic yet controllable futures. A key advantage of these models is that they can leverage internet-scale data and increasingly scalable model sizes to acquire broad predictive priors. Despite their strong generative capacity, these models still face important challenges in physical consistency and action following [83, 102], especially when deployed for robotics control. Our work addresses this through targeted exploration, which actively collects informative interactions with verification to improve the robustness of action-conditioned world models.

Inverse Dynamics on Videos. A common strategy for leveraging action-free observations is to infer actions from state transitions. Existing work uses inverse dynamics in different roles in world model and policy learning. As *policies*, IDMs convert predicted or planned future observations into executable actions, as in visual-planning and foresight-based systems [10, 14, 18, 28, 50, 79, 80, 111, 127]. As *regularizers*, inverse objectives have long been used to shape self-supervised representations and dynamics features [3, 91]. As *labelers*, inverse models can impute missing actions from state-only or video demonstrations [6, 55, 112, 122]. As *verifiers*, they can test whether a transition is action-consistent; conceptually, our approach is closest to this line and to RLIR [128]. However, our verifier differs in two key ways: it uses a reverse cycle anchored on plausible future states, and it checks reachability with a sparse action-relevant inverse model rather than dense full-state generation.

Self-improvement via verification. A parallel line of work studies self-improvement loops driven by internally generated feedback, especially in language models and self-play agents. In alignment, solver-verifier pipelines optimize policies against learned feedback signals [5, 90, 96]; explicit verifiers have also been used to filter and rerank candidate solutions in domains like math reasoning [23, 133]. Complementary approaches bootstrap supervision by generating candidate solutions and selectively adding verified/high-quality outputs back into training [116, 131], as well as test-time self-critique and refinement loops [81, 103]. Finally, self-play provides a principled mechanism to create increasingly challenging data and feedback [51, 73, 74, 105, 115, 132]. Our work shares the idea of using a verifiable internal signal to turn novel experience into training data. While self-improvement is well-studied for language models, *few works explore it for world models*, where verification is harder due to continuous dynamics and absent symbolic ground truth. WAV addresses this gap with a video-prior subgoal generator and sparse inverse-dynamics cycle consistency as a reachability verifier.

C Additional Theoretical Analysis

In this section, we analyze the exact conditions under which the sparse inverse verifier can generalize beyond the limited training distribution of labeled transitions. To analyze this, we model the observed state s^t as arising from a latent vector $\mathbf{z}^t = (\mathbf{z}_1^t, \ldots, \mathbf{z}_k^t)$. The learned mask M in (5) selects an action-relevant block $\mathcal{S}$ of this latent space; intuitively, $\mathcal{S}$ captures agent-centric variables (*e.g.*, proprioception or end-effector motion) and is largely insulated from the rest of the scene. The sparse

inverse model h_ψ from Sec. 2 thus operates on $(\mathbf{z}_{\mathcal{S}}^t, \mathbf{z}_{\mathcal{S}}^{t+1})$; we write the verifier as $\hat{\mathbf{a}}^t = h_\psi(\hat{\mathbf{z}}_{\mathcal{S}}^t, \hat{\mathbf{z}}_{\mathcal{S}}^{t+1})$, where $\hat{\mathbf{z}}^t$ denotes the encoder's latent estimate of $\mathbf{z}^t$.

Let P_{seed} denote the distribution induced by $\mathcal{D}_{\text{act}}$; we call a state–action pair *out-of-support* (OOS) when $(\mathbf{z}^t, \mathbf{a}^t) \notin \text{supp}(P_{\text{seed}})$. The key structural condition is the presence of a *generation–verification gap*: the full pair $(\mathbf{z}^t, \mathbf{a}^t)$ may be OOS, while the restricted pair $(\mathbf{z}_{\mathcal{S}}^t, \mathbf{a}^t)$ remains on support. This captures the regime in which scene-level consequences are novel, but the agent-side motion that encodes the action is still familiar.

Proposition C.1 (Informal)**.** *Assume there exists an identifiable verification subset $\mathcal{S}$ such that: (i) $\mathbf{z}_{\mathcal{S}}^{t+1}$ depends only on $(\mathbf{z}_{\mathcal{S}}^t, \mathbf{a}^t)$ and not on the rest of the scene; (ii) $(\mathbf{z}_{\mathcal{S}}^t, \mathbf{a}^t)$ stays on-support even when $(\mathbf{z}^t, \mathbf{a}^t)$ is OOS; and (iii) the action is identifiable from the subset transition $(\mathbf{z}_{\mathcal{S}}^t, \mathbf{z}_{\mathcal{S}}^{t+1})$. Then an inverse model trained on the seed data can recover the correct action from $(\hat{\mathbf{z}}_{\mathcal{S}}^t, \hat{\mathbf{z}}_{\mathcal{S}}^{t+1})$ on such compositional OOS transitions. Consequently, the forward–inverse mismatch used by WAV localizes forward-model error rather than action-label ambiguity.*

Interpretation. Proposition C.1 guarantees that the sparse inverse model h_ψ produces correct pseudo-labels whenever the agent's own motion pattern (*e.g.*, joint-angle trajectories) was seen during training, even if the full scene transition is novel. This is a strictly weaker requirement than asking the *full* transition to be on-support, which is what a dense forward model or a full-observation inverse model would need. The direct consequence for the self-improving cycle in Sec. 2.3 is that the discrepancy $\ell(\tilde{s}^{t+1}, \hat{s}^{t+1})$ between the subgoal and the forward rollout reflects genuine world-model error rather than action-label noise, so each exploration round adds trustworthy data that expands the world model's effective coverage. Appendix F.1 formalizes this claim and shows how the verification subset $\mathcal{S}$ can be identified from observations.

D Additional Discussions

Although our primary focus is verification-guided exploration for acquiring new interactions, WAV may also be useful in settings that benefit from robust error estimation, such as test-time scaling [60, 78, 87] and offline data curation [2, 16, 46]. WAV uses two forward passes and one inverse pass, whereas progress-based methods typically use paired forward models and ensemble uncertainty can require several forward passes. The inverse model is also substantially smaller than the world model. Nevertheless, improving efficiency through shared intermediate representations [67, 140] or adaptive computation [71, 123, 135] remains important for real-time deployment.

More broadly, our results suggest that the forward–inverse asymmetry may be especially pronounced in high-dimensional, uncertain environments. However, the current exploration and control experiments are limited to simulation, and MMBench-2 is an offline benchmark rather than a physical deployment. Autonomous real-robot exploration would additionally require reliable resets, explicit safety constraints, and robustness to sensing and actuation latency. Fully self-improving world models in such settings therefore remain elusive. Unlike language models, which can already improve from purely synthetic data on some reasoning tasks, our method still relies critically on additional environment feedback to correct action-conditioned prediction errors. Reducing this reliance will likely require substantially stronger verification mechanisms [61, 70], likely scaffolding on more capable pretrained models. Extending World Action Verifier with richer generative priors [17, 31] and more expressive inverse models [111, 127] in longer-horizon embodied tasks [11, 25, 72, 88] is a promising direction for future work.

E Additional Implementation Details

E.1 Compute Resource

For the MiniGrid experiments, we utilized $1\times$ NVIDIA RTX 4090 GPU, with each full experimental run requiring approximately 2 to 3 GPU hours. This duration encompasses world model training, verifier training and active learning iterations. For the robotic manipulation experiments, we used either $6\times$ NVIDIA L40S GPUs or $3\times$ NVIDIA H100 GPUs across all experiments. On average, WAV required approximately 40 GPU hours per robotic environment, while the baselines required approximately 36 GPU hours under comparable settings. These include world-model training, verifier training, and downstream imagination-based policy refinement.

E.2 MiniGrid Setting

We conduct experiments in the MiniGrid simulation environment, using `EmptyEnv` with three object types: key, ball, and box, each of which can be either red or blue. The agent has seven discrete actions: *turn left, turn right, move forward, pick up, drop, toggle, and swap.* The behavior of the toggle action is object-dependent: for keys and balls, it switches the object's color; for boxes, it acts as an exchange mechanism, swapping the item currently held by the agent with the item inside the box (or placing the held item inside if the box is empty). The swap action is not part of the original EmptyEnv. We define it as exchanging the object in front of the agent with the object it is carrying. Based on this setup, we designed three tasks in EmptyEnv, as detailed in Sec. E.2.1.

E.2.1 Task Definitions

To evaluate the agent's ability to handle long-horizon dependencies and compositional logic, we design three complex tasks in the MiniGrid environment. Each task requires the agent to manipulate objects (Key, Ball, Box) based on their attributes (Red, Blue).

