PREDICTION SERVES ACTION: LEARNING ACTION EFFECTS IN WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World Action Models (WAMs) improve action learning by introducing future prediction as auxiliary supervision. Existing WAMs, however, largely retain the complete future observation as the predictive target, even when future prediction is conditioned on robot actions. Such targets contain both scene changes and substantial persistent content, distributing supervision across the complete future observation even though different components might contribute differently to action learning. We therefore introduce Action Effect, a predictive target that focuses on the changes and consequences associated with an action and provides more targeted future supervision. Based on this idea, we propose Effect-WAM, a lightweight WAM that models Action Effects directly within a single DiT action-generation pathway, without a separate future branch. Effect-WAM employs a tightly coupled Propose-Anticipate-Act (PAA) mechanism that forms an intermediate action hypothesis, anticipates the corresponding effects, and uses the resulting Effect representations to refine subsequent action generation. We further construct complementary 2D semantic and robot-centric 3D geometric supervision to learn structured Action Effect representations. Experiments on LIBERO and RoboTwin 2.0, together with real-world evaluations, show strong manipulation performance with a lightweight architecture, supporting Action Effect modeling as an effective alternative to complete future generation. Code will be released.

1 INTRODUCTION

Robot manipulation requires understanding task goals and coordinating physical interactions with objects to achieve desired outcomes in the environment. Vision-Language-Action (VLA) models Qu et al. (2025); Ghosh et al. (2024) address this task by mapping visual observations and language instructions to robot actions. However, direct action supervision alone does not explicitly model how the scene evolves over time. Such modeling is important for manipulation Wu et al. (2024); Hu et al. (2025), where action generation depends on how object states evolve through interaction. To this end, World Action Models (WAMs) extend action learning with an auxiliary future modeling branch that provides additional supervision for modeling scene evolution and physical dynamics.

Within this future modeling branch, existing WAMs Qiu et al. (2026); Chen et al. (2026a); Lin et al. (2026) mainly differ in whether future prediction is conditioned on the demonstrated robot action. A common approach Yuan et al. (2026) predicts the future observation from the current scene and task context, as illustrated in Figure 1(a). Other approaches Pan et al. (2026); Zhu et al. (2025b) further condition future prediction on the demonstrated action, yielding the future observation corresponding to that action, as illustrated in Figure 1(b). Although these approaches differ in their conditioning information, they retain the same predictive target: the complete future observation. This target contains not only the changes that occur over the prediction horizon, but also substantial visual content that persists from the current scene, including background appearance, object texture, and other largely unchanged details. Consequently, the auxiliary objective distributes supervision across the entire future observation, even though different components of the future need not contribute equally to action learning.

This mismatch leads us to reconsider the predictive target itself: what information about the future is most useful for action learning? We hypothesize that, for manipulation control, modeling the changes and consequences associated with an action is more useful than reconstructing the complete

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.

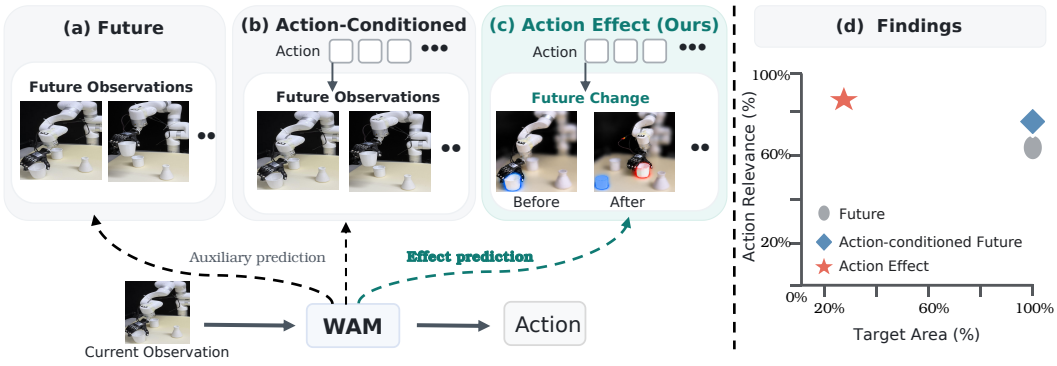

Figure 1: **Motivation and Findings. (a-c)** Existing WAMs model the complete future, with or without action conditioning, while we focus on Action Effects that capture action-associated changes. **(d)** Action Effect achieves higher Action Relevance with a substantially smaller Predictive Target Area than complete future modeling.

future observation. We refer to this information as the "Action Effect" (Figure 1(c)). By focusing on changes, Action Effect supervision emphasizes the parts of the scene that evolve through interaction, rather than distributing supervision across the whole future representation. These changes more directly reflect the relationship between an action and resulting scene evolution, making the auxiliary objective better aligned with action learning.

To test this hypothesis, we compare three predictive targets: Future, Action-conditioned Future, and Action Effect, as shown in Figure 1(d). We jointly examine Predictive Target Area and Action Relevance to assess how much information each target covers and how relevant this information is to action learning. Our results show that Action Effect achieves the highest Action Relevance while covering a substantially smaller area than the complete future. For policy learning, higher action relevance is important because it indicates that the supervised representation is more predictive of expert actions, making the auxiliary supervision more aligned with action learning Ma et al. (2023b); Yang et al. (2025); Ma et al. (2023a). The utility of future supervision therefore depends not only on whether future prediction is conditioned on action, but also on what information about the future is modeled. This motivates us to model Action Effects directly and use the anticipated effects to guide action generation.

