DOES ACTION CONDITIONING TRULY ANCHOR FUTURE PREDICTIONS IN CAUSAL DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) are widely favored for their presumed capacity to internalize causal dynamics. However, it remains unclear whether the process of predicting future frames truly equips WAMs with such causal understanding. In this work, we reveal the indispensable role of action conditioning in learning dynamics, and demonstrate that only under a multimodal training distribution can WAMs truly acquire causal dynamics, enabling them to evaluate the current state within their latent space. In vanilla WAM architectures, the action conditioning module may be absent; in such cases, the model degenerates into a dynamics-agnostic predictor whose predicted future frames and executable actions lack causal grounding. Yet, even when equipped with action conditioning, training exclusively on unimodal expert data constrains the learning of causal dynamics, as the model is restricted to predicting a single deterministic future regardless of the input action. Through extensive experiments analyzing these scenarios and variants, we demonstrate that under multimodal, suboptimal data, action conditioning enables WAMs to learn more robust causal dynamics and exhibit an implicit state-value function within their latent space.

1 INTRODUCTION

World Action Models (WAMs) have emerged as a central paradigm in embodied intelligence, aiming to capture how actions reshape the environment over time. Built upon powerful generative backbones such as Diffusion Transformers (DiT) (Peebles & Xie, 2023), modern WAMs jointly predict future visual states and executable actions (Wu et al., 2024; Cheang et al., 2024; Du et al., 2023; Ye et al., 2025; Bruce et al., 2024), optionally bridged by *action conditioning* that injects candidate actions into the generative process (Agarwal et al., 2025; Cheang et al., 2024). A widely held assumption underpins this paradigm: if a model can faithfully predict future frames given an action, it must have internalized the causal dynamics linking that action to the future it induces.

However, this assumption conflates generating visually plausible futures with truly understanding causal dynamics. A model might produce highly coherent future frames simply by exploiting spatial and temporal continuity, entirely bypassing how a specific action alters the environment. This risk is amplified by the standard training regime: in expert demonstrations, the action is nearly a deterministic function of the state ($a \approx \pi_E(s)$), and the observed futures concentrate on a narrow band of successful trajectories. Under such a unimodal distribution, a far simpler shortcut suffices ($p(o_{t+1} \mid o_t, a_t) \approx p(o_{t+1} \mid o_t)$), leaving the model no incentive to consult the action (De Haan et al., 2019; Geirhos et al., 2020). We therefore hypothesize that action grounding is conferred not by the action condition itself but by the training distribution, which must render the action informative beyond the state alone.

We verify this diagnosis with two complementary interventions. First, in WAMs without action conditioning, shuffling the temporal order of the predicted future frames has minimal impact on action generation, indicating that the predictions carry weak causal structure. Second, in WAMs equipped with action conditioning, removing or randomizing the action condition leaves the predictions largely intact, confirming that the action is silently bypassed. Together, these results show that visually plausible future prediction can emerge from dynamics-agnostic shortcuts rather than action-induced causal structure.