- **Task 1: Key Delivery.** The agent must: (1) locate a key, (2) change its color to match the target box, (3) place the key inside the box, (4) swap the box with a ball, (5) adjust the ball's color to match the box, and (6) reach the goal.
- **Task 2: Ball Delivery.** This is a structural mirror of Task 1 but swaps the roles of the key and the ball. The agent must place the ball inside the box before manipulating the key and reaching the goal.
- **Task 3: Object Matching.** The agent must: (1) identify the reference color of the box, (2) locate the key and ball, (3) synchronize the key's color with the box, (4) synchronize the ball's color with the box, (5) place both the key and the ball around the box, and (6) reach the goal.

E.2.2 Dataset Composition.

Random Play Dataset. We construct random play datasets based on the `EmptyEnv`, where objects can be freely placed. To study state complexity, we vary the number of objects $\{6, 8, 10, 12, 14\}$ and collect trajectories using random policies, resulting in environments with increasingly complex object configurations. In this setting, only the environment with 6 objects is used to construct both the training and test sets, while environments with higher object counts are used exclusively for testing, enabling controlled evaluation of generalization to more complex scenes.

To study environmental stochasticity, we vary the number of noisy floor tiles $\{0, 1, 2, 3, 4\}$, whose colors change randomly at every step. For each noise level, we collect trajectories with random policies and construct both training and test sets, allowing us to evaluate robustness under different levels of environmental noise.

Exploration Pool. We collect approximately 50,000 transitions across the three tasks defined in Sec. E.2.1, covering diverse state–action–next-state tuples. Roughly half of these transitions form an unlabeled pretraining set for the video model. The remainder supplies the exploration pool, including an action-labeled seed set of 200 transitions and an unlabeled candidate set of 20,000 transitions from which the acquisition strategies iteratively select informative samples. We additionally construct an action-balanced test set of 10,368 transitions for evaluation.

E.2.3 Evaluation Metrics.

Prediction Accuracy. We first report Dynamics Accuracy, which measures prediction accuracy only over elements that undergo temporal changes, including both visual grid cells and internal agent attributes (e.g., carried status), while masking out invariant background regions. By focusing on these dynamic components, Dynamics Accuracy mitigates metric inflation caused by static background dominance and better reflects the model's ability to capture action-driven dynamics.

In exploration experiments, we further evaluate all methods using the world model's next-state prediction loss on the held-out test set. We utilize prediction loss in this context because it provides a more sensitive signal of training stability and convergence behavior during exploration, whereas Dynamics Accuracy is better suited for interpreting final predictive performance.

Ranking Quality. To assess the quality of data selection, we compute the rank correlation between method-assigned scores and Oracle scores using Spearman's rank correlation coefficient (ρ) [106]

and Kendall's rank correlation coefficient (τ) [57]. Given a set of samples with scores $\{s_i\}$ and corresponding Oracle scores $\{o_i\}$, the Spearman correlation is defined as

$$\rho = 1 - \frac{6\sum_i (r_i - q_i)^2}{n(n^2-1)}, \tag{12}$$

where r_i and q_i denote the ranks of s_i and o_i, respectively.

Kendall's τ measures the consistency of pairwise orderings:

$$\tau = \frac{N_c - N_d}{\frac{1}{2}n(n-1)}, \tag{13}$$

where N_c and N_d are the numbers of concordant and discordant pairs. Higher values indicate stronger agreement with the Oracle ranking.

Action Following Score. In addition to prediction accuracy and ranking quality, we evaluate whether the learned world model can capture action-dependent dynamics. The *Action Following Score* (AFS) measures how well the model preserves distinctions between different actions in its predicted future states.

For each evaluation state s_0, we consider pick-up, toggle, and swap. Let $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ denote the oracle and predicted next-state attribute vectors for action a_i, including grid object/color/state channels and carried-object type/color. Predictions are discretized using argmax for grid channels and rounding for carried-object attributes.

Define the differing attributes as $\mathcal{D}_{ij} = \{k : x_{i,k} \neq x_{j,k}\}$, and the valid action pairs as $\mathcal{P}(s_0) = \{(i,j) : i < j, \mathcal{D}_{ij} \neq \varnothing\}$. We compute

$$\mathrm{AFS}(s_0) = \frac{1}{|\mathcal{P}(s_0)|} \sum_{(i,j)\in\mathcal{P}(s_0)} \prod_{k\in\mathcal{D}_{ij}} \mathbf{1}[\hat{x}_{i,k} = x_{i,k} \wedge \hat{x}_{j,k} = x_{j,k}]\,. \tag{14}$$

Thus, a pair scores one only if both predictions exactly match their oracle outcomes on every differing attribute. We average these state-level scores uniformly over evaluation states, all of which contain at least one valid pair. AFS lies in $[0, 1]$, with higher values indicating more accurate action-dependent predictions. For each aggregate panel, we average the round-level AFS over the indicated rounds (excluding round 0) within each seed, then report the mean and sample standard deviation across five seeds.

E.2.4 Models in MiniGrid

World Model. We employ a physics-aligned architecture that preserves spatial structure via coordinate-aware convolutions. The model incorporates a **supervised vector-quantized bottleneck** [113], which explicitly maps latent codes to discrete actions and conditions a residual dynamics engine through Feature-wise Linear Modulation (FiLM) [92], promoting object persistence and physical consistency.

Inverse Dynamics Models (IDM). To verify the robustness of sparse IDM, we compare two IDMs on the OOD test set:

- **Vanilla IDM:** Takes the entire observation frame and the agent's proprioceptive state as input.
- **Sparse IDM:** Built upon the vanilla IDM by applying a learnable feature mask to the input, which selectively filters out irrelevant information and yields a sparse representation. The mask is learned automatically during training, encouraging the model to focus on the most informative local features.

Architecturally, both variants share a convolutional encoder that extracts spatial features from observations, followed by a projection layer that integrates object-centric attributes. Action prediction is performed by jointly reasoning over embeddings of consecutive frames, augmented with explicit geometric cues including relative position and direction changes.

E.2.5 Exploration Methods in MiniGrid

The **Oracle** strategy used in Sec. 4 selects samples with the highest prediction loss, corresponding to the hardest transitions under the current world model. To better understand the role of sample difficulty during exploration, we further introduce two oracle variants:

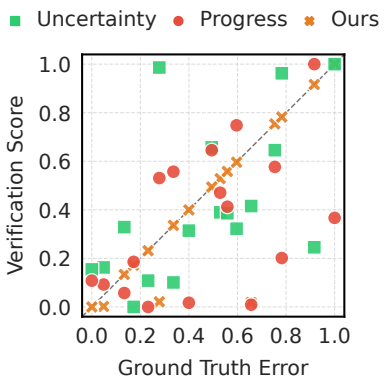

Figure 11: Verification score vs. ground-truth error. Our method exhibits a strong monotonic alignment with true error, whereas baselines show scattered distributions and frequent misranking.

- **Oracle-Easy:** selects samples with the lowest prediction loss, corresponding to the easiest transitions.
- **Oracle-Uniform:** partitions samples into disjoint prediction-loss intervals and selects high-loss samples within each interval, ensuring balanced coverage across different difficulty levels.

To analyze the behavior of different exploration strategies, we visualize the distribution of prediction errors—measured by the world model's prediction loss—over the samples selected by each method (Figure 12). This analysis reveals how different selection criteria bias the collected data toward specific difficulty regimes and provides insight into their impact on world model learning.

Qualitative Results. Figure 14 presents a qualitative comparison of world model predictions across interactive actions. While most methods perform similarly on simple motion-dominated transitions, clear differences arise for interaction-centric actions such as *Toggle* and *Swap*. In these cases, models trained with actively selected data more accurately capture interaction-induced state changes, whereas *Random* exhibit a strong bias toward predicting frequent but uninformative movement actions (e.g., *Turn*), highlighting the benefit of informative data selection under distribution shift.

Verification Score vs. Ground Truth Error. As shown in Figure 11, our method exhibits a strong monotonic relationship between verification score and true error, with samples distributed closely along the diagonal. This indicates that the proposed verification mechanism provides a faithful estimate of sample difficulty, assigning higher scores to transitions that are indeed harder for the world model.

In contrast, baseline methods such as *Uncertainty* and *Progress* display significantly weaker alignment, with scattered distributions and frequent misranking of samples. In particular, they tend to assign high scores to samples with relatively low true error or fail to consistently identify high-error transitions.

These observations are consistent with the quantitative results reported in Figure 4 (Mid), where our method achieves the highest Spearman and Kendall correlations. Together, the results further validate that our verification scores accurately capture the underlying difficulty of transitions, leading to more effective data selection.