Building on this insight, we introduce Effect-WAM, which replaces complete future generation with Action Effect modeling through a tightly coupled Propose-Anticipate-Act (PAA) mechanism. Specifically, PAA couples Action Effect modeling with action generation so that anticipated consequences can guide action refinement. In the Propose process, we form an action hypothesis from the action representation at an intermediate stage of the DiT. This hypothesis captures the evolving action intent before the final action is generated. In the Anticipate process, two horizon-specific Effect tokens read this hypothesis through an Action-to-Effect interaction, forming latent representations of the anticipated consequences. We supervise these Effect representations with structured Action Effect labels constructed from complementary 2D semantic changes and robot-centric 3D geometric changes. Finally, in the Act process, the action representation reads the learned Effect tokens through a gated Effect-to-Action interaction, using the anticipated consequences to refine the remaining action generation. At inference time, we remove the auxiliary prediction heads, while the Effect tokens and PAA interactions remain active without complete future generation.

Extensive experiments across multiple simulated benchmarks and real-world tasks demonstrate the effectiveness and generality of Effect-WAM, achieving strong manipulation performance.

Our main contributions are summarized as follows:

- We introduce Action Effect as a predictive target for manipulation, shifting future supervision from reconstructing complete future observations toward modeling action-associated changes and consequences.

- We propose Effect-WAM with a tightly coupled Propose-Anticipate-Act (PAA) mechanism, which learns structured Action Effects from intermediate action hypotheses and uses the anticipated consequences to refine action generation.
- We evaluate Effect-WAM across extensive simulation benchmarks and real-world robotic tasks, demonstrating its effectiveness and robustness across diverse manipulation settings.

2 Related Work

Vision-Language-Action Models. Vision-Language-Action (VLA) Zhao et al. (2025); Pertsch et al. (2025); Liu et al. (2025) models leverage large-scale vision-language pretraining to learn generalist robot policies from visual observations and language instructions. RT-2 Zitkovich et al. (2023) represents robot actions as language-model tokens, enabling vision-language knowledge to transfer to robotic control. OpenVLA Kim et al. (2025b) further develops an open VLA foundation model by combining pretrained vision-language representations with large-scale robot demonstrations. More recently, π_0 Black et al. (2025b) introduces flow matching for continuous action generation, supporting expressive action chunks for dexterous manipulation. Beyond direct action prediction, our Effect-WAM further incorporates Action Effect modeling into policy learning.

World Action Models. World Action Models (WAMs) Zhu et al. (2025a); Li et al. (2025) extend robot policy learning with predictive modeling of future world evolution, providing additional supervision for physical dynamics and temporal structure. Early efforts such as DreamZero Ye et al. (2026) build on pretrained video diffusion to jointly model future video and robot actions, while Motus Bi et al. (2026) further unifies world modeling, video generation, and action prediction within a latent action world model. Building on this paradigm, Fast-WAM Yuan et al. (2026) disentangles training-time video modeling from test-time future imagination, showing that future supervision can benefit action learning without explicit future generation at inference. Faster-WAM Zhao et al. (2026b) subsequently explores more efficient ways to incorporate future information into action generation. More recently, SelfWAM Pan et al. (2026) strengthens the connection between future modeling and robot behavior by explicitly conditioning future RGB prediction on demonstrated actions and introducing self-mask supervision to improve action sensitivity. Despite these advances in architecture, conditioning, and the use of future representations, existing WAMs largely retain the future state or observation as the predictive target. In contrast, Effect-WAM shifts predictive supervision from complete future states to action-associated changes captured by Action Effects.

Action-Relevant Representations. Learning representations that emphasize information useful for control has long been explored in robot learning. R3M Nair et al. (2023) learns reusable visual representations from large-scale human videos for downstream manipulation. Moving toward predictive representations, DINO-WM Zhou et al. (2025) models future dynamics directly in pretrained DINO feature space rather than reconstructing pixels. More recently, VLAFlow Xia et al. (2026) uses future latent alignment to improve state-transition and action-outcome modeling, while SG-WAM Zhao et al. (2026a) predicts action-conditioned future states in a geometry policy representation space. Action-Effect Memory Zhou et al. (2026) instead learns compact action-conditioned temporal representations from vision-action histories for downstream control. Different from them, we construct action-relevant representations around Action Effects, grounding them in complementary 2D semantic changes and 3D geometric transitions Lu et al. (2025); Zhen et al. (2025).

3 Methodology

3.1 Action Effect Construction

As introduced in Section 1, we define an *Action Effect* as the changes and consequences associated with an action, capturing how the scene changes relative to its current state over a future horizon. To operationalize this concept for policy learning, this section constructs supervision for learning Action Effect representations. Since standard robot demonstrations Liu et al. (2023); Open X-Embodiment Collaboration (2024) provide observation and action trajectories but no direct Action Effect labels, we derive the required labels from the observed changes between temporally paired