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

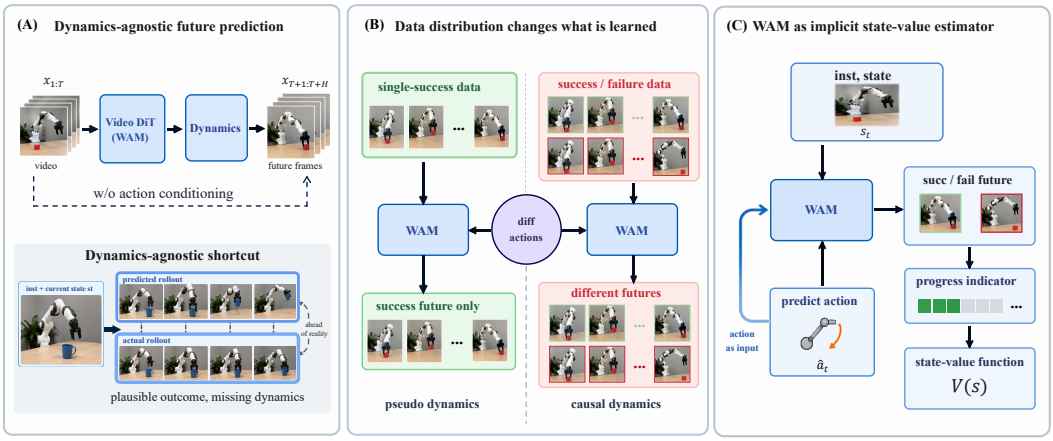


Figure 1: **Overview.** (A) *Dynamics-agnostic future prediction*: without action conditioning, the WAM still emits visually plausible futures, but they are outcome-level snapshots insensitive to temporal order and to the action's consequences (Sec. 5). (B) *Data distribution changes what is learned*: trained on unimodal single-success data, the action condition is bypassed and the model reproduces a success template (pseudo dynamics); trained on statistically multimodal success/failure data, the same architecture must bind each action to its divergent future, recovering causal dynamics (Sec. 6). (C) *WAM as implicit state-value estimator*: once this binding is established, conditioning on the policy's own action lets the model imagine successful or failed futures, and its action-independent first-frame latent separates these outcomes and tracks task progress, yielding an implicit state-value function $V(s)$ without reward supervision (Sec. 7).

Furthermore, we show that action conditioning can indeed anchor future prediction in causal dynamics—once the training distribution makes the action indispensable. Concretely, training on multimodal, sub-optimal trajectories (Fu et al., 2020; Kumar et al., 2020; Kostrikov et al., 2021) raises the heterogeneity of actions observed from similar states, leading to divergent futures. Under this statistically multimodal transition distribution, the video objective cannot be minimized without consulting the action, forcing the model to bind each action to the future it actually induces. To empirically substantiate this causal binding, we further quantify the visual fidelity of the predicted videos. On successful trajectories, the expert-only model yields marginally lower PSNR and SSIM scores than its multimodal counterpart trained on augmented failure trajectories. Crucially, this performance discrepancy becomes significantly more pronounced when predicting future frames for failure trajectories, highlighting the expert-only model's inability to generalize to sub-optimal outcomes.

Beyond acquiring causal dynamics through action conditioning and multimodal training, we uncover a property in the WAM's latent space: its potential to function as an implicit state value function. By probing the first-frame latent representation encoded by the WAM, we find it capable of accurately tracking task progress and linearly separating states by their likelihood of success. In essence, this implicit value-like state evaluation — mathematically aligning with a policy-conditioned state value $V^{\pi}(s) \approx P(\text{success} \mid s, \pi)$ (Vincent et al., 2018; Sutton, 1988; Mnih et al., 2015) — is acquired entirely without reward supervision during world-model training.

Our main contributions are summarized as follows:

- We show that, without action conditioning or under unimodal expert data, future-frame prediction relies on dynamics-agnostic shortcuts, remaining weakly sensitive to both temporal ordering and the action itself.
- We establish that action conditioning is activated only when the training distribution renders the action informative, forcing the model to bind each action to its actual future rather than bypassing the condition.
- We reveal that, once activated, the action-independent latent representation encodes an implicit value-like evaluation of the current state, emerging entirely without reward supervision.

2 Related Work

2.1 Action-Conditioned World Models and Video Prediction

Learning environment dynamics has long been central to reinforcement learning and robotics. Dyna-style architectures (Sutton, 1991) used learned models to generate synthetic experience, and the Dreamer series (Hafner et al., 2019; 2020; 2023) showed that latent world models support policy learning through imagined rollouts; TD-MPC2 (Hansen et al., 2024) extended this paradigm to continuous control. In parallel, video-generation-based world models such as Genie (Bruce et al., 2024), together with embodied video predictors like GR-1/2 (Wu et al., 2024; Cheang et al., 2024), UniPi (Du et al., 2023), and LAPA (Ye et al., 2025), demonstrate that future prediction can serve as a scalable pretraining signal for manipulation. A common thread across these works is that they inject actions (or latent actions) into the predictive process and assume this conditioning grounds the prediction in action-induced dynamics. Our work does not propose a new architecture; instead, we scrutinize this very assumption. We show that action conditioning can be silently bypassed (without conditioning) or degenerate into a deterministic mapping (under unimodal expert data), and only anchors future prediction in causal dynamics under a statistically multimodal training distribution.

2.2 Learning from Suboptimal and Failure Data

Suboptimal or failed trajectories have been exploited in several lines of research. In offline reinforcement learning, CQL (Kumar et al., 2020) and IQL (Kostrikov et al., 2021) treat suboptimal data as a *constraint*, keeping the learned policy within the data support; DAgger (Ross et al., 2011) treats corrective, possibly suboptimal demonstrations as *additional supervision*; and preference-based reward learning (Christiano et al., 2017) uses success–failure comparisons to fit a separate reward model. In contrast, we use failure trajectories neither as a constraint nor as supervision for a reward head. Instead, failures act as a contrastive signal that breaks the redundancy of unimodal expert data: by making the same state admit divergent outcomes, they force the action-conditioned module to bind each action to its actual future. Moreover, we decouple understanding from acting: the shared backbone is first trained on failure trajectories, whose extreme action heterogeneity yields a statistically multimodal transition distribution, to anchor action conditioning in causal dynamics; the policy head is subsequently fine-tuned on successes alone to specialize in expert behavior.