E.3 Robotic Domains Setting

The experiments are conducted on simulated robotic manipulation tasks from RoboMimic [142] (`Lift`, `Can`, `Square`) and ManiSkill [86] (`PullCube`, `PokeCube`, `LiftPeg`). Figure 13 provides a visualization of these tasks. For each task, the agent observes RGB images from both a wrist-mounted camera and a front-facing camera. In addition, proprioceptive observations are provided, including the end-effector position and orientation, as well as the gripper position. The action space is a 7-dimensional continuous vector in $[-1, 1]$, including changes in the end-effector position and orientation and the gripper's opening and closing state.

E.3.1 Additional Details on Setups.

Dataset Collection. For both benchmarks, we train 10 diffusion policy models to collect a diverse set of trajectories, resulting in a total of 1500 samples per task. We follow the default horizon

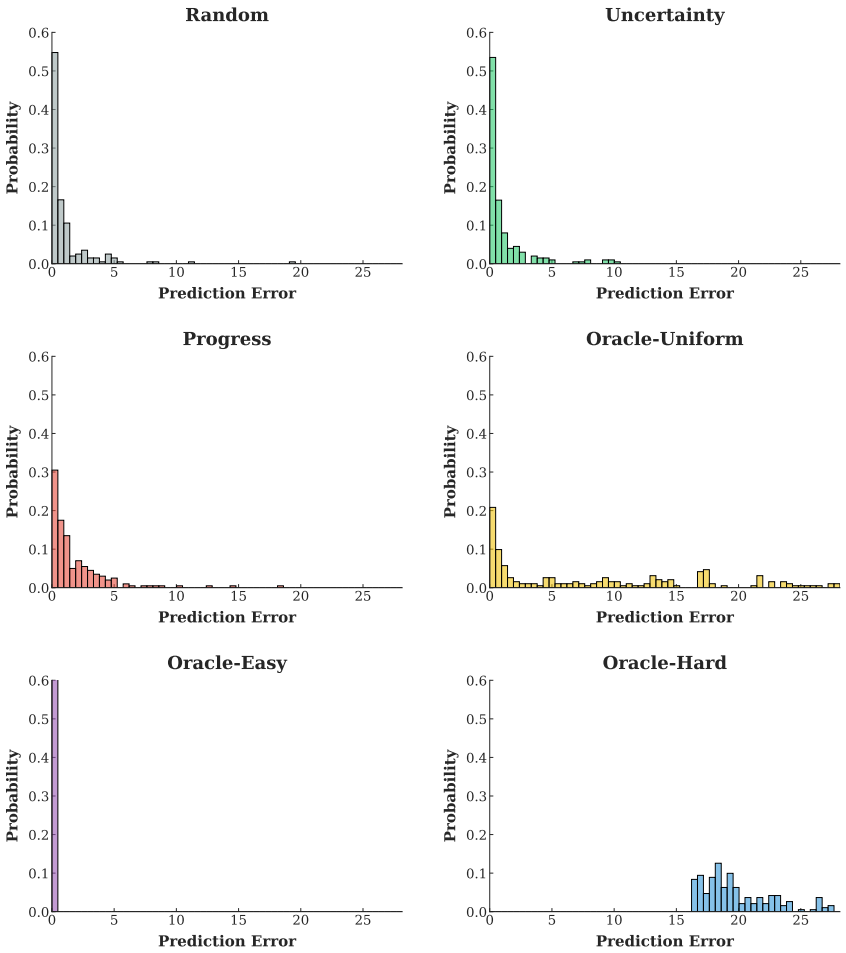

Figure 12: The distribution of world model's prediction error on the data selected by different exploration methods in MiniGrid.

settings of each environment: 100 steps for `Lift`, 200 for `Can` and `Square`, 50 for `PullCube`, 100 for `PokeCube`, and 150 for `LiftPeg`. We use nine of these checkpoints to construct the exploration dataset, and reserve the remaining checkpoint as a validation set for evaluating world model learning quality.

Reward Evaluation. We follow the reward formulation of SAILOR [54]. Specifically, we train a reward model (using the same architecture and hyperparameters as in SAILOR) to score the latent states of our world model based on how expert-like they are. The reward model is trained as a discriminator between latent embeddings from expert rollouts and learner rollouts, using a moment-matching objective with a gradient penalty.

E.3.2 Details on World Models.

We adopt an action-conditioned world model based on Dreamer-V3 [41]. Visual observations are encoded using a convolutional encoder with stride-2 convolutions, and proprioceptive states are embedded with a 5-layer MLP. Based on empirical performance, image and state inputs are processed by separate encoders. For decoding, image observations are reconstructed using a transposed-convolutional decoder with stride-2 upsampling, and proprioceptive states are reconstructed via a 5-layer MLP.

The latent state z_t comprises a deterministic recurrent component h_t and a stochastic component s_t. The deterministic state is modeled by a GRU with a 512-dimensional hidden state and is updated using the previous latent state z_{t-1} and action a_{t-1}. The resulting recurrent state is then used by the dynamics model to parameterize the distribution of the stochastic latent variable s_t. The number

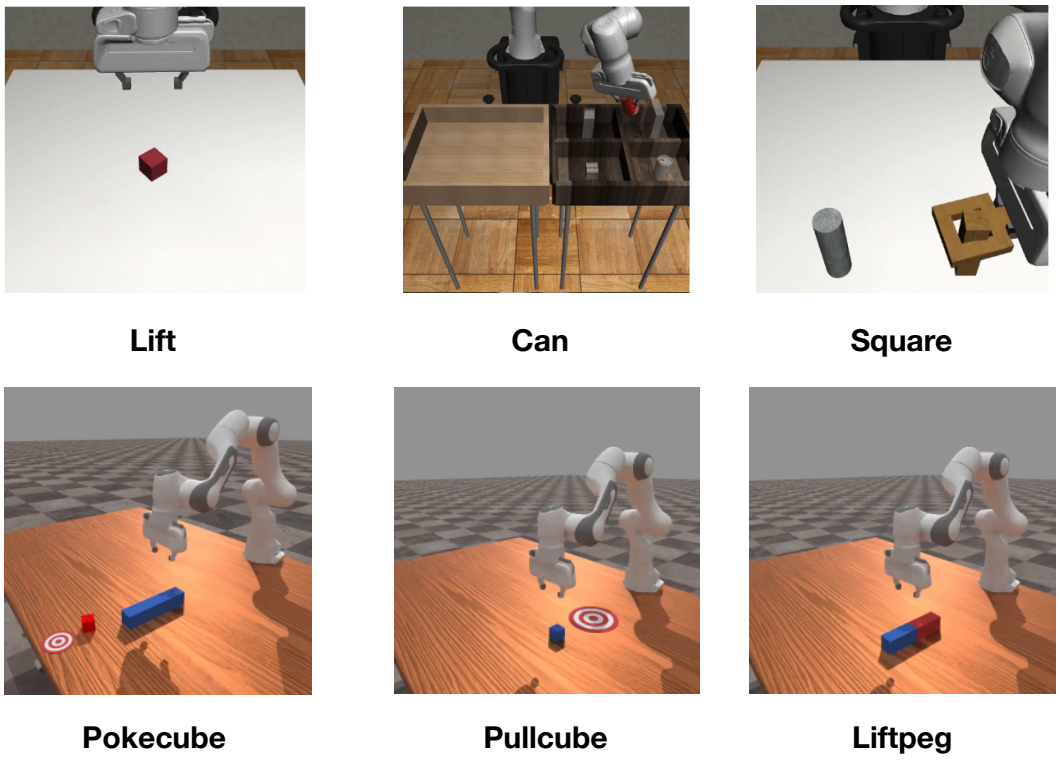

Figure 13: Visual overview of the RoboMimic (top row) and ManiSkill (bottom row) tasks.

Table 7: World model training hyperparameters.

World Model	
Replay capacity	1×10^5
Batch size	16
Batch length	32
Optimizer	Adam
Reconstruction loss scale	1.0
Learning rate	1×10^{-4}

of dimensions of stochastic representation is 1024. The number of the rollout horizon is 32. Other training hyperparameters are given in Table 7.

E.4 Details on Inverse Dynamic Models.