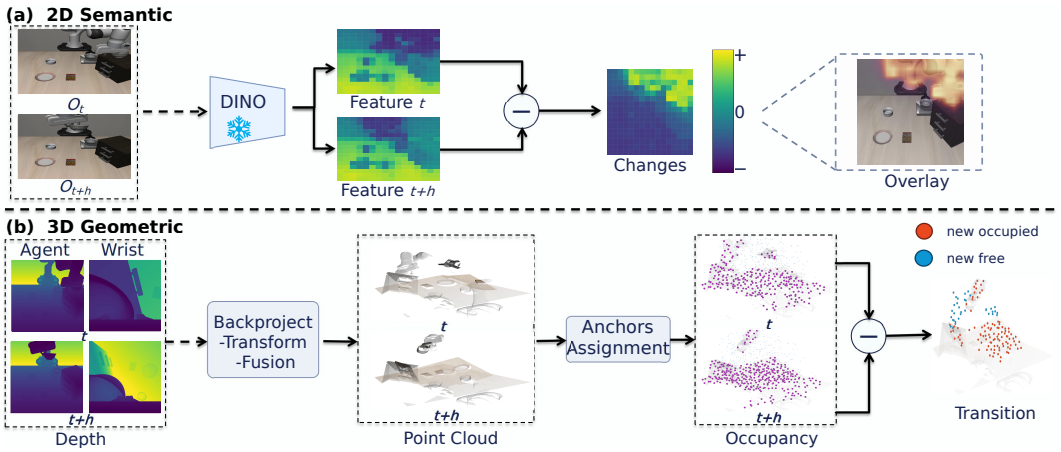

Figure 2: **Construction of Action Effect supervision. (a)** 2D semantic changes are obtained from differences between current and future DINO features. **(b)** 3D geometric changes are obtained from occupancy transitions at fixed canonical anchors in the robot-centric frame.

observations. These changes contain complementary semantic and geometric information: the former describes what changes in the scene, while the latter captures how its spatial structure changes Ke et al. (2025). We therefore construct Action Effect supervision using **2D semantic changes** together with **robot-centric 3D geometric changes**.

For a robot demonstration, we define the trajectory as $\tau = \{(\mathbf{o}_t, \mathbf{a}_t)\}_{t=1}^{T}$, where $\mathbf{o}_t$ and $\mathbf{a}_t$ denote the robot observation and executed action at time step t, respectively. At each time step t, we construct Action Effect labels at two future horizons $h \in \mathcal{H} = \{16, 32\}$ from the observations $\mathbf{o}_t$ and $\mathbf{o}_{t+h}$.

2D Semantic Change. As illustrated in Figure 2(a), for the 2D component, we use spatial DINO features Siméoni et al. (2026) extracted from the current observation $\mathbf{o}_t$ and the future observation $\mathbf{o}_{t+h}$ to capture changes in visual semantics. Specifically, we compute the token-wise difference between the corresponding DINO feature maps and use it as the 2D Action Effect label $\mathbf{y}_{t,h}^{\text{2D}}$. This label describes the semantic changes relative to the current observation over horizon $h \in \mathcal{H}$.

3D Geometric Change. While 2D DINO features capture semantic changes in the scene, they do not explicitly represent the underlying 3D spatial structure. We therefore complement the 2D representation with robot-centric geometric changes, as shown in Figure 2(b). At each time step, we construct a robot-centric point cloud by back-projecting the synchronized agent-view and wrist-view depth observations, transforming the reconstructed points into the robot-base coordinate frame, and fusing the two views. Applying the same procedure at time steps t and $t + h$ places the current and future geometry in a shared spatial reference frame, enabling direct comparison across time.

Within the robot-base workspace, we first sample a fixed set of m canonical 3D anchors and keep their locations unchanged across all observations. These anchors serve as shared spatial query points rather than points sampled from the observed geometry. For each observation, we use the robot-centric point cloud together with the free-space information provided by the depth observations to assign each anchor one of three occupancy states: `FREE`, `OCCUPIED`, or `UNKNOWN`. Since the same anchors are used at time steps t and $t + h$, their occupancy states establish a direct anchor-wise correspondence across time. We then compare the state of each valid anchor between the two time steps to construct the 3D Action Effect label $\mathbf{y}_{t,h}^{\text{3D}}$, indicating whether the anchor becomes occupied, becomes free, or remains unchanged.

3.2 The overview of Effect-WAM

An overview of Effect-WAM is shown in Figure 3. Given the current observation $\mathbf{o}_t$ and language instruction $\mathbf{l}$, Effect-WAM generates an action chunk $\hat{\mathbf{a}}_{1:K}$ of length K. The observation, language instruction, and action variables are processed by the visual, text, and action encoders, respectively,