2.3 Value Functions and Emergent Action Evaluation

Value functions and critics are classically learned *explicitly* from reward signals via temporal-difference methods (Sutton, 1988; Mnih et al., 2015) or actor-critic objectives (Mnih et al., 2016). A related line conditions generation or policies on returns or values, as in decision transformers (Chen et al., 2021) and return-conditioned reinforcement learning (Peng et al., 2019), while model-based methods estimate values by expanding rollouts with a learned dynamics model (Silver et al., 2017; Schrittwieser et al., 2020). All of these require explicit reward, return, or value targets. Our finding is of a different nature: the value function *emerges* from purely generative training. Once action conditioning is activated by statistically multimodal data, the latent space separates successful from failed futures and scores actions by their likelihood of success—functioning as an implicit value function without any reward, return, or value supervision.

3 Preliminary

We build our study upon a unified World-Action Model (WAM) that jointly performs action-conditioned video prediction and action generation within a shared transformer backbone. Concretely, the architecture combines two complementary designs: the action-generation mechanism is adopted from Fast-WAM (Yuan et al., 2026), in which a dedicated Action Expert attends exclusively to the first-frame representation, and the action-conditioning (injection) mechanism of the video backbone follows Cosmos (Agarwal et al., 2025), in which candidate actions are injected into the generative process. This section fixes notation and details the architecture and the training objectives.

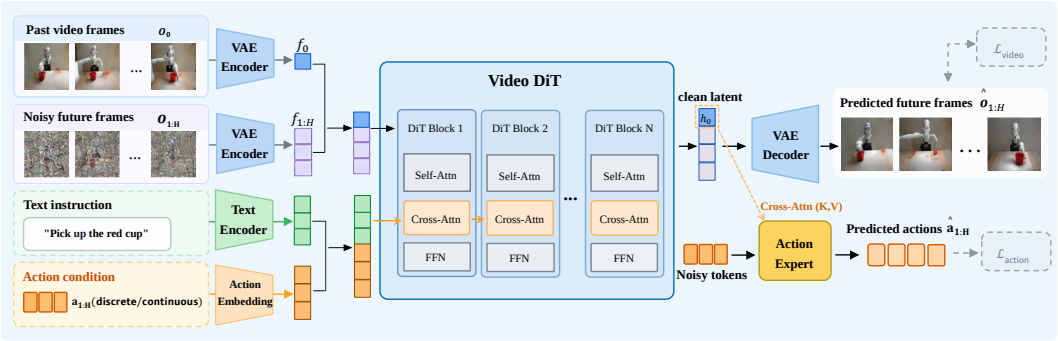


Figure 2: Overview of the action-conditioned World-Action Model. A shared Video DiT backbone processes visual observations and performs action-conditioned future prediction; an Action Expert generates robot actions by cross-attending to the key–value cache of the DiT's first-frame representation h_0, grounding policy generation in representations of the current state without rollout-based planning. The representation h_0 is precisely the latent we probe in Sec. 7.

3.1 Architecture Overview

The model processes a trajectory $\tau = (o_{0:H}, a_{0:H-1})$, where o_t denotes the visual observation at step t and a_t the robot action. It consists of two tightly coupled components:

Video DiT (world modeling). The observations o_t are tokenized into a latent space. Following the Cosmos action-conditioning design (Agarwal et al., 2025), the Video DiT is conditioned on the action sequence $a_{0:H-1}$ and predicts the latent representations of the future frames $o_{1:H}$.

Action Expert (policy generation). Following the Fast-WAM design (Yuan et al., 2026), the Action Expert operates in a reactive yet grounded manner rather than planning over predicted futures: it generates $\hat{a}_{0:H-1}$ by cross-attending to the key–value cache produced when the shared backbone encodes the *first frame*, i.e., the forward pass that yields the first-frame representation h_0. Reusing this cached context lets the policy ground action generation in physics-aware representations of the current state while bypassing explicit rollout-based planning.

3.2 Training Objectives

Joint training combines two objectives.

Action-conditioned video prediction. We employ a flow-matching objective. Let $\epsilon \sim \mathcal{N}(0, I)$ be Gaussian noise and $o^{\epsilon}_{1:H}$ the noised latents of the future frames. The Video DiT ϵ_θ is trained to predict the noise (or velocity field) conditioned on the initial observation and the action sequence:

$$\mathcal{L}_{\text{video}} = \mathbb{E}_{\tau,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(o^{\epsilon}_{1:H},\, o_0,\, a_{0:H-1})\right\|^2\right]. \tag{1}$$

Note, however, that faithfully predicting future frames does not by itself imply that the model has internalized the causal dynamics linking actions to their consequences; the visual shortcuts that can silently undermine this objective are diagnosed in Sec. 5.

