ACTEFFECT: FROM PREDICTIVE FORESIGHT TO CONSEQUENCE FEEDBACK IN ROBOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot actions are interventions on the physical world, and their quality is ultimately determined by the state transitions and task outcomes they induce. However, most robot policies are trained to match demonstrated actions, while predictive world models typically provide foresight for action generation rather than feedback on the outcomes of generated actions. We present **ActEffect**, which augments policy learning with a world model that predicts the consequences of proposed actions and feeds them back to guide policy optimization, internalizing awareness of action outcomes into the policy itself. To make this feedback effective for policy refinement, we develop a decoupled consequence modeling framework that predicts future transitions in a separate DINOv3 latent space, preserving spatial and structural information while reducing interference from task-specific semantics. We then introduce a comparative consequence refinement objective that uses the predicted consequences of coarse-to-fine action proposals from our policy to construct a relative feedback signal. By comparing the refined proposal against the best auxiliary consequence, ActEffect guides policy refinement toward actions whose predicted futures better align with the demonstrated outcome. The world model is used only during training and removed at deployment, introducing no inference overhead. Experiments on simulation benchmarks demonstrate that ActEffect achieves state-of-the-art performance while delivering a 2.5$\times$ inference speedup over the strongest baseline, reaching average success rates of 99.0% on LIBERO and 67.5% on RoboCasa-GR1. On real-world contact-rich manipulation tasks, ActEffect further outperforms the baselines by 13.5 percentage points in success rate while demonstrating strong robustness under perturbations.

1 INTRODUCTION

A robot action is an intervention on the physical world. Its quality is therefore ultimately determined by the state transition and task outcome it induces, rather than only by its numerical similarity to a demonstrated action. This mismatch is particularly relevant in contact-rich robot manipulation, where small differences between action commands can result in substantially different execution outcomes. Although imitation learning has enabled increasingly capable and general robot policies (Chi et al., 2025; Kim et al., 2025b; Bjorck et al., 2025; Li et al., 2026a), minimizing action prediction error alone does not necessarily guarantee successful physical execution. A central challenge is thus to make robot policies aware not only of what action should be generated, but also of how that action will change the future.

Recent robot learning systems have advanced along two complementary directions. Vision-Language-Action models (VLAs) (Kim et al., 2025b; Intelligence et al., 2025; Bjorck et al., 2025; Yuan et al., 2026a) leverage pretrained multimodal representations to connect task understanding with continuous action generation, while World-Action Models (WAMs) (Kim et al., 2026; Ye et al., 2026b; Li et al., 2026a; Yuan et al., 2026c) augment policy learning with predictions of future visual frames or latent representations. Despite substantial progress, these predictive representations still primarily serve as *foresight* for action generation, either by conditioning the policy (Yuan et al., 2026c; Ma et al., 2026) or by coupling future prediction with the action generation process (Ye et al., 2026b; Bi et al., 2026). This leaves a comparatively underexplored question: can a model predict the consequences of the policy's own proposed actions and use them as feedback to improve the policy itself? This question is motivated by the temporal structure of physical interaction,

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.

in which the current state is followed by an action intervention and its resulting outcome. Such a perspective reframes the role of world modeling from *foresight* to *feedback*: instead of supplying predictive context for action generation, the world model evaluates the predicted consequences of actions already proposed by the policy and uses them to guide policy optimization. Unlike model-based planning (Hansen et al., 2024; Zhang et al., 2026a), which evaluates candidate actions through model rollouts at deployment, our goal is to internalize consequence awareness into a direct policy during training, eliminating world-model rollout and candidate search at deployment.

Turning the predicted consequences of a policy's action proposals into effective training-time feedback raises two closely related challenges. First, the consequence space should emphasize the structural state changes induced by robot actions, while suppressing action-irrelevant variations and remaining decoupled from semantic information in other modalities, such as instructions. Second, within such a consequence space, the predicted consequences of different action proposals should provide a reliable basis for evaluating their relative quality, allowing consequence differences to be converted into a learning signal for improving the policy itself.