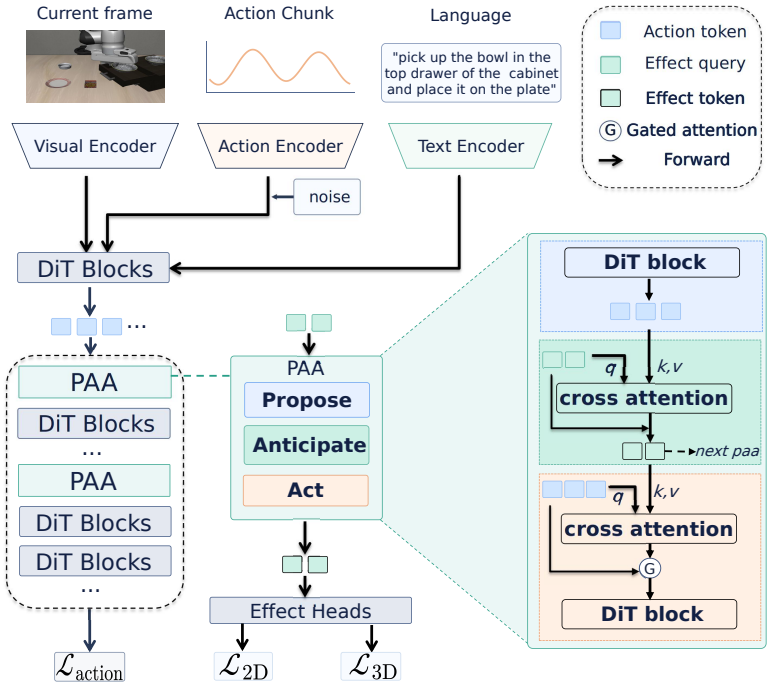

Figure 3: Overview of Effect-WAM. Effect-WAM maintains a single DiT action-generation pathway and introduces Action Effect modeling through Propose-Anticipate-Act (PAA). Propose forms an Intermediate Action Hypothesis, Anticipate conditions horizon-specific Effect queries on this hypothesis to form Effect representations, and Act feeds the anticipated Effects back to the action stream for further refinement. During training, the Effect representations are supervised by complementary 2D semantic and robot-centric 3D geometric Action Effect labels.

and then passed to the DiT backbone Peebles & Xie (2023). Effect-WAM retains a single action-generation pathway and introduces Action Effect modeling into the intermediate computation of DiT through the **Propose-Anticipate-Act (PAA)** mechanism. During training, the 2D semantic and 3D geometric Action Effect labels defined in Section 3.1 provide auxiliary supervision for learning the Effect representations.

3.3 Propose-Anticipate-Act (PAA)

The Propose-Anticipate-Act (PAA) mechanism couples Action Effect modeling with action generation inside the DiT backbone, allowing the evolving action representation to interact with its anticipated consequences during generation. PAA operates directly on the intermediate DiT representations while maintaining the action token sequence as the sole action generation stream.

Propose. Before anticipating an Action Effect, the model first forms an action hypothesis from an intermediate stage of DiT. Early representations are still dominated by incomplete action formation, while representations near the output leave limited capacity for subsequent refinement. We therefore use an intermediate action representation, denoted as $\mathbf{H}_{\text{mid}}$, where the action intent has become sufficiently established while remaining open to further adjustment. We refer to $\mathbf{H}_{\text{mid}}$ as the Intermediate Action Hypothesis. It is not a separately decoded action or the result of an additional generation pass, but an internal representation produced along the same DiT pathway. This hypothesis provides the action-dependent context for anticipating the corresponding Action Effects.

Anticipate. Given the Intermediate Action Hypothesis, we introduce two horizon-specific Effect queries to anticipate the corresponding Action Effects at $h \in \mathcal{H}$. Each query is initialized from a shared learnable Effect query together with a horizon embedding, providing a common Effect representation while preserving its temporal identity. The Effect queries then read the current action

hypothesis through Action-to-Effect (A2E) cross-attention, using action representation $\mathbf{H}_{\text{mid}}$ as the keys and values. For the Effect query $\mathbf{D}_h$ at horizon h, the update is defined as:

$$\mathbf{D}'_h = \mathbf{D}_h + \text{CrossAttn}\left(Q = \mathbf{D}_h, K = \mathbf{H}_{\text{mid}}, V = \mathbf{H}_{\text{mid}}\right), \tag{1}$$

where $\mathbf{D}'_h$ denotes the updated Effect representations. Through this interaction, the initially generic Effect queries become conditioned on the current action hypothesis, forming Effect tokens. During training, the horizon-specific 2D and 3D Action Effect supervision grounds these latent representations to encode the corresponding action-associated consequences. The Effect tokens are maintained as latent side states rather than being appended to the action-token sequence, and continue to evolve with the action representation across subsequent PAA modules.

Act. The anticipated Effect representations are then incorporated back into the action stream, so that subsequent refinement can account for the consequences associated with the current action hypothesis. We realize this interaction through Effect-to-Action (E2A) cross-attention, where the evolving action representation serves as the query and the horizon-specific Effect tokens provide the keys and values. The action representation $\mathbf{H}'_{\text{mid}}$ is updated as:

$$\mathbf{H}'_{\text{mid}} = \mathbf{H}_{\text{mid}} + w_{\text{effect}}(s)\ \text{CrossAttn}\left(Q = \mathbf{H}_{\text{mid}}, K = \mathbf{D}', V = \mathbf{D}'\right), \tag{2}$$

where $w_{\text{effect}}(s)$ is a gating coefficient that controls the contribution of the anticipated Effect representations at flow step s. It is determined by the clean-action coefficient in the flow interpolation, assigning weaker Effect feedback when the action is highly corrupted and progressively increasing its contribution as the action hypothesis becomes more reliable. The updated representation $\mathbf{H}'_{\text{mid}}$ is then passed through the subsequent DiT layers for further refinement.

PAA is applied sparsely at selected intermediate DiT layers, leaving DiT-only layers between successive interactions and after the final interaction. The Effect tokens persist across these PAA modules and are progressively updated throughout the action-generation process. After the final PAA interaction, the resulting Effect representations are decoded by the auxiliary 2D and 3D prediction heads, where the corresponding Action Effect supervision is applied.