Action generation. The Action Expert π_ϕ generates the action sequence from the first-frame representation h_0 extracted by the shared backbone, optimized with a regression loss:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{\tau}\left[\left\|a_{0:H-1} - \pi_\phi(h_0)\right\|_1\right]. \tag{2}$$

The total objective is a weighted sum:

$$\mathcal{L}_{\text{total}} = \lambda_V \mathcal{L}_{\text{video}} + \lambda_A \mathcal{L}_{\text{action}}, \tag{3}$$

where λ_V, λ_A balance the two terms, so the shared representation is simultaneously optimized for predicting physical consequences and generating executable actions.

4 EXPERIMENTAL SETUP

This section details the data, tasks, and training protocols used throughout our experiments.

Tasks and environment. We evaluate on six tabletop manipulation tasks instantiated in the RoboTwin simulation platform(Chen et al., 2025), spanning diverse object geometries and interaction patterns: *Adjust Bottle*, *Grab Roller*, *Move Can Pot*, *Move Playing-card Away*, *Pick Diverse Bottles*, and *Pick Dual Bottles*. Each task requires a multi-step policy to complete within a fixed horizon.

Expert demonstrations. For each task, we collect a set of expert demonstrations $\mathcal{D}_{\text{succ}}$ via scripted policies or teleoperation, ensuring every trajectory reaches a successful outcome; the collection episodes are drawn from random seeds starting from seed 0. These demonstrations serve as the sole training data for the conventional (expert-only) baseline and for Stage 2 fine-tuning in our two-stage paradigm.

Failure trajectories. To construct the failure set $\mathcal{D}_{\text{fail}}$, we roll out the policy in the simulator and inject controlled noise perturbations into the executed actions at random decision steps, so that the rollout deviates from the successful trajectory and terminates in a failure; only failed rollouts are retained. These rollouts are collected over random seeds starting from seed 200, disjoint from the expert seeds. This procedure yields trajectories with extreme *action heterogeneity*: the same or similar initial states can lead to drastically different futures depending on the perturbation applied.

Two-stage training protocol. Our training paradigm decouples causal understanding from policy execution:

- **Stage 1 (Understanding):** The shared Video DiT backbone is trained *exclusively* on failure trajectories $\mathcal{D}_{\text{fail}}$ to learn action-induced dynamics. No successful demonstrations are used at this stage. The action-conditioned module (ac-cond) is trained jointly with the video prediction objective (Eq. 1).
- **Stage 2 (Acting):** The Action Expert is fine-tuned on expert demonstrations $\mathcal{D}_{\text{succ}}$ alone, inheriting the causal representations acquired in Stage 1 while specializing in expert behavior.

We denote the failure-to-expert data ratio as $r = |\mathcal{D}_{\text{fail}}|/|\mathcal{D}_{\text{succ}}|$, where $\mathcal{D}_{\text{succ}}$ serves as a reference scale for reporting the volume of failure data used in Stage 1.

Evaluation protocol. Downstream success is measured by executing the learned policy on held-out evaluation episodes and reporting the per-task and mean success rate; evaluation episodes use 100 distinct random seeds starting from seed 430000. Prediction quality is assessed via SSIM and PSNR between predicted and ground-truth future frames. The expert, failure, and evaluation collections start from seeds 0, 200, and 430000 respectively and occupy mutually disjoint seed ranges, ensuring that evaluation initial states never overlap with those seen during data collection or training.

5 WHEN ACTION CONDITIONING FAILS

Before showing when action conditioning works, we characterize how it fails under the conventional training recipe (i.e., on the unimodal expert distribution $\mathcal{D}_{\text{succ}}$). We diagnose this failure from two perspectives: the lack of causal structure in predicted futures, and the redundancy of the action condition.

5.1 PREDICTED FUTURES LACK CAUSAL STRUCTURE

We first examine the predicted futures of the *uncond* model. If future-frame prediction captured action-induced dynamics, the predicted sequence should exhibit a strict temporal evolution. We test this by shuffling the temporal order of the predicted future frames before passing them to downstream evaluation during training,while keeping all other settings unchanged. As Tab. 1 shows,

Table 1: Temporal insensitivity of the *uncond* model. Shuffling the predicted future frames degrades performance only marginally, indicating the lack of temporal causal structure.