In this paper, we present ActEffect, a robot policy that internalizes awareness of action outcomes by augmenting policy learning with consequence feedback from a world model. We represent action consequences in a separate frozen DINOv3 latent space (Siméoni et al., 2025), whose features preserve spatial and structural information while suppressing appearance variations and irrelevant background changes. For action generation, we adopt an iterative action regression policy, instantiated with Minimal Iterative Policy (MIP) (Pan et al., 2026), which explicitly produces complete action proposals during training for direct evaluation by the world model, rather than relying on interactions between the world model and intermediate policy representations as in flow-based policies (Won et al., 2026; Ye et al., 2026a). Together, these designs further isolate consequence modeling from vision-language hidden-state interactions while allowing us to directly measure how closely each action-induced future matches the demonstrated future. Building on this, we introduce a comparative consequence refinement objective that leverages the predicted consequences of policy-generated action proposals, using the best auxiliary consequence as a reference to guide the final refined action toward the demonstrated outcome and provide consequence-aware feedback for policy optimization. The world model is used only during training and removed at deployment, introducing no additional inference overhead. We evaluate ActEffect on LIBERO (Liu et al., 2023) and RoboCasa-GR1 (Nasiriany et al., 2024), achieving state-of-the-art performance with a $2.5\times$ inference speedup and average success rates of 99.0% and 67.5%, respectively. On real-world contact-rich USB insertion tasks, ActEffect further improves success by 13.5 percentage points over the strongest baseline and remains robust under perturbations.

Specifically, our contributions can be summarized as follows:

1. We introduce ActEffect, a robot policy that shifts world modeling from foresight for action generation to training-time consequence feedback for refining its own action proposals, thereby internalizing awareness of action outcomes.
2. To enable effective feedback, we develop a decoupled consequence modeling framework that leverages a spatially structured DINOv3 latent space and explicit action proposals to isolate consequence modeling from task semantics. We further introduce a comparative consequence refinement objective that uses the best auxiliary consequence as a reference to guide the final action toward the demonstrated outcome, converting relative consequence differences into targeted feedback for policy optimization.
3. Experiments across simulation and real-world contact-rich manipulation tasks demonstrate state-of-the-art performance, inference efficiency, robustness under perturbations, and the emergence of corrective behaviors induced by training-time consequence feedback.

2 Related Work

2.1 Vision Language Action Models

Vision-Language-Action models (VLAs) extend pretrained vision-language models to robotic control, leveraging their rich semantic representations to map visual observations and language instructions to executable robot actions (Kim et al., 2025b; Black et al., 2024; Intelligence et al., 2025;

Bjorck et al., 2025; Zheng et al., 2026; Yuan et al., 2026a; Yang et al., 2026). With large-scale robot data and pretrained multimodal backbones, recent VLAs have demonstrated strong instruction following, generalization, and multi-task manipulation capabilities (Intelligence et al., 2025; Yuan et al., 2026a). However, most VLAs remain primarily trained through action imitation, directly predicting demonstrated actions without modeling the resulting evolution of the physical environment. Such action-only supervision provides limited signals for learning interaction dynamics, particularly as manipulation tasks become more complex.

2.2 World Action Models

World-Action Models (WAMs) address this limitation by introducing predictive world modeling into robot policies, jointly or sequentially modeling future scene evolution and executable robot actions (Zhu et al., 2025; Bi et al., 2026; Kim et al., 2026; Ye et al., 2026b; Li et al., 2026a). To avoid costly future-frame generation at inference, recent methods increasingly exploit predictive representations directly: DiT4DiT (Ma et al., 2026) conditions actions on video-denoising features, Image-WAM (Zhang et al., 2026b) uses representations from image editing model, and Fast-WAM (Yuan et al., 2026c) removes explicit future prediction at test time. Despite these variants, future prediction still serves as *foresight*: predicted futures enrich the context or representations from which actions are generated, so inaccurate predictions may propagate into the resulting actions.