3.4 Training Objective

Action Objective. We follow the standard flow matching objective Lipman et al. (2023) to train the DiT pathway. Given an expert action chunk $\mathbf{a}^*_{1:K}$, we sample Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and a flow step $s \sim \mathcal{U}(0, 1)$, and construct the interpolated action as:

$$\mathbf{a}^{(s)}_{1:K} = (1 - s)\mathbf{a}^*_{1:K} + s\boldsymbol{\epsilon}. \tag{3}$$

Conditioned on the current observation $\mathbf{o}_t$, language instruction $\mathbf{l}$, interpolated action $\mathbf{a}^{(s)}_{1:K}$, and flow step s, the DiT velocity predictor f_θ predicts the corresponding velocity field. Since the target velocity along this interpolation path is $\boldsymbol{\epsilon} - \mathbf{a}^*_{1:K}$, we optimize the action pathway with:

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{\mathbf{a}^*_{1:K},\, \boldsymbol{\epsilon},\, s}\left[\left\| f_\theta\left(\mathbf{a}^{(s)}_{1:K}, s, \mathbf{o}_t, \mathbf{l}\right) - (\boldsymbol{\epsilon} - \mathbf{a}^*_{1:K})\right\|_2^2\right]. \tag{4}$$

This objective trains the DiT pathway to generate continuous action chunks.

2D Action Effect Objective. For each horizon $h \in \mathcal{H}$, a lightweight 2D prediction head maps the corresponding Effect representation to the predicted semantic Action Effect $\hat{\mathbf{y}}^{\text{2D}}_{t,h}$. We supervise the prediction using the 2D Action Effect label $\mathbf{y}^{\text{2D}}_{t,h}$ defined in Section 3.1. The 2D objective is formulated as:

$$\mathcal{L}_{\text{2D}} = \frac{1}{|\mathcal{H}|}\sum_{h \in \mathcal{H}} \text{SmoothL1}\left(\hat{\mathbf{y}}^{\text{2D}}_{t,h}, \mathbf{y}^{\text{2D}}_{t,h}\right), \tag{5}$$

where $\text{SmoothL1}(\cdot, \cdot)$ denotes the standard SmoothL1 loss Girshick (2015). This objective supervises the Effect representations to capture the semantic changes associated with each future horizon.

3D Action Effect Objective. For each horizon $h \in \mathcal{H}$, the 3D prediction head takes the corresponding Effect representation together with the canonical anchor coordinates to predict the geometric Action Effect. For each sampled canonical anchor indexed by i, we factorize each anchor 3D Action Effect label $\mathbf{y}^{\mathrm{3D}}_{t,h,i}$ into a binary change target $c_{t,h,i}$ and, for changed anchors, a transition-direction target $d_{t,h,i}$. The former indicates whether the occupancy state changes, while the latter distinguishes free-to-occupied from occupied-to-free transitions. We optimize both predictions using cross-entropy:

$$\mathcal{L}_{\mathrm{3D}} = \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} \left[\frac{1}{|\mathcal{S}_h|} \sum_{i \in \mathcal{S}_h} \mathrm{CE}\left(\hat{c}_{t,h,i}, c_{t,h,i}\right) + \frac{1}{|\mathcal{S}_h^+|} \sum_{i \in \mathcal{S}_h^+} \mathrm{CE}\left(\hat{d}_{t,h,i}, d_{t,h,i}\right) \right], \tag{6}$$

where $\mathrm{CE}(\cdot,\cdot)$ denotes the standard cross entropy loss, $\mathcal{S}_h$ denotes the set of canonical anchors sampled at horizon h, and $\mathcal{S}_h^+ \subseteq \mathcal{S}_h$ denotes the subset with a ground-truth occupancy change. Here, $\hat{c}_{t,h,i}$ and $\hat{d}_{t,h,i}$ are the predicted change state and transition direction. The direction term is evaluated only for anchors with a ground-truth occupancy change, separating the detection of geometric changes from the prediction of their directions. More details are provided in the Appendix.

Overall Training Objective. Combining the action objective with the two structured Action Effect objectives, our overall training loss is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{action}} + \lambda_{\mathrm{2D}}\mathcal{L}_{\mathrm{2D}} + \lambda_{\mathrm{3D}}\mathcal{L}_{\mathrm{3D}}, \tag{7}$$

where λ_{2D} and λ_{3D} are hyperparameters that control the relative contributions of the 2D semantic and 3D geometric Action Effect supervision, respectively.

3.5 Inference

At inference time, we remove the auxiliary 2D and 3D prediction heads together with all future supervision signals. During training, these heads serve only to ground the Effect tokens with structured semantic and geometric supervision, while the Effect tokens themselves are generated and updated through PAA from the current action representation. Once trained, the Effect tokens can therefore anticipate action-associated consequences directly from the evolving action hypothesis without explicitly decoding the 2D or 3D targets. Effect-WAM retains only the DiT backbone, Effect tokens, and PAA interactions to generate action chunks through the same single action-generation pathway.

4 Experiments

4.1 Implementation Details

We use a frozen DINOv3 ViT-B/16 Siméoni et al. (2026) as the visual encoder, T5 Raffel et al. (2020) as the text encoder, and a 30-layer DiT as the action backbone. The action horizon is set to $K = 32$. We insert one PAA module every five DiT layers and use two Effect tokens corresponding to horizons $h \in \{16, 32\}$. The Action Effect supervision weights are set to $\lambda_{\mathrm{2D}} = 1$ and $\lambda_{\mathrm{3D}} = 0.5$. For 3D supervision, we use $m = 3328$ canonical anchors and dynamically sample 256 anchors per horizon during training. We optimize the model with AdamW Loshchilov & Hutter (2019) using a learning rate of 1×10^{-4} and weight decay 0.01. At inference, we use 10 Action Flow solver steps with a scheduler shift of 5.0. The policy predicts a 32-step action chunk and executes the first 10 actions before replanning. Temporal ensembling and Effect warm-start are not used.