Model & Setting	Adj. Bot.	Grab Rol.	Move Can	Move Card	Pick Div.	Pick Dual	Avg. Succ.
Original futures (uncond)	0.67	0.59	0.47	0.06	0.31	0.25	0.39
Shuffled futures (uncond)	0.64	0.57	0.39	0.09	0.27	0.19	0.36

Table 2: Redundancy of action conditioning in *ac-cond*. Disrupting the action input at inference yields no degradation, proving the model ignores the action and acts as a deterministic mapping.

Model & Setting	Adj. Bot.	Grab Rol.	Move Can	Move Card	Pick Div.	Pick Dual	Avg. Succ.
uncond (Baseline)	0.67	0.59	0.47	0.06	0.31	0.25	0.39
ac-cond (Intact actions)	0.59	0.56	0.42	0.09	0.24	0.27	0.36
ac-cond (Action removed)	0.65	0.47	0.42	0.06	0.26	0.23	0.35
ac-cond (Action random.)	0.64	0.65	0.47	0.08	0.25	0.11	0.37

shuffling the futures changes downstream success only marginally. This confirms that the predicted frames lack causal temporal structure and behave as a collection of spatially plausible snapshots; combined with the action-insensitivity in Sec. 5.2, this indicates the vanilla model relies on outcome-level shortcuts rather than action-induced dynamics.

5.2 Action Condition is Redundant under Unimodal Data

A natural remedy is to inject action information via *ac-cond*. However, under unimodal expert data, this fails to restore causal sensitivity. We evaluate the *ac-cond* model by disrupting its action input at inference time (removing or randomizing the actions). Tab. 2 shows that *ac-cond* with intact actions yields no downstream gain over *uncond*. Furthermore, removing or randomizing the actions does not degrade its performance, and the AC pathway receives negligible gradients during training. This proves the action module is effectively bypassed: it degenerates into a deterministic mapping that ignores the action entirely.

Taken together, the two experiments indicate that the failure of action conditioning is not an architectural deficiency but a property of the training distribution. The predicted futures lack causal structure (Sec. 5.1), and the action condition is entirely ignored (Sec. 5.2). The unimodality of $\mathcal{D}_{\text{succ}}$ removes the causal signal that conditioning is supposed to exploit: when every action leads to the same future, no action input can recover outcome-level causality. This motivates the question addressed in the next section: which training distribution restores the causal signal?

6 When Action Conditioning Takes Effect

Sec. 5 attributed both failure modes to the unimodal nature of the expert distribution. To avoid any confusion with sensory multi-modality (e.g., combining vision and language), we use *statistical multimodality* to refer strictly to the presence of multiple modes in the transition distribution, where a single state admits divergent futures. This unimodality stems from the *action homogeneity* of expert data, where only optimal actions are present and the future is entirely predictable from the state. We now test the converse: does introducing *action heterogeneity* to create statistical multimodality recover the causal signal? We answer in two steps. First, we sweep the composition of the training mixture and show that statistical multimodality is the activation condition for action conditioning (Sec. 6.1). Second, we verify that the activated conditioning corresponds to a genuine causal binding rather than a spurious dependence, using an outcome-stratified fidelity probe (Sec. 6.2).

6.1 Statistical Multimodality as the Activation Condition

As detailed in Sec. 4, our two-stage paradigm first trains the shared backbone exclusively on failure trajectories $\mathcal{D}_{\text{fail}}$ (Stage 1), then fine-tunes the Action Expert on expert demonstrations $\mathcal{D}_{\text{succ}}$ (Stage 2). Unlike expert data, $\mathcal{D}_{\text{fail}}$ exhibits extreme *action heterogeneity*, naturally creating a *statistically multimodal* transition distribution where the same state can unfold into drastically different futures.

Table 3: Activation and scaling of action conditioning via failure data. At $r = 0/50$ the ac-cond model offers no advantage over uncond (mean 0.36 vs 0.39); as r grows, ac-cond improves monotonically (0.36 $\rightarrow$ 0.40 $\rightarrow$ 0.45 $\rightarrow$ 0.52) while uncond remains flat (0.39 at $r = 0/50$ vs 0.38 at $r = 50/50$), showing that failure data helps only when action conditioning is present to exploit it. Volume control: doubling expert demonstrations at $r = 0/100$ raises the mean only to 0.44 (uncond), far below the ac-cond gain at $r = 50/50$, confirming the effect is not a data-volume artifact.