We adopt the inverse dynamics modeling framework from CLAM [69] as our base IDM architecture. Given consecutive observations (o_t, o_{t+1}), a latent inverse dynamics model infers a continuous latent action $z_t = f_\phi(o_t, o_{t+1})$ that captures the underlying transition. This latent action is then used to condition a latent forward dynamics model, $g_\psi(o_{t+1} \mid o_t, z_t)$, which predicts the next observation $\hat{o}_{t+1}$. An action decoder $p_\omega(a_t \mid z_t)$ maps the latent action back to the environment action space. The IDM, FDM, and action decoder are jointly trained using a combination of reconstruction and action prediction losses over both labeled and unlabeled trajectories. Compared to CLAM, we encourage sparsity in the latent action space by applying an ℓ_1 regularization term to the inferred latent actions z_t. encouraging the model to discover structured and task-relevant factors.

For visual observations, following CLAM, we adopt a space-time Transformer [8] for encoders. Each $256 \times 256 \times 3$ RGB image is first partitioned into non-overlapping 16×16 patches, yielding 16 visual tokens per frame, which are projected into a shared hidden space via a linear embedding layer. The encoder is composed of the stacked ST attention layers. In the decoder, each ST block performs cross-attention between visual tokens and the latent action representations produced by the encoder. Other hyperparameters are given in Table 8.

Table 8: Hyperparameters for inverse dynamics and action decoding.

Hyperparameter	Value
Num updates	500,000
Train action decoder every	2
Action decoder batch size	128
Action decoder loss weight	1
Action decoder hidden dim	[1024, 1024, 1024]
Action decoder embedding dim	512
Reconstruction loss weight	1
Sparsity loss weight	0.1
Latent action dim	16
Context len	2
Embedding dim	128

E.4.1 Visualization

Figures 15–16 visualize open-loop rollouts on RoboMimic Lift and RoboMimic Square. Overall, the base model exhibits poor visual predictions, with noticeable degradation in both rendering quality and dynamical consistency. Incorporating uncertainty- and progress-aware exploration substantially improves visual fidelity with more samples, producing sharper and more coherent renderings over time. In contrast, the vanilla IDM improves the accuracy of the underlying dynamics, indicating that inverse dynamics supervision helps recover action-relevant transitions. However, over long horizons, particularly in the final one to two frames, noticeable discrepancies remain, such as misaligned gripper orientations and inaccurate object occlusions. Our sparse IDMs further mitigate these long-horizon errors, yielding more stable dynamics and better-preserved fine-grained details in the predicted rollouts.

F Additional Theoretical Derivations

F.1 Detailed Derivation for Sec. C

This appendix provides a compact, self-contained proof of the self-improvement guarantee stated in Sec. C. The key idea is that, under a *generation–verification gap*, the full transition $(\mathbf{z}^t, \mathbf{a}^t) \mapsto \mathbf{z}^{t+1}$ may be out-of-support, while a small *verification subset* of latent variables remains on-support and is sufficient to recover the action. WAV uses this subset to *verify* (infer) missing action labels on OOS transitions, thereby expanding its action-labeled support.

F.1.1 Time-Lagged Latent Causal Model (TLCM)

We consider a *time-lagged latent causal model* (TLCM) with k latent blocks $\mathbf{z}^t := (\mathbf{z}_1^t, \dots, \mathbf{z}_k^t)$, actions $\mathbf{a}^t$, and observations $\mathbf{x}^t$. For clarity, we present the (deterministic) Markovian case used in our analysis:

$$\mathbf{z}^{t+1} = g(\mathbf{z}^t, \mathbf{a}^t), \tag{15}$$

$$\mathbf{x}^t = \varphi(\mathbf{z}^t), \tag{16}$$

where φ is a diffeomorphism onto its image (so φ^{-1} is well-defined on observed states). The induced causal graph over $(\mathbf{z}^t, \mathbf{a}^t, \mathbf{z}^{t+1})$ factorizes as

$$p(\mathbf{z}^{t+1} \mid \mathbf{z}^t, \mathbf{a}^t) \;=\; \prod_{i=1}^{k} p(\mathbf{z}_i^{t+1} \mid \mathrm{Pa}(\mathbf{z}_i^{t+1})). \tag{17}$$

F.1.2 Support, compositional OOS, and the verification subset

Let $\mathcal{D}_{\mathrm{act}} \subset \{(\mathbf{x}^t, \mathbf{a}^t, \mathbf{x}^{t+1})\}$ be the action-labeled seed dataset inducing a seed distribution $P_{\text{seed}}(\mathbf{z}^t, \mathbf{a}^t, \mathbf{z}^{t+1})$. Write

$$S_{\text{seed}} \;:=\; \mathrm{supp}\left(P_{\text{seed}}(\mathbf{z}^t, \mathbf{a}^t)\right)$$

for the on-support set of state–action pairs.

Definition F.1 (On-support vs. out-of-support (OOS))**.** A state–action pair $(\mathbf{z}^t, \mathbf{a}^t)$ is *on-support* if it lies in S_{seed} and is *out-of-support* (OOS) otherwise. For any index set $\mathcal{U} \subseteq \{1, \dots, k\}$ we also define the marginal support

$$S^{\mathcal{U}}_{\text{seed}} \;:=\; \operatorname{supp}\big(P_{\text{seed}}(\mathbf{z}^t_{\mathcal{U}}, \mathbf{a}^t)\big),$$

and say $(\mathbf{z}^t_{\mathcal{U}}, \mathbf{a}^t)$ is on-support if it lies in $S^{\mathcal{U}}_{\text{seed}}$.

Definition F.2 (Compositional OOS transition)**.** A transition $(\mathbf{z}^t, \mathbf{a}^t, \mathbf{z}^{t+1})$ is *compositional OOS* (w.r.t. P_{seed}) if $(\mathbf{z}^t, \mathbf{a}^t) \notin S_{\text{seed}}$ but there exists a subset of variables $\mathcal{U}$ such that $(\mathbf{z}^t_{\mathcal{U}}, \mathbf{a}^t) \in S^{\mathcal{U}}_{\text{seed}}$. Intuitively, novelty comes from an unseen *combination* of factors, while at least one subset transition remains within the training support.

We now formalize which subset can serve as a verifier.

Definition F.3 (Source (insulated) set)**.** Let $\mathcal{S}_{\text{src}} \subseteq \{1, \dots, k\}$ be a set of latent blocks that is *causally insulated* from its complement in the TLCM graph:

$$\forall i \in \mathcal{S}_{\text{src}}, \qquad \operatorname{Pa}(\mathbf{z}^{t+1}_i) \;\subseteq\; \{\mathbf{z}^t_j : j \in \mathcal{S}_{\text{src}}\} \cup \{\mathbf{a}^t\}.$$

Equivalently, there are no directed edges from $\mathbf{z}^t_{\setminus \mathcal{S}_{\text{src}}}$ into $\mathbf{z}^{t+1}_{\mathcal{S}_{\text{src}}}$.

Definition F.4 (Verification subset)**.** Let $\mathcal{S}_{\text{act}} := \{i : \mathbf{a}^t \in \operatorname{Pa}(\mathbf{z}^{t+1}_i)\}$ denote the *action-influenced* latent blocks. We define the *verification subset* as

$$\mathcal{S} \;:=\; \mathcal{S}_{\text{src}} \cap \mathcal{S}_{\text{act}}.$$

The subset we require to remain on-support for verification is the *intersection* of (i) source/insulated variables and (ii) action-influenced variables.

Assumption F.5 (Generation–verification gap (information asymmetry))**.** *There exists a verification subset $\mathcal{S}$ such that for every (potentially OOS) transition $(\mathbf{z}^t, \mathbf{a}^t, \mathbf{z}^{t+1})$ we wish to label, the restricted state–action pair remains on-support:*

$$(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}^t) \in S^{\mathcal{S}}_{seed},$$

even though $(\mathbf{z}^t, \mathbf{a}^t)$ may lie outside S_{seed}.

F.1.3 Two identifiability ingredients from prior work

Our proof uses (i) identifiability of the latent blocks (up to permutation / element-wise transforms) and (ii) identifiability of the action from on-support subset transitions.