A complementary line reverses the conditioning direction by predicting futures from action representations rather than generating actions from predicted futures: FLARE (Zheng et al., 2025) aligns intermediate diffusion policy features with future observation representations, while DUST (Won et al., 2026) and GigaWorld-Policy (Ye et al., 2026a) couple action and visual modeling through shared Transformer attention. These methods share a similar motivation of leveraging future modeling during policy learning while avoiding explicit imagine-then-act inference. However, compared with ActEffect, these methods have two limitations. First, future prediction shares the representation pathway used for task reasoning and action generation, potentially entangling task semantics with action-induced consequence modeling. Second, predicted consequences are leveraged implicitly through representation alignment or denoising objectives, rather than being explicitly evaluated across action proposals to provide feedback for policy refinement. In contrast, ActEffect models consequences in a separate, task-decoupled latent space and explicitly converts the predicted outcomes of alternative proposals into feedback for policy optimization.

3 Preliminaries

3.1 Problem Setup

Let $o_t = (o_t^v, o_t^p)$ denote the observation at timestep t, comprising the multi-view visual observation o_t^v and proprioceptive state o_t^p, and let l denote the language instruction. A standard VLA policy directly predicts an action chunk $a_{t:t+H-1}$ conditioned on these inputs as $\pi(a_{t:t+H-1} \mid o_t, l)$, where H denotes the action horizon (Intelligence et al., 2025; Bjorck et al., 2025). WAMs extend this formulation by jointly modeling the action chunk and its corresponding future visual observations, $p(o^v_{t+1:t+H}, a_{t:t+H-1} \mid o_t, l)$ (Lyu et al., 2026; Bi et al., 2026; Ye et al., 2026b). This joint distribution admits two complementary factorizations that characterize different roles of future prediction in policy learning:

$$\begin{aligned} p(o^v_{t+1:t+H}, a_{t:t+H-1} \mid o_t, l) &= p(o^v_{t+1:t+H} \mid o_t, l)\pi(a_{t:t+H-1} \mid o_t, o^v_{t+1:t+H}, l), \quad \text{(Foresight)} \\ &= \pi(a_{t:t+H-1} \mid o_t, l)p(o^v_{t+1:t+H} \mid o_t, a_{t:t+H-1}, l). \quad \text{(Feedback)} \end{aligned} \tag{1}$$

The first predicts future scene evolution before action generation and uses it as additional policy context, corresponding to the *foresight* role adopted by many existing approaches (Yuan et al., 2026c; Ma et al., 2026). This work focuses on the largely unexplored second factorization, which retains dense future-modeling supervision while reducing inference-time error accumulation and latency, and better follows the temporal structure, as illustrated in Sec. 1. For notational simplicity, we use a_t to denote the complete action chunk $a_{t:t+H-1}$ hereafter.

3.2 Minimal Iterative Policy

Minimal Iterative Policy (MIP) (Pan et al., 2026) is a regression policy that retains noise-perturbed training and iterative action generation from diffusion and flow-matching policies, while reducing denoising to two supervised prediction stages. This lightweight formulation retains competitive control performance while exposing explicit intermediate action proposals during training. Specifically, given an expert action $a^\star$ and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, MIP constructs $a^\tau = \tau a^\star + (1-\tau)\epsilon$, and supervises the policy as

$$\mathcal{L}_{\text{MIP}} = \|\pi_\theta(c, 0, 0) - a^\star\|_2^2 + \|\pi_\theta(c, a^\tau, \tau) - a^\star\|_2^2, \tag{2}$$

where c denotes the policy condition. At inference, stochastic perturbation is removed and the two stages are executed sequentially:

$$a^0 = \pi_\theta(c, 0, 0), \qquad a^1 = \pi_\theta(c, \tau a^0, \tau). \tag{3}$$

This property is particularly well suited to our setting: since both a^0 and a^1 are complete action proposals rather than intermediate noisy states, their predicted consequences can be independently evaluated and compared by the world model. In contrast, flow-based policies can only provide intermediate noisy states at sampled timesteps during training, whose deviation from complete executable actions can introduce additional error into consequence evaluation.

4 Method

In this section, we present ActEffect, which uses predicted action consequences as training-time feedback for policy learning. We first introduce the overall policy architecture, then describe how effective feedback is enabled through decoupled consequence modeling and comparative consequence refinement, and finally present consequence-guided policy learning.