r	Model	Adj. Bot.	Grab Rol.	Move Can	Move Card	Pick Div.	Pick Dual	Mean
0/50	uncond	0.67	0.59	0.47	0.06	0.31	0.25	0.39
50/50	uncond	0.69	0.59	0.35	0.06	0.29	0.28	0.38
0/50	ac-cond	0.59	0.56	0.42	0.09	0.24	0.27	0.36
10/50	ac-cond	0.64	0.62	0.44	0.11	0.28	0.32	0.40
20/50	ac-cond	0.72	0.68	0.46	0.13	0.33	0.38	0.45
0/100	uncond	0.79	0.68	0.41	0.14	0.34	0.29	0.44
50/50	ac-cond	**0.81**	**0.79**	**0.49**	**0.16**	**0.39**	**0.45**	**0.52**

To characterize how the volume of failure data in Stage 1 affects learning, we systematically vary the ratio $r = |\mathcal{D}_{\text{fail}}|/|\mathcal{D}_{\text{succ}}|$ (defined in Sec. 4). This sweep serves a dual purpose: first, to identify the *activation condition* of action conditioning; second, to verify whether such noise-perturbed failure data exhibits *data scaling behavior*—that is, whether injecting more failure trajectories enables the model to learn more diverse and comprehensive action-induced dynamics.

Tab. 3 reveals two key findings. First, the transition from $r = 0/50$ to $r > 0/50$ acts as a sharp activation switch: ac-cond offers no advantage over uncond at $r = 0/50$ (reproducing Sec. 5.2), but immediately surpasses the uncond baseline as failure data is introduced. Second, as the scaling factor r continues to grow, the downstream performance of the ac-cond model improves consistently, asymptotically approaching its upper bound. This scaling behavior indicates that a larger volume of failure trajectories allows the model to cover a broader spectrum of action heterogeneity, thereby refining its action-to-future binding towards a more complete understanding of the causal dynamics. Statistical multimodality, driven by action heterogeneity, is therefore not merely a binary activation condition, but a scalable source of causal signal.

6.2 Action-to-Future Binding

Activation shows that the prediction now depends on the action; it does not by itself show that this dependence is causal, i.e., that the predicted future tracks the action the way the environment does. The most direct test of such binding is to ask whether the model's imagination of its own action's consequence matches what actually happens. For each replan window, we therefore condition the Video DiT on the policy's own executed action a and compare the imagined future $\hat{o}(a)$ with the future the environment actually produced; we refer to this agreement as *prediction fidelity*. To examine how fidelity behaves under different circumstances, we further split the rollouts by whether the episode eventually succeeds or fails.

On successful rollouts, both regimes track reality reasonably well (Tab. 4): the executed actions are sensible and the resulting futures follow a common, smooth pattern that even a bypassed model can reproduce. The two regimes diverge sharply on failed rollouts. The expert-only model loses fidelity there (mean PSNR drops from 17.33 to 16.22): having seen only successful outcomes, it cannot anticipate that the same state, under an errant action, unfolds into a bad future, and instead reproduces its success template. The failure-pretrained model, by contrast, keeps high fidelity on failed rollouts—indeed higher than on successful ones (mean PSNR 20.06)—and substantially exceeds the expert-only model on the very same failed rollouts. A model that reproduces a template fits only the common mode; a model that binds actions to futures predicts the consequences of *any* action it is given, including the divergent, catastrophic ones. This is the signature of genuine action-to-future binding: under statistical multimodality, action conditioning anchors future prediction in causal dynamics.

Table 4: Prediction fidelity between the imagined future $\hat{o}(a)$ and the executed future. Rollouts are split by eventual outcome. On successful rollouts (top), both models track reality well. On failed rollouts (bottom), only the failure-pretrained model maintains high fidelity, confirming that it binds actions to their actual divergent futures, whereas the expert-only model reproduces a success template and loses fidelity. We report PSNR (↑) / SSIM (↑).

Regime	Model	Adj. Bot.	Grab Rol.	Move Can	Move Card	Pick Div.	Pick Dual	**Mean**
Successful Rollouts	ac-cond (on $\mathcal{D}_{\text{succ}}$)	16.73 / 0.814	15.78 / 0.768	21.67 / 0.904	18.44 / 0.851	18.34 / 0.830	15.61 / 0.786	17.33 / 0.813
	ac-cond (with failures)	17.51 / 0.826	16.56 / 0.775	22.70 / 0.911	19.36 / 0.865	17.51 / 0.819	18.11 / 0.837	18.11 / 0.824
Failed Rollouts	ac-cond (on $\mathcal{D}_{\text{succ}}$)	19.06 / 0.861	14.79 / 0.740	19.98 / 0.888	17.60 / 0.852	13.88 / 0.731	13.08 / 0.718	16.22 / 0.798
	ac-cond (with failures)	**23.61 / 0.892**	**16.88 / 0.778**	**24.25 / 0.924**	**21.44 / 0.891**	**17.69 / 0.809**	**16.89 / 0.810**	**20.06 / 0.855**

Table 5: Linear-probe prediction of eventual success from the action-free latent h_0, aggregated over decision steps. Multimodal training makes the outcome signal stronger and consistent across tasks, whereas the expert-only backbone is weaker and erratic.