Condition F.6 (Identifiable latent blocks via mechanism sparsity)**.** *This condition is adapted from Proposition 7 (together with Assumption 5) of Lachapelle et al. [63]. Consider a TLCM whose observation model is a diffeomorphism (so φ is invertible on its image) and whose transition model is Markov with respect to a sparse dependency graph between $(\mathbf{z}^t, \mathbf{a}^t)$ and $\mathbf{z}^{t+1}$ (as in Equation* (17)*). Assume we learn a second TLCM $(\hat{\varphi}, \hat{g}, \hat{G})$ that is $(\mathbf{z}, \mathbf{a})$-consistent with the ground-truth model in the sense of Lachapelle et al. [63] and that the ground-truth graph satisfies their* graphical criterion *(Assumption 5). Then the learned latent variables are identifiable up to a permutation and element-wise invertible transformations ("complete disentanglement"), so we can treat the learned blocks as the true blocks up to a fixed relabeling.*

Condition F.7 (Identifiable action from subset transitions)**.** *This condition is adapted from Theorem 1 of Lachapelle [62] by substituting $\mathbf{x} \equiv \mathbf{z}^t_{\mathcal{S}}$ and $\mathbf{x}' \equiv \mathbf{z}^{t+1}_{\mathcal{S}}$. Let $\mathcal{S}$ be a fixed verification subset and define the subset dynamics $g_{\mathcal{S}}(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}^t) := [g(\mathbf{z}^t, \mathbf{a}^t)]_{\mathcal{S}}$ (well-defined whenever $\mathcal{S} \subseteq \mathcal{S}_{\text{src}}$). Assume the following hold on $S^{\mathcal{S}}_{seed}$:*

1. ***Continuity:*** *for each action value $\mathbf{a}$, the map $\mathbf{z}^t_{\mathcal{S}} \mapsto g_{\mathcal{S}}(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a})$ is continuous;*
2. ***Injectivity:*** *for every $\mathbf{z}^t_{\mathcal{S}}$, $g_{\mathcal{S}}(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}_1) = g_{\mathcal{S}}(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}_2)$ implies $\mathbf{a}_1 = \mathbf{a}_2$;*
3. ***Connected conditional support:*** *for each $\mathbf{a}$ in the action support,* $\operatorname{supp}\big(P_{seed}(\mathbf{z}^t_{\mathcal{S}} \mid \mathbf{a})\big)$ *is connected;*
4. ***Support overlap:*** *for any $\mathbf{a}_1, \mathbf{a}_2$ in the action support,* $\operatorname{supp}\big(P_{seed}(\mathbf{z}^t_{\mathcal{S}} \mid \mathbf{a}_1)\big) \cap \operatorname{supp}\big(P_{seed}(\mathbf{z}^t_{\mathcal{S}} \mid \mathbf{a}_2)\big) \neq \varnothing$.

Then the action is identifiable from the subset transition $(\mathbf{z}^t_{\mathcal{S}}, \mathbf{z}^{t+1}_{\mathcal{S}})$ up to a fixed relabeling: there exists an injective map v (independent of $\mathbf{z}^t_{\mathcal{S}}$) such that any solution of the latent-action reconstruction problem in Lachapelle [62] recovers $v(\mathbf{a})$ deterministically from $(\mathbf{z}^t_{\mathcal{S}}, \mathbf{z}^{t+1}_{\mathcal{S}})$. In particular, when the learned action alphabet matches the true discrete action set, this corresponds to a permutation of action labels.

F.1.4 Verified self-improvement

We now state and prove the main appendix result.

Theorem F.8 (Identifiability of Self-Improvement)**.** *Let $\mathcal{D}_{\text{act}} \subset \{(\mathbf{x}^t, \mathbf{a}^t, \mathbf{x}^{t+1})\}$ be action-labeled transitions sampled from P_{seed}, and let $(\mathbf{x}^{*,t}, \mathbf{x}^{*,t+1})$ be an additional unlabeled transition sampled from some test distribution p_{test}. Let $(\mathbf{z}^{*,t}, \mathbf{a}^*, \mathbf{z}^{*,t+1})$ denote the corresponding latent transition in the TLCM, i.e. $\mathbf{z}^{*,t+1} = g(\mathbf{z}^{*,t}, \mathbf{a}^*)$ and $\mathbf{x}^{*,t+1} = \varphi(\mathbf{z}^{*,t+1})$. Assume:*

1. ***Latent blocks are identified** up to a fixed permutation / element-wise transform (Condition F.6);*
2. ***Generation–verification gap** holds for the verification subset $\mathcal{S}$ (Assumption F.5);*
3. ***Action is identifiable from subset transitions** on $S^{\mathcal{S}}_{seed}$ (Condition F.7).*

Let $h_\psi : (\mathbf{z}^t_{\mathcal{S}}, \mathbf{z}^{t+1}_{\mathcal{S}}) \mapsto \mathbf{a}^t$ be an inverse dynamics model trained on the on-support subset transitions in $\mathcal{D}_{\text{act}}$. Then the missing action label for the unlabeled transition is uniquely determined by $(\mathbf{z}^{*,t}_{\mathcal{S}}, \mathbf{z}^{*,t+1}_{\mathcal{S}})$, and

$$\hat{\mathbf{a}}^* \;:=\; h_\psi(\mathbf{z}^{*,t}_{\mathcal{S}}, \mathbf{z}^{*,t+1}_{\mathcal{S}})$$

recovers the true action (up to the fixed label relabeling in Condition F.7; with labeled data this relabeling is resolved so that $\hat{\mathbf{a}}^* = \mathbf{a}^*$). Consequently, the tuple $(\mathbf{x}^{*,t}, \hat{\mathbf{a}}^*, \mathbf{x}^{*,t+1})$ is correctly labeled while it may satisfy $(\mathbf{z}^{*,t}, \mathbf{a}^*) \notin S_{seed}$, i.e. it can expand the action-labeled support.

Proof. By Condition F.6 (a restatement of Lachapelle et al. [63, Prop. 7] in our notation), the learned representation can be aligned with the ground-truth latent blocks up to a fixed permutation and element-wise invertible transforms. This alignment preserves the block structure and (up to a fixed relabeling) the parent/child relations in the TLCM graph, so the verification subset $\mathcal{S} = \mathcal{S}_{\text{src}} \cap \mathcal{S}_{\text{act}}$ is well-defined and accessible from observations via the learned encoder.

By Assumption F.5, the subset state–action pair $(\mathbf{z}^{*,t}_{\mathcal{S}}, \mathbf{a}^*)$ lies in the on-support set $S^{\mathcal{S}}_{\text{seed}}$, even if the full pair $(\mathbf{z}^{*,t}, \mathbf{a}^*)$ is OOS. Therefore, the on-support subset transitions contained in $\mathcal{D}_{\text{act}}$ are sufficient to train an inverse model h_ψ for $g_{\mathcal{S}}$.

Finally, by Condition F.7 (adapted from Lachapelle [62, Thm. 1] with $\mathbf{x} \equiv \mathbf{z}^t_{\mathcal{S}}$ and $\mathbf{x}' \equiv \mathbf{z}^{t+1}_{\mathcal{S}}$), the action is identifiable from the subset transition $(\mathbf{z}^t_{\mathcal{S}}, \mathbf{z}^{t+1}_{\mathcal{S}})$ up to a fixed relabeling. Hence applying h_ψ to the on-support subset transition $(\mathbf{z}^{*,t}_{\mathcal{S}}, \mathbf{z}^{*,t+1}_{\mathcal{S}})$ recovers the correct action label (after resolving the fixed relabeling using the labeled actions in $\mathcal{D}_{\text{act}}$). This yields the correctly labeled tuple $(\mathbf{x}^{*,t}, \hat{\mathbf{a}}^*, \mathbf{x}^{*,t+1})$, which can lie outside the original support S_{seed} and thus expands the action-labeled coverage. □

F.1.5 Implications of the generation–verification gap

We now interpret Theorem F.8 and Assumption F.5 through a practical lens, identifying when WAV is most beneficial, when it degrades, and when it fails.

How large is the gap? WAV separates *verification* (recovering the missing action) from *generation* (predicting the full next state). Verification only uses the subset transition on $\mathbf{z}_{\mathcal{S}}$, while generation must model the full $\mathbf{z}$. As a rule of thumb, WAV becomes most attractive when $\dim(\mathbf{z}_{\mathcal{S}}) \ll \dim(\mathbf{z})$: the inverse model stays simple and stable even as the world model faces increasingly many OOS compositions.

Condition 1: fixed verifier, growing scene (maximum benefit). If $\mathbf{z}_{\mathcal{S}}$ is agent-centric and fixed-dimensional (e.g., proprioception) while the rest of the scene grows in complexity (more objects, tools, contacts), then action recovery continues to rely on the same low-dimensional signal while the world model must extrapolate over a much larger state space. In this regime, the benefit of WAV grows with scene complexity. *Example (warehouse robot).* A 7-DoF manipulator arm has $\dim(\mathbf{z}_{\mathcal{S}}) = 7$ (joint angles/velocities). In a warehouse with many objects, the world model must predict the state of each object (and their interactions), so $\dim(\mathbf{z})$ grows with scene complexity. As scene complexity grows, predicting the full next state requires accounting for many interacting factors beyond the agent's direct control, which increases the difficulty of accurate forward prediction. In many control settings, however, the most useful signal is the action-imprinted change. This motivates focusing the inverse

model on an agent-centric subset $\mathbf{z}_{\mathcal{S}}$, where action recovery can remain stable as the rest of the scene grows, yielding a practical forward–inverse gap that we exploit for self-improvement.