4.1 Model Overview

As illustrated in Fig. 1, ActEffect comprises a vision-language policy that generates complete action proposals and a world model that evaluates their consequences in a separate DINOv3 latent space to provide training-time feedback (Siméoni et al., 2025).

Specifically, we append learnable action queries $\mathcal{Q}$ to the language instruction and process them together with the current visual observation using the vision-language model, yielding task representations q_t from (o_t^v, l) and action-query states q_t^{act} from $\mathcal{Q}$. In parallel, the visual observation and proprioceptive state are encoded as $z_t^v = \text{sg}(E_{\text{DINO}}(o_t^v))$ and $z_t^p = E_{\text{State}}(o_t^p)$, respectively. The resulting policy condition is $c_t = (q_t, z_t^v, z_t^p)$, where the DINO features provide complementary spatial and structural cues for action generation. The action-query states are mapped directly to the continuous action space to produce a feedforward proposal, $a_t^{\text{ff}} = \text{MLP}(q_t^{\text{act}})$. This direct pathway encourages the vision-language representation to retain action-relevant information while providing an additional action hypothesis for consequence evaluation (Kim et al., 2025a; Intelligence et al., 2025; Zhong et al., 2026). In parallel, the DiT action head follows the two-stage MIP refinement process in Sec. 3.2, producing a coarse proposal a_t^0 and a refined proposal a_t^1. MIP exposes complete coarse-to-fine action predictions during training while retaining a two-stage refinement process at inference, making it well suited for explicit consequence evaluation (Pan et al., 2026). ActEffect therefore exposes three complete action proposals, $\mathcal{A}_t = \{a_t^{\text{ff}}, a_t^0, a_t^1\}$, corresponding to the feedforward, coarse, and refined stages. These proposals share the same current context but represent distinct action hypotheses whose predicted consequences are subsequently compared to guide the policy refinement.

4.2 Decoupled Consequence Modeling

To enable effective consequence feedback, we first establish a task-decoupled world model that emphasizes action-induced state changes through two design choices.

First, following recent latent world models, we model future evolution in a compact latent space rather than reconstructing future observations in pixel space. Specifically, instead of modeling the pixel-space transition $p(o_{t+1:t+H}^v \mid o_t, a_t, l)$ in equation 1, we consider its latent counterpart

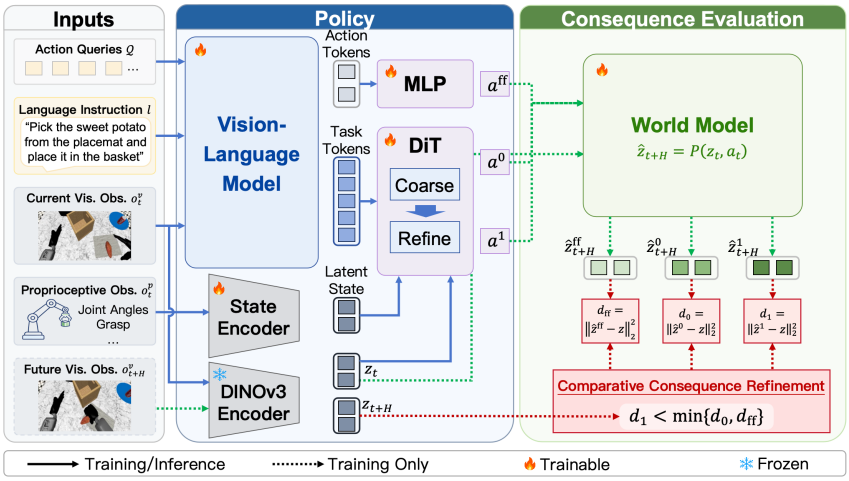


Figure 1: **Model Architecture of ActEffect.** The policy generates feedforward, coarse, and refined action proposals, while a world model predicts their consequences in a frozen DINOv3 latent space and provides comparative feedback during training; the world model is removed at inference.