Model	Metric	Adj. Bot.	Grab Rol.	Move Can	Move Card	Pick Div.	Pick Dual	Mean ↑
ac-cond (on $\mathcal{D}_{\text{succ}}$)	Acc	0.580	0.547	0.620	0.422	0.546	0.499	0.536
	AUROC	0.703	0.606	0.662	0.381	0.636	0.502	0.582
ac-cond (with failures)	Acc	0.629	0.625	0.576	0.670	0.712	0.594	0.634
	AUROC	0.709	0.681	0.691	0.633	0.735	0.701	0.692

7 The Emergent Implicit State-Value Function

Sec. 6 showed that failure-driven statistical multimodality anchors future prediction in causal dynamics. Here we ask what this binding leaves in the shared representation. We study h_0, the first-frame encoding of the Video DiT. In all experiments below we extract h_0 by feeding a null action (disabling the action-conditioning branch) and running a single forward pass without any denoising, so h_0 depends only on the current observation o_0; any information it carries about success or failure must therefore come from what the backbone has learned about the scene itself. We examine two things: whether h_0 predicts the eventual outcome and how this ability differs across training regimes (Sec. 7.1), and whether the ability reflects learned task dynamics rather than superficial visual cues (Sec. 7.2).

7.1 The Latent Predicts Whether the Scene Will Succeed

We train a linear probe on h_0 to predict whether the current scene will eventually lead to success. The probe sees only h_0, never the action, so it can only use information already present in the representation. Under failure-driven pretraining the probe predicts the eventual outcome above chance on all six tasks (Tab. 5, mean accuracy 0.634, AUROC 0.692).

The informative comparison is across regimes. A backbone trained only on $\mathcal{D}_{\text{succ}}$ also carries some outcome information (mean AUROC 0.582), but it is erratic: it reaches 0.703 on Adjust Bottle yet falls to 0.381 on Move Playing-card and 0.502 on Pick Dual Bottles. Under failure-driven pretraining the same probe stays above chance on every task (AUROC 0.633–0.735). Failure-driven multimodality therefore does not create outcome information from nothing; it is what makes the outcome signal in h_0 strong and consistent across tasks.

Because the probe is trained to predict eventual success under the evaluated policy, its score estimates $P(\text{success} \mid s, \pi)$; with success as the only reward this is the policy-conditioned state value $V^{\pi}(s)$. The AUROC above 0.5 on all tasks confirms the score orders scenes by their likelihood of success rather than acting as a hard yes/no decision. We do not claim a precisely calibrated value function: the probe's moderate accuracy shows the signal is coarse, and its evidential value lies in the gap between regimes.

7.2 The Latent Also Tracks How the Task Unfolds

Beyond predicting whether a state will succeed, we ask whether h_0 also represents how far the task has progressed. We project h_0 of successful executions onto the leading principal components and