4.2 Experimental Setup

We evaluate Effect-WAM on two simulation benchmarks, LIBERO Liu et al. (2023) and RoboTwin 2.0 Chen et al. (2026b), together with a set of real-world manipulation tasks.

LIBERO. LIBERO consists of four suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long, covering 40 language-conditioned manipulation tasks in total. Each task provides 50 expert demonstrations. We train Effect-WAM for $20k$ steps and test each task over 50 episodes, resulting in 2,000 rollouts across the benchmark.

Table 1: Results on LIBERO. Success rates (%) across four task suites.

Method	Spatial	Object	Goal	Long	Avg.
OpenVLA-OFT Kim et al. (2025a)	97.6	98.4	97.9	94.5	97.1
$\pi_{0.5}$ Black et al. (2025a)	98.8	98.2	98.0	92.4	96.9
Motus Bi et al. (2026)	96.8	99.8	96.6	97.6	97.7
LingBot-VA Li et al. (2026)	98.5	99.6	97.2	98.5	98.5
Fast-WAM Yuan et al. (2026)	98.2	100.0	97.0	95.2	97.6
Effect-WAM (Ours)	98.4	99.6	98.0	96.4	98.1

RoboTwin 2.0. RoboTwin 2.0 is a challenging benchmark for bimanual manipulation with strong scene randomization. Following the multi-task setting Yuan et al. (2026), we use a mixture of $2{,}500$ demonstrations collected in clean scenes and $25{,}000$ demonstrations collected under heavy randomization, spanning over 50 tasks. We train Effect-WAM for $30k$ steps and report average success rates over 100 trials per task under both clean and randomized settings.

Real-World Tasks. Our real-world setup consists of a 7-DoF xArm7 equipped with an XHand12 dexterous hand and two camera views, providing a different embodiment from the parallel-gripper platforms used in our simulation experiments. We consider two manipulation tasks: stacking bowls and placing an apple onto a plate, which require accurate grasping, spatial alignment, and controlled placement. We collect 200 teleoperated demonstrations in total, with 100 demonstrations for each task. The Action Effect labels follow the same construction procedure as in simulation.

4.3 Experimental Results

LIBERO. Table 1 compares Effect-WAM with representative VLA and WAM baselines on the four LIBERO suites. Effect-WAM achieves an average success rate of 98.1%, outperforming strong VLA models such as OpenVLA-OFT and $\pi_{0.5}$, as well as representative WAMs including Motus and Fast-WAM. Compared with LingBot-VA, which achieves the best overall average, Effect-WAM remains competitive and shows stronger performance on Goal Suite, improving success rates by 0.8%. These results show that Effect-WAM can achieve strong manipulation performance across diverse task settings without relying on a separate complete future generation branch.

Table 2: Results on RoboTwin 2.0. Success rates (%) under clean and randomized environments.

Method	Clean	Randomized	Avg.
$\pi_{0.5}$ Black et al. (2025a)	82.7	76.8	79.8
Motus Bi et al. (2026)	88.7	87.0	87.8
LingBot-VA Li et al. (2026)	92.9	91.5	92.2
Fast-WAM Yuan et al. (2026)	91.9	91.8	91.8
Effect-WAM (Ours)	92.7	91.8	92.2

RoboTwin 2.0. As shown in Table 2, we further evaluate Effect-WAM on RoboTwin 2.0, a more challenging benchmark featuring diverse bimanual manipulation tasks. Effect-WAM achieves success rates of 92.7% and 91.8% under the clean and randomized settings, respectively, with an average success rate of 92.2%. This performance ranks the best among the compared methods, outperforming Fast-WAM by 0.4% on average. The strong performance under randomized environments further demonstrates the robustness of Effect-WAM across challenging bimanual manipulation settings.

Real-World Tasks. As shown in Figure 4, we test Effect-WAM on two real-world tasks with 50 rollouts per task. Effect-WAM successfully completes $32/50$ rollouts for bowl stacking and $35/50$ rollouts for apple placement, showing strong real-world manipulation performance. Additional results under OOD visual distractors are provided in the Appendix.

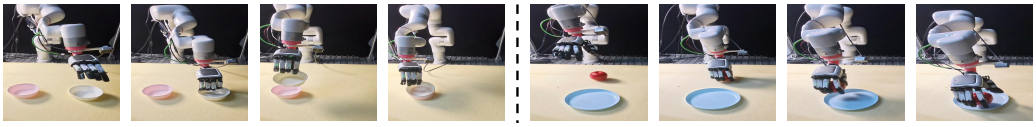

Figure 4: Real-world experiments. Qualitative results of Effect-WAM on two manipulation tasks: bowl stacking and apple placement.

Table 3: Ablation studies on LIBERO. The contribution of each component in EffectWAM.

Method	PAA	2D	3D	E2A	Spatial	Object	Goal	Long	Avg.
Action Only					94.8	98.2	95.8	89.8	94.7
PAA + 2D	✓	✓		✓	96.8	97.6	98.0	90.6	95.8
PAA + 3D	✓		✓	✓	95.8	98.6	96.6	90.2	95.3
PAA w/o Effect Act	✓	✓	✓		96.2	**99.8**	97.2	93.2	96.6
Effect-WAM	✓	✓	✓	✓	**98.4**	99.6	**98.0**	**96.4**	**98.1**

4.4 Ablation Study

Key Components Analysis. Table 3 evaluates the main components of Effect-WAM on LIBERO. Both 2D semantic and 3D geometric supervision improve over the Action only baseline by 1.1% and 0.6%, respectively. The 3D objective further brings a clear gain on the Object Suite of 1.0%, highlighting the advantage of explicitly modeling occupancy and spatial-structure changes beyond 2D semantic variations. Combining 2D and 3D supervision achieves the best performance, while removing the Effect-to-Action act reduces success by 1.5%. These results support both the complementary Action Effect supervision and its integration into action generation through PAA.