$p(z^v_{t+1:t+H} \mid z_t, a_t, l)$. Latent consequence modeling suppresses appearance-level variations and irrelevant background details while avoiding expensive future-frame decoding, allowing the world model to focus on action-induced physical state changes. In our implementation, we instantiate this formulation using the terminal latent consequence z^v_{t+H} to represent the outcome of the complete action chunk.

Second, once the current state is fixed, the resulting physical transition should be governed by the action intervention rather than task-semantic information. Therefore, the transition model should be independent of the language instruction:

$$p(z^v_{t+H} \mid z_t, a_t, l) \equiv p(z^v_{t+H} \mid z_t, a_t). \tag{4}$$

This is important because language conditioning can bias the model toward task-correlated patterns rather than differences induced by the action itself, weakening action sensitivity. For the same reason, the choice of latent space z should also be decoupled from task semantics. Prior works (Zheng et al., 2025; Lyu et al., 2026; Ye et al., 2026a; Won et al., 2026) typically use language-conditioned or VLM task representations directly as the latent space for future modeling, which can blur the effect of action interventions.

Following these considerations, we instantiate the consequence space with a frozen DINOv3 encoder. Unlike VLM representations, DINOv3 provides dense spatial and structural features, making it well suited for representing action-induced physical state changes while isolating consequence modeling from task semantics. We therefore encode the current and demonstrated terminal observations as $z^v_t = \mathrm{sg}\left(E_{\mathrm{DINO}}(o^v_t)\right)$ and $z^v_{t+H} = \mathrm{sg}\left(E_{\mathrm{DINO}}(o^v_{t+H})\right)$. Given the current state representation and an action proposal, the world model P_ϕ predicts the corresponding terminal consequence in the DINOv3 latent space. For each proposal $a^i_t \in \mathcal{A}_t$, $i \in \{\mathrm{ff}, 0, 1\}$, we define its consequence discrepancy with respect to the demonstrated outcome as

$$d_i = d(a^i_t) = \frac{1}{KD} \left\| P_\phi(z_t, a^i_t) - z^v_{t+H} \right\|_2^2, \tag{5}$$

where K and D denote the number and dimensionality of DINO tokens, respectively. A smaller $d(a^i_t)$ indicates a predicted consequence more consistent with the demonstrated future.

The world model is supervised using the observed transition induced by the demonstrated action, i.e., $\mathcal{L}_{\mathrm{wm}} = d(a^\star_t)$. This objective grounds consequence prediction in observed action–outcome pairs, providing the transition supervision used to construct feedback for policy refinement.

4.3 Comparative Consequence Refinement

To translate predicted action consequences into effective training-time feedback, we introduce comparative consequence refinement (CCR) as our consequence feedback objective, which leverages the relative consequences of multiple action proposals to guide the refinement of the final policy action.

Given ActEffect's three action proposals $\mathcal{A}_t = \{a_t^{\text{ff}}, a_t^0, a_t^1\}$ and their corresponding consequence discrepancies d_{ff}, d_0, and d_1, we construct the reference discrepancy by selecting the smaller discrepancy between the feedforward and coarse proposals: $d_{\text{ref}} = \min\{\text{sg}(d_0), \text{sg}(d_{\text{ff}})\}$, where $\text{sg}(\cdot)$ denotes the stop-gradient operation. The resulting reference corresponds to the auxiliary proposal whose predicted consequence is closer to the demonstrated outcome. We then define the comparative consequence refinement objective as

$$\mathcal{L}_{\text{CCR}} = T \ \text{softplus}\left(\frac{d_1 - d_{\text{ref}} + m}{T}\right), \tag{6}$$

where $T > 0$ controls the smoothness of the objective and $m \geq 0$ specifies the desired improvement margin. The objective encourages the refined proposal to induce a predicted consequence closer to the demonstrated outcome than the auxiliary reference. When the refined proposal exhibits a larger consequence discrepancy than the reference, it receives a stronger consequence-guided learning signal. As its predicted consequence improves relative to the auxiliary proposals, the feedback strength decreases. Importantly, gradients are stopped through the reference discrepancy, preventing the objective from being minimized by directly degrading the reference proposals. Instead, consequence feedback acts through the refined proposal, aligning the optimization target with the action executed at deployment. In this way, CCR converts relative consequence differences among action proposals into a unified training-time feedback signal, guiding the policy toward refined actions whose predicted outcomes better match the demonstrated consequence.