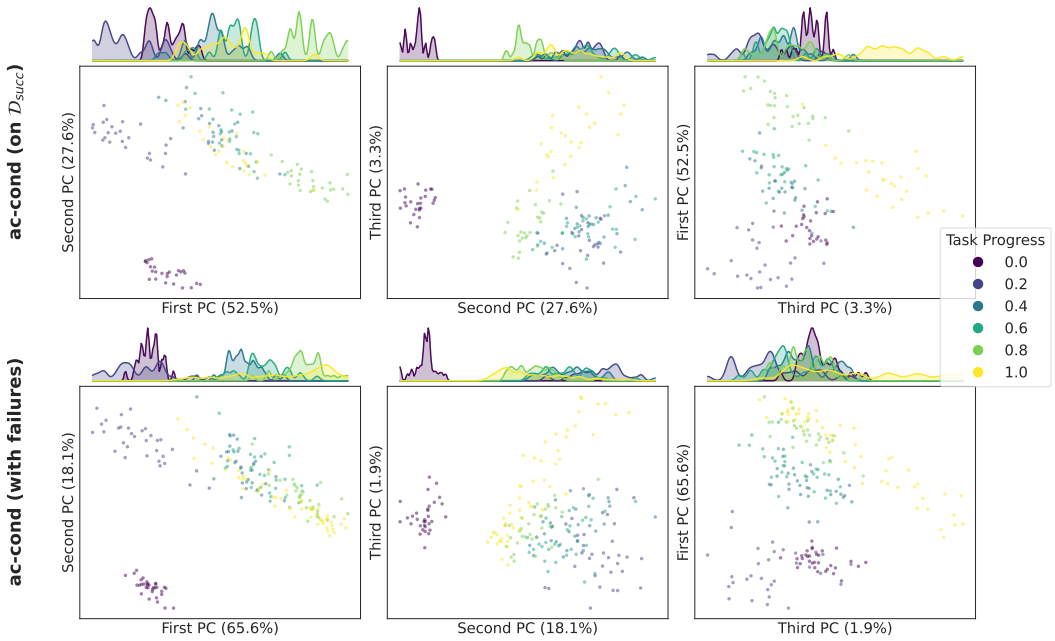


Figure 3: PCA projections of the action-free latent h_0 from successful executions, colored by normalized task progress, for the expert-only (top) and failure-pretrained (bottom) backbones. The failure-pretrained latent forms a smoother, more monotonic progress manifold.

color each point by its normalized task progress (Fig. 3). We use only successful executions so that the final outcome is held constant; any structure along progress therefore reflects the task's evolution rather than its outcome. Under multimodal training, successive progress bins form a smooth, monotonic path from 0 to 1; under expert-only pretraining the path is less ordered. Thus the same binding that makes h_0 predictive of outcome also organizes h_0 along the task's temporal evolution: the latent tracks not only where the scene will end, but where it currently stands.

In summary, anchoring action conditioning in causal dynamics leaves a useful byproduct in the latent: h_0 predicts whether the current scene will succeed and tracks how far the task has progressed; together these show the latent has absorbed the task's temporal and causal structure. This state-value-like ability is never trained with any reward or success label; it appears as a side effect of generative world-model training on statistically multimodal failure data, and it needs no reward, no critic, and no action input at test time.

8 Conclusion

In this work, we challenged the assumption that future-frame prediction in World Action Models (WAMs) inherently captures causal dynamics. We demonstrated that under unimodal expert data, action conditioning is silently bypassed, reducing the model to a dynamics-agnostic predictor. However, by introducing statistically multimodal failure trajectories, we showed that action conditioning is activated, forcing the model to bind actions to their divergent futures and thereby recovering genuine causal dynamics. Strikingly, this causal binding triggers the emergence of an implicit state-value function in the action-independent latent space, which tracks task progress and evaluates success likelihood entirely without reward supervision. While our current failure data relies on simulation-based noise injection, future work will explore real-world sub-optimal data collection and investigate how to explicitly harness this emergent latent value signal to guide downstream policy optimization and model-predictive control.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

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.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Pim De Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. *Advances in neural information processing systems*, 32, 2019.

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.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

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

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PmLR, 2016.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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.

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.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Francois-Lavet Vincent, Henderson Peter, Islam Riashat, G Bellemare Marc, and Pineau Joelle. An introduction to deep reinforcement learning. *Foundations and trends® in machine learning*, 11 (3-4):219–354, 2018.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

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.

A AI USE STATEMENT

In this work, we used generative AI tools (specifically, Large Language Models) for several tasks that require mandatory disclosure, including assisting with translation, proposing and refining hypotheses regarding the role of action conditioning in causal learning, and interpreting the results of latent space probing. We have not used generative AI tools to generate synthetic datasets, develop the core theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, or assist in the writing of proofs. Furthermore, tasks such as cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work, as we utilized established benchmark datasets (e.g., RoboTwin) and standard quantitative metrics (e.g., PSNR, SSIM).

Additionally, we used generative AI tools for tasks with recommended disclosure. Specifically, we utilized LLMs to edit the research paper to improve readability and academic tone, draft parts of the manuscript, summarize and analyze existing literature, create and edit software code for model implementation and training pipelines, suggest a structure for the research paper, and propose titles

and keywords. We have not used generative AI tools to create or modify the scientific figures and images, nor to formulate questions for surveys or interviews.

We have rigorously reviewed all AI-assisted work. Specifically, all LLM-generated and refined text was manually checked by the authors to ensure factual accuracy, logical consistency, and alignment with our core scientific claims. The hypotheses and experimental designs were independently conceived by the authors and verified through extensive empirical testing. Any LLM-assisted code was manually reviewed, debugged, and tested for correctness by the authors. We take full responsibility for the final content of this work, including all text, claims, methodology, and artifacts produced with the aid of generative AI.