Condition 2: compositional OOS with preserved source-set (strong benefit). WAV succeeds when OOS novelty is concentrated in $\mathbf{z}_{\setminus\mathcal{S}}$ while the verifying subset behaves as it did in training. Concretely, even when the full pair $(\mathbf{z}^t, \mathbf{a}^t) \notin S_{\text{seed}}$, the subset transition on $\mathbf{z}_{\mathcal{S}}$ remains on-support, so an inverse model trained on $\mathcal{D}_{\text{act}}$ can still recover $\mathbf{a}^t$ reliably. *Example (tool–object contact).* Training contains "move knife in free space" and "touch apple with hand," but not "slice apple with knife." At test time, the full transition is OOS (contact dynamics are novel), yet the arm motion $\mathbf{z}_{\mathcal{S}}$ follows familiar trajectories. The inverse model can still recover the action, enabling the world model to learn the novel contact outcome.

Stochastic extension. For the remaining discussion we consider a stochastic generalization of the TLCM where $\mathbf{z}^{t+1} = g(\mathbf{z}^t, \mathbf{a}^t, \boldsymbol{\epsilon}^t)$ with exogenous noise $\boldsymbol{\epsilon}^t$; Theorem F.8 holds in the deterministic special case $\boldsymbol{\epsilon}^t = \mathbf{0}$.

Condition 3: weak injectivity / action aliasing (degradation). Condition F.7 (2) requires the mapping $\mathbf{a} \mapsto g_{\mathcal{S}}(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}, \boldsymbol{\epsilon})$ to be injective. When different actions produce indistinguishable (or nearly indistinguishable) subset transitions, recovered actions become ambiguous:

$$\exists\, \mathbf{a} \neq \mathbf{a}' \text{ s.t. } g_{\mathcal{S}}(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}, \boldsymbol{\epsilon}) \approx g_{\mathcal{S}}(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}', \boldsymbol{\epsilon}). \tag{18}$$

Example (underactuation / latency). In a soft gripper, different motor commands may produce nearly identical proprioceptive changes due to compliance. Similarly, unmodeled communication delays can smear the action's effect across timesteps, violating injectivity. In such cases, pseudo-labels drift and self-improvement degrades.

Condition 4: back-action from OOS into the verifier (failure). The verifying subset must remain insulated from OOS components (the *source set* property of Definition 3). WAV fails when OOS variables feed back into the verifier so that $\mathbf{z}_{\mathcal{S}}$ itself goes out-of-support. One way to view the failure mode is that the verifier dynamics no longer depend only on $(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}^t)$, but also on $\mathbf{z}^t_{\setminus\mathcal{S}}$:

$$\mathbf{z}^{t+1}_{\mathcal{S}} = g_{\mathcal{S}}(\mathbf{z}^t_{\mathcal{S}}, \mathbf{a}^t, \mathbf{z}^t_{\setminus\mathcal{S}}, \boldsymbol{\epsilon}^t). \tag{19}$$

Example (compliant contact). When a robot arm makes stiff contact with a deformable object, contact forces feed back into joint-level torques and velocities. The "verifying" proprioceptive dynamics now depend on the OOS object state, breaking the source-set assumption and causing action recovery to fail.

Summary. WAV is most effective when (i) the verifying subset is small and fixed-dimensional relative to the full state, (ii) OOS novelty is confined to non-verifying blocks while $\mathbf{z}_{\mathcal{S}}$ stays on-support, and (iii) the action imprint on $\mathbf{z}_{\mathcal{S}}$ is strong (high injectivity). It degrades under action aliasing and fails when OOS dynamics causally influence the verifier.

F.2 Detailed Derivation for Sec. 3

This appendix provides the exact derivation behind Sec. 3. The purpose of the analysis is to isolate the statistical asymmetry exploited by WAV: predicting the full next state can be substantially harder than verifying the action from a low-dimensional action-relevant slice.

Setup. Let $s \in \mathbb{R}^{d_s}$ be the state, $a \in \mathbb{R}^{d_a}$ the action, and suppose the one-step dynamics are linear with additive Gaussian noise:

$$s' = As + Ba + \xi, \qquad \xi \sim \mathcal{N}(0, \sigma_s^2 \mathbf{I}_{d_s}). \tag{20}$$

Assume also that there exists an action-relevant slice $z = Ms \in \mathbb{R}^{d_z}$, with $d_z \mathbf{l} d_s$, from which the action can be linearly recovered up to irreducible ambiguity:

$$a = H \begin{bmatrix} z \\ z' \end{bmatrix} + \eta, \qquad z' = Ms', \qquad \eta \sim \mathcal{N}(0, \sigma_a^2 \mathbf{I}_{d_a}). \tag{21}$$

We compare a dense forward regressor $\hat{f}$ trained on

$$x_F := \begin{bmatrix} s \\ a \end{bmatrix} \in \mathbb{R}^{d_s+d_a}$$

and a sparse inverse regressor $\hat{h}$ trained on

$$x_I := \begin{bmatrix} z \\ z' \end{bmatrix} \in \mathbb{R}^{2d_z}.$$

For analytic tractability, we assume both feature vectors have been whitened:

$$x_F \sim \mathcal{N}(0, \mathbf{I}_{d_s+d_a}), \qquad x_I \sim \mathcal{N}(0, \mathbf{I}_{2d_z}), \tag{22}$$

and both models are fit by ordinary least squares on n i.i.d. labeled transitions. (The whitening assumption simplifies the algebra; for general covariance Σ the excess risk scales with $\mathrm{tr}(\Sigma^{-1})$—see, *e.g.*, Hsu et al. [48]—and the qualitative three-factor decomposition is preserved.)

As in the main text, we compare both models in the state space:

$$\mathcal{E}_F := \frac{1}{d_s}\,\mathbb{E}\Big[\|\hat{f}(s,a) - f^\star(s,a)\|_2^2\Big], \tag{23}$$

$$\mathcal{E}_I := \frac{1}{d_s}\,\mathbb{E}\Big[\|f^\star(s,\hat{h}(z,z')) - f^\star(s,h(z,z'))\|_2^2\Big]. \tag{24}$$

Lemma F.9 (OLS excess risk under isotropic Gaussian covariates)**.** *Consider scalar regression*

$$y = \langle w^\star, x\rangle + \epsilon, \qquad x \sim \mathcal{N}(0, \mathbf{I}_D), \qquad \epsilon \sim \mathcal{N}(0, \nu^2),$$

with $n > D+1$. If $\hat{w}$ is the OLS estimator fit on n i.i.d. samples, then its expected excess risk is

$$\mathbb{E}\big[(\langle \hat{w} - w^\star, x\rangle)^2\big] = \nu^2\,\frac{D}{n-D-1}. \tag{25}$$

This is a classical exact expression for well-specified linear regression with isotropic Gaussian design; see, e.g., Hsu et al. [48], Mourtada [85].

Proposition F.10 (Exact forward–inverse gap in the linear–Gaussian model)**.** *Under the setup above, let $\lambda := \|B\|_{\mathrm{op}}$. If $n > d_s + d_a + 1$ and $n > 2d_z + 1$, then*

$$\mathbb{E}[\mathcal{E}_F] = \sigma_s^2\,\frac{d_s+d_a}{n-(d_s+d_a)-1}, \tag{26}$$

$$\mathbb{E}[\mathcal{E}_I] \le \lambda^2\,\frac{d_a}{d_s}\,\sigma_a^2\,\frac{2d_z}{n-2d_z-1}. \tag{27}$$

Consequently, the error ratio satisfies

$$\Gamma(n) := \frac{\mathbb{E}[\mathcal{E}_F]}{\mathbb{E}[\mathcal{E}_I]} \;\ge\; \Big(\frac{d_s+d_a}{2d_z}\cdot\frac{d_s}{d_a}\Big)\cdot\Big(\frac{\sigma_s}{\lambda\,\sigma_a}\Big)^2\cdot\Big(\frac{n-2d_z-1}{n-(d_s+d_a)-1}\Big). \tag{28}$$

Proof. We apply Lemma F.9 separately to the forward and inverse regressions.