4.4 Consequence-Guided Policy Learning

We optimize ActEffect by combining direct action supervision, world-model learning, and comparative consequence refinement. The feedforward proposal is supervised with an ℓ_1 loss, while the coarse and refined proposals follow the MIP objective in Sec. 3.2, giving

$$\mathcal{L}_{\text{policy}} = \mathcal{L}_{\text{MIP}} + \lambda_{\text{ff}} \|a_t^{\text{ff}} - a_t^\star\|_1. \tag{7}$$

Together with the world model training objective $\mathcal{L}_{\text{wm}}$ in Sec. 4.2 and the comparative consequence refinement objective $\mathcal{L}_{\text{CCR}}$ in Sec. 4.3, the complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{policy}} + \lambda_{\text{wm}}\mathcal{L}_{\text{wm}} + \lambda_{\text{CCR}}\mathcal{L}_{\text{CCR}}. \tag{8}$$

The three objectives provide complementary supervision: $\mathcal{L}_{\text{policy}}$ learns the demonstrated action, $\mathcal{L}_{\text{wm}}$ grounds consequence prediction in observed state transitions, and $\mathcal{L}_{\text{CCR}}$ converts relative consequence differences across policy proposals into feedback for refining the final action. Gradients from $\mathcal{L}_{\text{CCR}}$ propagate through the refined consequence branch to both the policy and world model, allowing consequence prediction to co-evolve with the policy's progressively refined action proposals, while the auxiliary reference is stop-gradient. We analyze the resulting optimization dynamics in Appendix A and B.2.

At deployment, the world model is removed, and the policy directly generates the executed action a_t^1 through the two-stage MIP refinement, requiring no consequence prediction or evaluation at inference.

5 Experiments

We evaluate ActEffect to answer three primary questions: **(Q1)** How does ActEffect compare with state-of-the-art imitation-learning-based VLA and WAM policies in both task success rate and inference efficiency? **(Q2)** How much does each component of ActEffect contribute to its overall performance? **(Q3)** Does the learned world model provide meaningful consequence signals for policy refinement? We investigate these questions across a diverse range of manipulation tasks, addressing Q1 in Sec. 5.2, Q2 in Sec. 5.3, and Q3 in Sec. 5.4.

Table 1: **Success rate (%) on LIBERO.** Each suite contains 10 tasks evaluated over 50 trials per task; Avg. is the macro-average across suites.

Method	Long ↑	Goal ↑	Object ↑	Spatial ↑	Avg. ↑
VLA					
π_0 (Black et al., 2024)	88.4	94.4	96.8	98.0	94.4
$\pi_{0.5}$ (Intelligence et al., 2025)	92.4	98.0	98.2	98.8	96.9
GR00T-N1.6 (Bjorck et al., 2025)	94.4	97.5	98.5	97.7	97.0
VLA-JEPA (Sun et al., 2026)	95.8	97.2	99.6	96.2	97.2
WAM					
FLARE (Zheng et al., 2025)	92.2	95.6	**100.0**	96.8	96.2
DUST (Won et al., 2026)	92.6	96.0	99.8	96.2	96.2
Motus (Bi et al., 2026)	97.6	96.6	99.8	96.8	97.7
Fast-WAM (Yuan et al., 2026c)	95.2	97.0	**100.0**	98.2	97.6
Cosmos-Policy (Kim et al., 2026)	97.6	98.2	**100.0**	98.1	98.5
LingBot-VA (Li et al., 2026a)	**98.5**	97.2	99.6	98.5	98.5
DIT4DIT (Ma et al., 2026)	97.6	**98.6**	99.6	98.4	98.6
LaWAM (Chen et al., 2026)	97.0	98.4	99.6	99.4	98.6
OURS					
ActEffect	97.6	98.4	**100.0**	**99.8**	**99.0**

Table 2: **Average success rate (%) on RoboCasa-GR1.** Results cover all 24 tabletop tasks.