Efficiency Analysis. Figure 5 compares the training and inference efficiency of Effect-WAM with Fast-WAM Yuan et al. (2026). Effect-WAM removes the large trainable video world-modeling backbone of Fast-WAM and instead models action effect through a compact PAA pathway with lightweight Effect supervision. As a result, Effect-WAM requires 80.2% fewer trainable parameters, together with lower training time and higher throughput. At inference, the resulting policy uses over 40% fewer parameters and GPU memory than Fast-WAM. The additional PAA interactions introduce a small computational overhead, leading to a modest 5.9% increase in inference latency.

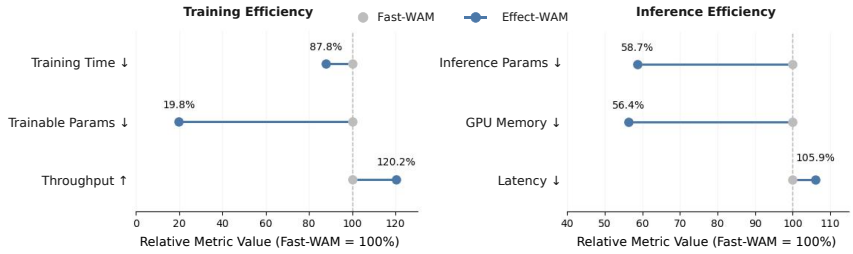

Figure 5: Training and inference efficiency of Effect-WAM compared with Fast-WAM.

5 Conclusion

In this paper, we introduced Action Effect as a predictive target for robot manipulation, focusing future supervision on the changes and consequences associated with actions. Based on this idea, we proposed Effect-WAM, which couples Action Effect modeling with action generation through the Propose-Anticipate-Act mechanism. Experiments on LIBERO, RoboTwin 2.0, and real-world manipulation tasks show that Effect-WAM achieves strong manipulation performance while avoiding complete future generation. In future work, we plan to investigate additional forms of auxiliary supervision that are better suited to WAMs and contain richer action-relevant information, and to develop more lightweight WAM architectures.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the implementation of author-designed methods and translation. We have not used generative AI tools to provide feedback on the research methodology or experiments, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for code editing and debugging, literature search, and language editing. We have reviewed all AI-assisted work. AI-assisted code was tested and verified by the authors, and all AI-assisted text was reviewed for accuracy and consistency with the reported methods and results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our real-world experiments involve robotic manipulation in a controlled laboratory environment. Data collection and evaluation were conducted with appropriate safety precautions. The collected demonstrations do not include personally identifiable or sensitive information.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training and inference configurations, data processing procedures, and evaluation protocols in the main paper and appendix. We also report the key hyperparameters and implementation details required to reproduce our experiments. The code and data-processing pipeline will be released upon publication.

REFERENCES

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A Unified Latent Action World Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *Proceedings of The 9th Conference on Robot Learning*, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, 2025b.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, Yuanbo Xu, and Chao Yu. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *Proceedings of the International Conference on Machine Learning*, 2026b.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, 2024.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, et al. Video prediction policy: A generalist robot policy with predictive visual representations. In *Proceedings of the International Conference on Machine Learning*, 2025.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *Proceedings of the Conference on Robot Learning*, 2025.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Proceedings of Robotics: Science and Systems*, 2025a.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. In *Proceedings of The 8th Conference on Robot Learning*, 2025b.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. In *Proceedings of Robotics: Science and Systems*, 2026.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Proceedings of Robotics: Science and Systems*, 2025.

Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. Jepa-wam: Learning vision-language-action policies with joint-embedding world modeling. *arXiv preprint*, 2026.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, 2023.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: A diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. LIV: Language-Image Representations and Rewards for Robotic Control. In *Proceedings of the 40th International Conference on Machine Learning*, 2023a.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards Universal Visual Reward and Representation via Value-Implicit Pre-Training. In *The Eleventh International Conference on Learning Representations*, 2023b.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A Universal Visual Representation for Robot Manipulation. In *Proceedings of The 6th Conference on Robot Learning*, 2023.

Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models. In *IEEE International Conference on Robotics and Automation*, 2024.

Bikang Pan, Fan Liu, Haotao Lu, Jingya Wang, and Ye Shi. SelfWAM: A Self-Grounded Unified World Action Model for Fast Robot Control. *arXiv preprint arXiv:2608.00725*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025.

Lu Qiu, Yizhuo Li, Yi Chen, Yuying Ge, Yixiao Ge, and Xihui Liu. Making foresight actionable: Repurposing representation alignment in world action models. *arXiv preprint*, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for vision-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3. *Transactions on Machine Learning Research*, 2026.

Hongtao Wu, Ya Jing, Chilam Cheang, et al. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2024.

Guoyang Xia, Fengfa Li, Hongjin Ji, Lei Ren, Fangxiang Feng, Kun Zhan, and Yan Xie. VLAFlow: A Unified Training Framework for Vision-Language-Action Models via Co-training and Future Latent Alignment. *arXiv preprint arXiv:2607.01586*, 2026.