Forward model. Each coordinate of s' in (20) is a scalar linear regression on the feature vector $x_F \in \mathbb{R}^{d_s+d_a}$ with noise variance σ_s^2. Therefore,

$$\mathbb{E}\Big[(\hat{f}_j(s,a) - f_j^\star(s,a))^2\Big] = \sigma_s^2\,\frac{d_s+d_a}{n-(d_s+d_a)-1}$$

for each state coordinate $j \in \{1,\dots,d_s\}$. Averaging over the d_s coordinates yields

$$\mathbb{E}[\mathcal{E}_F] = \sigma_s^2\,\frac{d_s+d_a}{n-(d_s+d_a)-1},$$

which is exactly (26).

Inverse model in action space. Similarly, each coordinate of a in (21) is a scalar linear regression on $x_I \in \mathbb{R}^{2d_z}$ with noise variance σ_a^2. Hence

$$\mathbb{E}\Big[(\hat{h}_k(z,z') - h_k(z,z'))^2\Big] = \sigma_a^2 \, \frac{2d_z}{n - 2d_z - 1}$$

for each action coordinate $k \in \{1, \dots, d_a\}$. Summing across the d_a coordinates gives

$$\mathbb{E}\Big[\|\hat{h}(z,z') - h(z,z')\|_2^2\Big] = d_a \, \sigma_a^2 \, \frac{2d_z}{n - 2d_z - 1}. \tag{29}$$

Mapping inverse error back to state space. Because the true dynamics $f^\star$ are linear in the action,

$$f^\star(s, \hat{h}(z,z')) - f^\star(s, h(z,z')) = B\big(\hat{h}(z,z') - h(z,z')\big).$$

Therefore,

$$\mathcal{E}_I = \frac{1}{d_s} \, \mathbb{E}\left[\left\| B\big(\hat{h}(z,z') - h(z,z')\big)\right\|_2^2\right] \tag{30}$$

$$\leq \frac{\|B\|_{\mathrm{op}}^2}{d_s} \, \mathbb{E}\Big[\|\hat{h}(z,z') - h(z,z')\|_2^2\Big] \tag{31}$$

$$= \lambda^2 \, \frac{d_a}{d_s} \, \sigma_a^2 \, \frac{2d_z}{n - 2d_z - 1}, \tag{32}$$

which proves (27) after taking expectations and using (29).

Finally, dividing (26) by (27) yields (28). □

Reading the bound. The ratio in (28) cleanly factorizes into three interpretable pieces. The first term is a *dimensionality advantage*: the forward model must estimate a map from $d_s + d_a$ inputs, whereas the sparse inverse model only uses $2d_z$ inputs. The second term is a *stochasticity advantage*: forward prediction suffers from environment noise σ_s, while inverse verification only suffers from the ambiguity of recovering the action from the selected slice, measured by σ_a, after accounting for the state-space gain λ. The third term is a *sample-size advantage*: when n is only modestly larger than the dense forward dimension, the forward estimator is statistically much less stable.

Scope of the stylized model. This analysis is intentionally minimal. It does not claim that real robotic dynamics are linear or globally Gaussian. Instead, it isolates a statistical regime that matches the intuition behind WAV: if a low-dimensional subset preserves the action imprint while the full scene is high-dimensional, noisy, and sparsely labeled, then sparse inverse verification can be substantially more reliable than dense forward prediction.

G Broader Impact

WAV aims to improve the data efficiency and reliability of world-model learning for embodied agents. Potential benefits include reducing the amount of costly robot interaction needed for model improvement and making learned simulators more useful for policy evaluation. Potential risks include over-reliance on imperfect learned world models in safety-sensitive robotics settings, especially when verifier assumptions fail under domain shift. Any real-world deployment should therefore include task-specific validation, monitoring, and human oversight.

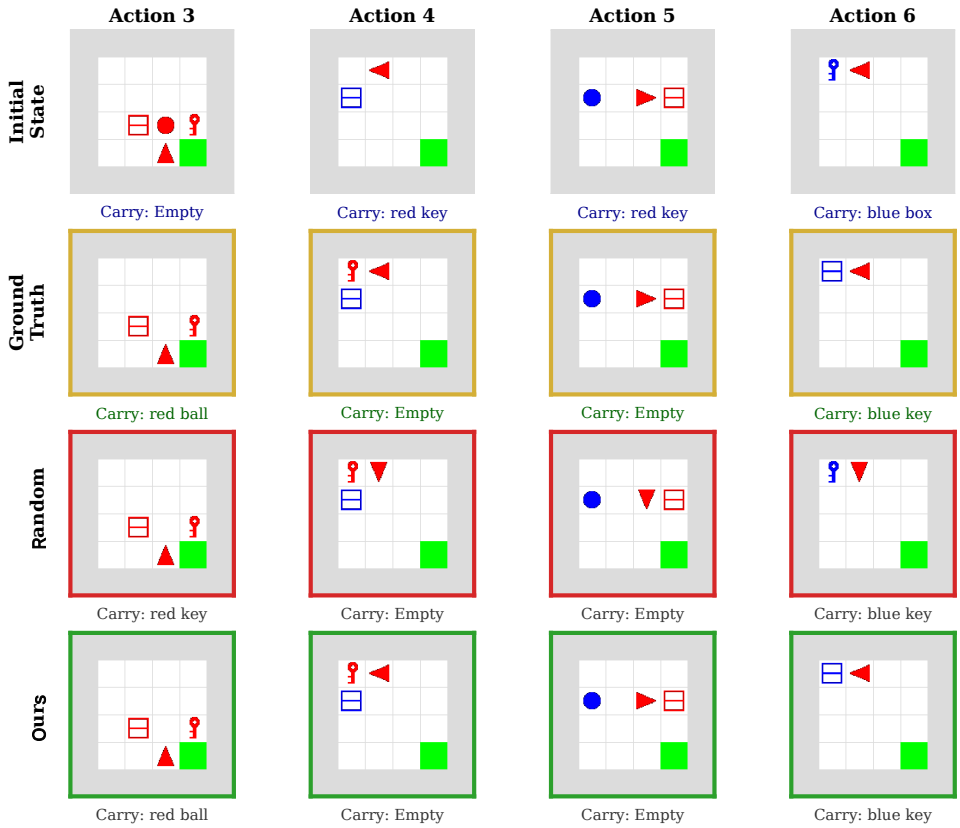

Figure 14: **Qualitative comparison of world-model rollouts under diverse interaction actions (Part I–II). Gold** borders denote ground-truth next observations; green and red borders indicate correct and incorrect predictions, respectively. Across both task sets, our method better preserves action-dependent state changes—notably for structured interactions such as *Toggle* and *Swap*—than exploration-based baselines.

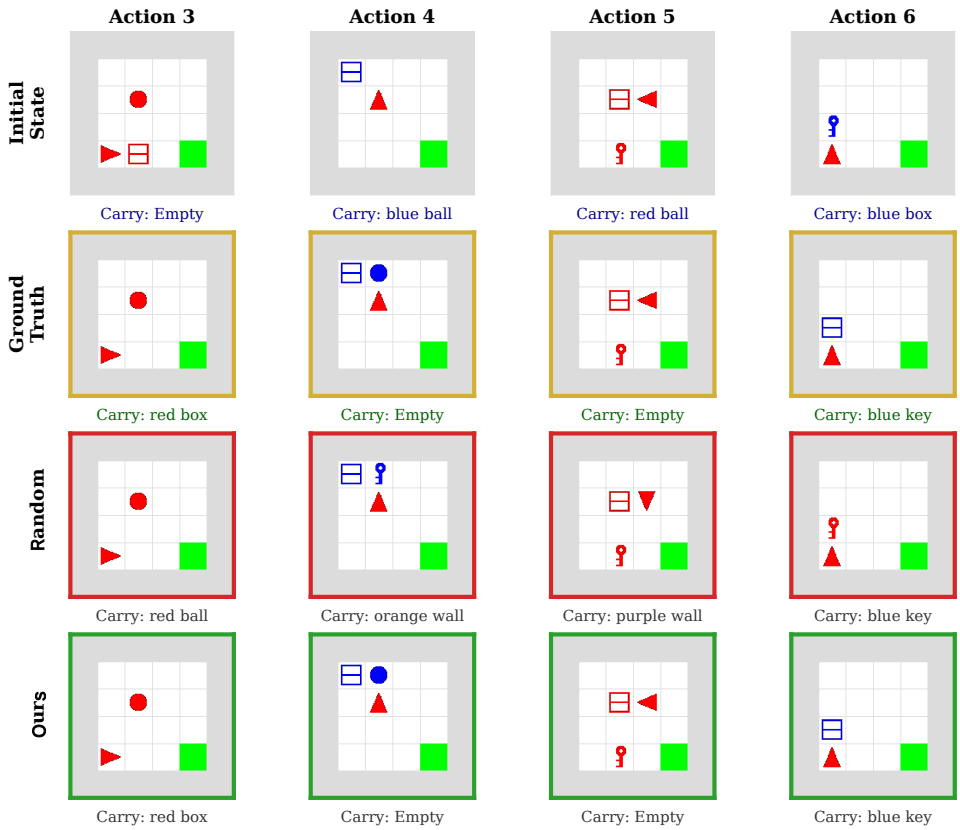

Figure 14: (Continued.) **Task Set B.** The **Random** baseline frequently collapses to predicting the most common primitive motions (e.g., *Turn*), failing to model interaction-induced state changes (e.g., *Toggle*). In contrast, models trained with data selected by our method capture these state transitions.