Method	Avg. ↑
VLA	
$\pi_{0.5}$ (Intelligence et al., 2025)	37.0
GR00T N1.6 (Bjorck et al., 2025)	47.6
FoMoVLA (Li et al., 2026b)	56.9
ABot-M0 (Yang et al., 2026)	58.3
WAM	
UWM (Zhu et al., 2025)	19.3
FLARE (Zheng et al., 2025)	36.3
DUST (Won et al., 2026)	42.0
DIT4DIT (Ma et al., 2026)	50.8
Cosmos-policy (Kim et al., 2026)	52.3
Motus (Bi et al., 2026)	53.1
LDA-1B (Lyu et al., 2026)	55.4
Fast-WAM (Yuan et al., 2026c)	56.7
OURS	
ActEffect	**67.5**

5.1 EXPERIMENTAL SETUP

We comprehensively evaluate ActEffect on three task suites, including two widely used simulation benchmarks and a real-world robot manipulation task. For each task suite, we compare against a representative set of top-performing baselines. Detailed training and evaluation protocols, together with additional baseline descriptions, are provided in the appendix.

LIBERO The LIBERO benchmark (Liu et al., 2023) focuses on tabletop manipulation tasks with a Franka Emika Panda robot arm. We follow the standard protocol and train on four standard suites: Spatial, Object, Goal, and Long. Each suite contains 10 tasks with 50 demonstrations per task, resulting in a total of 2,000 training demonstrations. For evaluation, each task is evaluated over 50 rollouts, and we report the average success rate.

RoboCasa-GR1 The RoboCasa-GR1 benchmark (Nasiriany et al., 2024) evaluates generalist manipulation on a substantially more complex embodiment. Built upon the RoboCasa simulation framework, the benchmark features the Fourier GR-1 humanoid equipped with two 7-DoF arms, two 6-DoF dexterous hands, and a 3-DoF waist, resulting in a 29-dim action space. The benchmark consists of 24 household tabletop manipulation tasks spanning diverse pick-and-place and articulated-object interactions, with 1,000 human-collected demonstrations provided for each task. We evaluate all 24 tasks with 50 rollouts per task and report the average success rate across tasks.

Real-world manipulation tasks. We use a custom wheeled dual-arm humanoid robot platform based on Gento Skye for real-world experiments. Our evaluation suite focuses on a contact-rich USB manipulation task, in which the robot is instructed to pull a USB plug from a receptacle on the left and insert it into a receptacle on the right. The robot and task setup are shown in Figure 3. We consider four evaluation conditions spanning both in-distribution and out-of-distribution settings: (1) **fixed** initial states, (2) **randomized** initial states, (3) **background** variations, and (4) **lighting** variations, with the specific scene configurations illustrated in Figure 4. All policies receive three RGB views (head, waist, and right-wrist) and a 15-dimensional proprioceptive state comprising right-arm joint positions, joint efforts and gripper state. Each policy predicts 50-step action chunks at 50 Hz; each action controls the seven joints of the right arm and the right gripper, while the left arm, waist, and head remain fixed. The chunks are executed asynchronously in a receding-horizon manner. Each policy is trained on 600 demonstrations totaling approximately 2.5 hours of robot data and evaluated over 50 rollouts; we report the average success rate.

Baselines. In LIBERO and RoboCasa-GR1, we compare against recent top-performing methods reported on each benchmark, covering both **VLA** and **WAM** policies. In the real-world evaluation, we compare against a competitive subset spanning the same model families: $\pi_{0.5}$ (Intelligence et al., 2025) as a representative VLA policy, Fast-WAM (Yuan et al., 2026c) as a foresight-based WAM, and GigaWorld-Policy as a feedback-based WAM (Ye et al., 2026a).

Table 3: **Real-world evaluation results.** Success rates (%) under four evaluation conditions.