Jonathan Heewon Yang, Chuyuan Fu, Dhruv Shah, Dorsa Sadigh, Fei Xia, and Tingnan Zhang. Bridging perception and action: Spatially-grounded mid-level representations for robot generalization. In *Proceedings of Robotics: Science and Systems*, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Naveed Malik, Kyungmin Lee, William Liang, Nadun Ranawaka Arachchige, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Danfei Xu, Yilun Du, Ryan Julian, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World Action Models are Zero-shot Policies. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026.

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.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Ruiteng Zhao, Zhengshen Zhang, Yue Su, Wenshuo Wang, Jiahui Li, Zhiyuan Yang, Francis E. H. Tay, Marcelo H. Ang, Jr., and Haiyue Zhu. SG-WAM: Self-Guided World Modeling in Geometry-Aware Policy Space. *arXiv preprint arXiv:2608.01397*, 2026a.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-wam: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026b.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Learning 4d embodied world models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann Lecun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Yijing Zhou, Qiwei Liang, Sitong Zhuang, Jiaxi Li, Xianpeng Wang, Boyang Cai, Yunyang Mo, and Renjing Xu. Action-Effect Memory Pretraining for Robot Manipulation. *arXiv preprint arXiv:2606.12499*, 2026.

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. In *Proceedings of Robotics: Science and Systems*, 2025a.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, 2023.

A Appendix

A.1 Additional Training Details

Gradient Routing. We use a single-detach strategy to control gradient propagation between Action Effect modeling and action generation. During the forward pass, the PAA computation remains unchanged: the updated Effect representations participate in E2A cross-attention to refine the action representation. During backpropagation of $\mathcal{L}_{\text{action}}$, however, the Effect representations used as the keys and values of E2A are detached. As a result, the action objective updates the E2A interaction and the DiT action pathway, but does not propagate through the Effect representations into the preceding A2E interaction or Effect queries. This detach affects only backward gradient propagation and does not alter the forward computation.

In contrast, $\mathcal{L}_{\text{2D}}$ and $\mathcal{L}_{\text{3D}}$ backpropagate through the auxiliary prediction heads, Effect representations, and A2E interactions into the intermediate DiT representations. We stop these auxiliary gradients before the current-observation projector, preventing the Action Effect objectives from directly optimizing the current observation representation.

A.2 3D Action Effect Construction and Objective Details

Canonical Anchors. We use a fixed bank of $m = 3328$ canonical anchors distributed within the robot-base workspace. The anchor locations are constructed once and shared across all observations, such that each anchor corresponds to the same physical location throughout a trajectory. For each frame, we compute the occupancy state of the full anchor bank from the fused robot-centric geometry and retain the resulting `FREE`, `OCCUPIED`, and `UNKNOWN` states for subsequent transition construction.

Dynamic Anchor Sampling. Computing the 3D objective over the full anchor bank at every training step is unnecessary. We therefore dynamically sample 256 anchors for each horizon from the precomputed canonical field. The sampled set balances anchors with and without occupancy changes, preventing the substantially more frequent unchanged states from dominating the 3D objective. Unchanged anchors are sampled from both regions near geometric changes and the remaining workspace, providing negatives with different spatial contexts.

Factorized 3D Supervision. The sampled anchors are used to optimize the factorized objective. Change detection is supervised on the complete sampled set, whereas transition direction is supervised only on anchors with a ground-truth occupancy change. This factorization separates whether a geometric change occurs from whether the transition is free-to-occupied or occupied-to-free, avoiding domination by the unchanged class while retaining explicit supervision for the direction of geometric transitions.

A.3 Predictive Target and Action Relevance Diagnostic

We compare Future, Action-conditioned Future, and Action Effect in the same DINO feature space at horizon $h = 32$, using matched model capacity, training budget, and data splits. To ensure a controlled comparison, this diagnostic uses only the 2D DINO representation.

Predictive Target Area. Predictive Target Area measures the fraction of DINO spatial tokens covered by the predictive target. Future and Action-conditioned Future model the complete future feature field, while Action Effect measures the spatial support of significant feature changes between the current and future observations.

Action Relevance. Action Relevance measures how predictive the learned representation is of expert actions. We freeze each predicted representation and train the same downstream action probe using the current representation together with the predicted representation. The resulting action-prediction R^2 is used as the Action Relevance metric, with a Current-only probe serving as the reference.

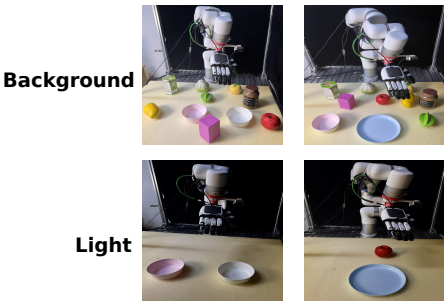

Figure 6: **Real-world OOD evaluation.** Effect-WAM is tested under two visual distribution shifts: **(a)** cluttered tabletop backgrounds with task-irrelevant objects and **(b)** low-light conditions.

A.4 Real-World Experimental Details

In addition to the standard real-world setting, we test Effect-WAM under two out-of-distribution visual conditions. The first introduces cluttered tabletop backgrounds with additional task-irrelevant objects, while the second reduces the ambient lighting to create a darker visual environment. We use the same trained policy without additional fine-tuning and keep the task objectives unchanged across all settings. Under the cluttered-background setting, Effect-WAM successfully completes 28/50 rollouts for bowl stacking and 33/50 rollouts for apple placement. Under the low-light setting, the corresponding results are 32/50 and 34/50 rollouts, respectively. Qualitative examples of the two OOD conditions are shown in Figure 6.