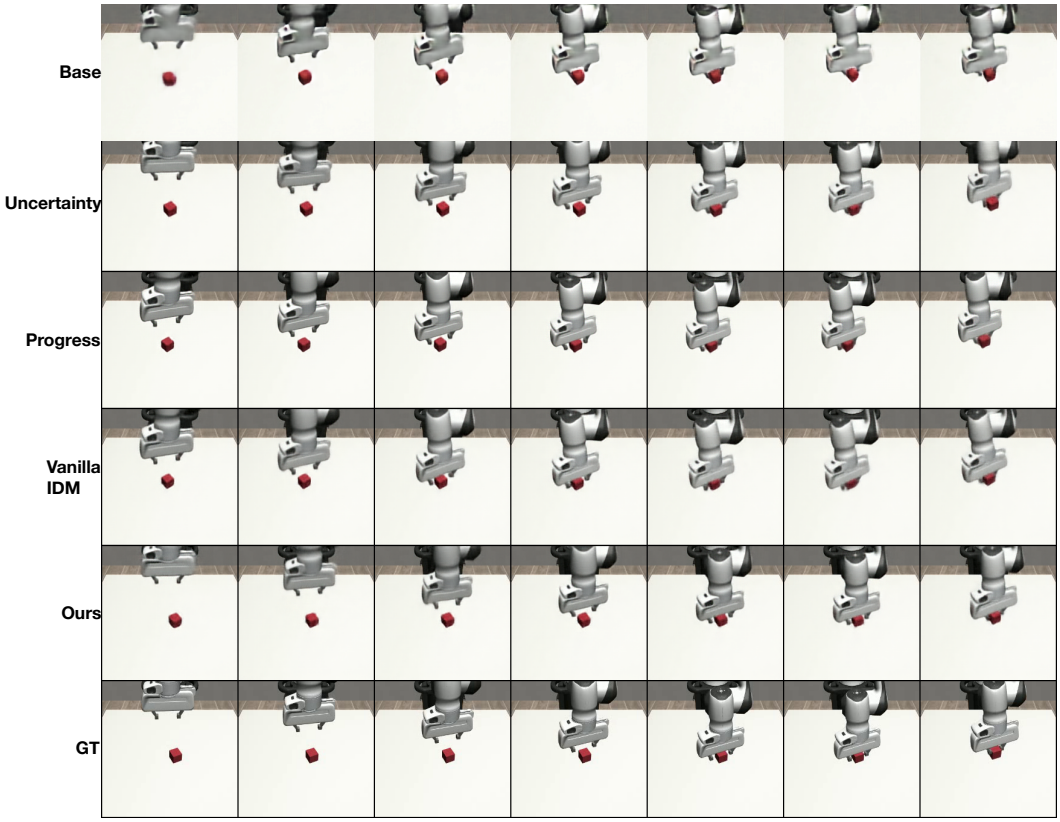

Figure 15: Qualitative comparison of world-model predictions across different methods on RoboMimic Lift.

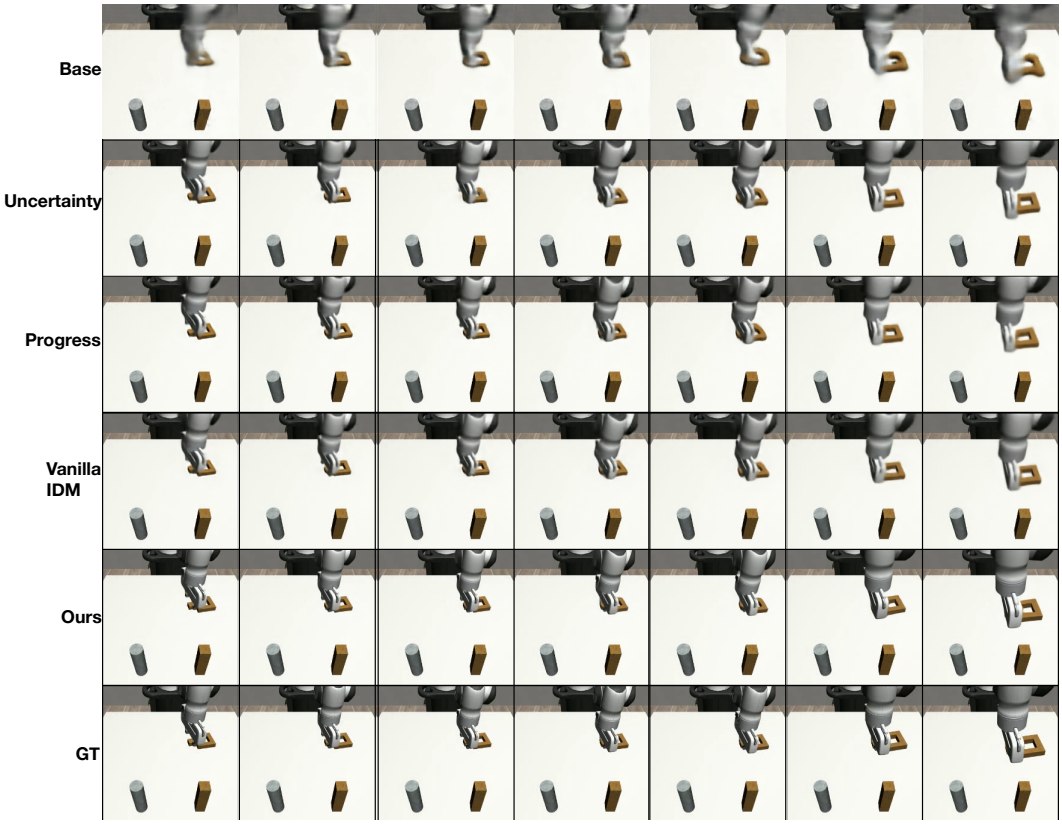

Figure 16: Qualitative comparison of world-model predictions across different methods on RoboMimic Square.

H NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: The abstract and introduction state the method, assumptions, scope, and main empirical claims, which are supported by the theory and experiments.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: We discuss the limitations of the method in Section D.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [Yes]
   Justification: The main text states the theoretical setup and results, and Appendix F.1 and Appendix F.2 provide the assumptions and proofs.
4. **Experimental result reproducibility**
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?
   Answer: [Yes]
   Justification: The main text and appendix describe the datasets, baselines, acquisition budgets, metrics, architectures, and hyperparameters needed to reproduce the main experimental results.
5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes]
   Justification: The supplementary material includes code and reproduction instructions; the experimental setup, data generation, and evaluation protocols are described in the main text and appendix.
6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?
   Answer: [Yes]
   Justification: The experimental setup is described in Section 4, with dataset composition, metrics, model details, and hyperparameters provided in Appendix A and Appendix E.2.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [Yes]
   Justification: MiniGrid results are averaged over five seeds, while the robotic world-model and OOD results report standard error over three seeds.
8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes]
   Justification: Compute resource details are included in the appendix.
9. **Code of ethics**
   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?
   Answer: [Yes]
   Justification: The work is conducted in simulated environments, does not involve human subjects, and does not release high-risk data or models.
10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
Answer: [Yes]
Justification: Appendix G discusses potential positive impacts, potential risks, and the need for validation and oversight before real-world deployment.

11. **Safeguards**
Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?
Answer: [N/A]
Justification: The paper does not release high-risk data or models such as pretrained language models, image generators, or scraped datasets.

12. **Licenses for existing assets**
Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?
Answer: [Yes]
Justification: The paper cites the existing benchmark environments and assets used in the experiments; detailed license handling is included in the appendix.

13. **New assets**
Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?
Answer: [N/A]
Justification: The paper does not release a new dataset, model, or benchmark asset.

14. **Crowdsourcing and research with human subjects**
Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
Answer: [N/A]
Justification: The paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**
Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?
Answer: [N/A]
Justification: The paper does not involve crowdsourcing or research with human subjects.

16. **Declaration of LLM usage**
Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.
Answer: [N/A]
Justification: The core method development in this research does not use LLMs as an important, original, or non-standard component.