Method	Fixed ↑	Random ↑	Bg. ↑	Light ↑	Avg. ↑
$\pi_{0.5}$	<u>94.0</u>	74.0	<u>76.0</u>	68.0	<u>78.0</u>
Fast-WAM	86.0	68.0	64.0	58.0	69.0
GigaWorld-Policy	88.0	<u>78.0</u>	70.0	<u>74.0</u>	77.5
ActEffect	**100.0**	**96.0**	**84.0**	**86.0**	**91.5**

Table 4: **Inference efficiency.** Inference latency (ms) on our three task suites.

Method	LIBERO ↓	RoboCasa-GR1 ↓	Real-world ↓
$\pi_{0.5}$	<u>251.12</u>	<u>249.13</u>	<u>298.42</u>
Fast-WAM	265.52	256.59	302.86
GigaWorld-Policy	387.41	374.12	437.26
ActEffect	**99.08**	**91.79**	**114.37**

5.2 Main Experimental Results

Simulation benchmarks. As shown in Tables 1 and 2, ActEffect achieves the best average success rate on both benchmarks, reaching 99.0% on the already near-saturated LIBERO and 67.5% on RoboCasa-GR1, where it exceeds the foresight-based Fast-WAM and DiT4DiT by 10.8% and 16.7%, respectively. This suggests that predicting the consequences of the policy's own action proposals and feeding them back into training is a more effective use of a world model than supplying predictive context for action generation, and that such consequence feedback remains beneficial under a higher-dimensional action space and more complex manipulation dynamics. ActEffect also surpasses feedback-based WAMs such as FLARE and DUST on both benchmarks, indicating that explicitly evaluating the consequences of complete action proposals and using their relative differences to guide action refinement provides an effective learning signal beyond implicit coupling between future representations and action generation.

Real-world results. As shown in Table 3, ActEffect achieves the highest success rate under all four conditions, averaging 91.5% and outperforming the strongest baseline $\pi_{0.5}$, the foresight-based Fast-WAM, and the feedback-based GigaWorld-Policy by 13.5%, 22.5%, and 14.0%. Small differences in the generated actions can easily change the outcome of this contact-rich task, and the gap widens further under perturbations, growing largest under randomized initial states and lighting variations. This indicates that consequence-guided policy refinement is particularly effective in contact-rich tasks where small action deviations can lead to substantially different execution outcomes, while the decoupled DINOv3 consequence space reduces sensitivity to action-irrelevant appearance changes.

Inference efficiency. As shown in Table 4, ActEffect achieves the lowest inference latency across all three task suites, with a 2.5×–2.7× speedup over the fastest baseline. This efficiency follows from its training-time use of consequence modeling: the world model is removed at deployment, and the policy generates each action chunk with only two MIP iterations, without future prediction.

5.3 Ablation Study

We conduct ablation studies on LIBERO and RoboCasa-GR1 to examine the contribution of each major component in ActEffect, with results summarized in Table 5. **(1) Consequence feedback.** Removing comparative consequence refinement reduces success from 99.0% to 97.0% on LIBERO and from 67.5% to 48.9% on RoboCasa-GR1, demonstrating the importance of consequence-based supervision beyond direct action imitation. **(2) Comparative refinement.** Replacing L_{CCR} with an absolute consequence loss, $\mathcal{L}_{\mathrm{abs}} = d_1$, reduces performance to 97.9% and 59.0%, respectively, demonstrating the benefit of using relative consequence differences to guide policy refinement. **(3) Consequence space.** Replacing the decoupled DINO consequence space with VLM representations reduces performance to 97.3% and 54.0%, supporting the separation of task-semantic representations from consequence modeling. **(4) Feedforward proposal.** Removing a^{ff} decreases performance to 98.2% and 63.4%, indicating that the additional task-conditioned proposal benefits overall policy learning. Overall, these results demonstrate the importance of ActEffect's key designs in consequence feedback, comparative refinement, and decoupled consequence modeling.

Table 5: **Ablation Study Results.** Average success rates (%) of ActEffect variants.

Variant	LIBERO ↑	RoboCasa-GR1 ↑
w/o Feedback	97.0	48.9
w/ Absolute loss	97.9	59.0
w/o Decoupling	97.3	54.0
w/o FF Proposal	<u>98.2</u>	<u>63.4</u>
Full	**99.0**	**67.5**

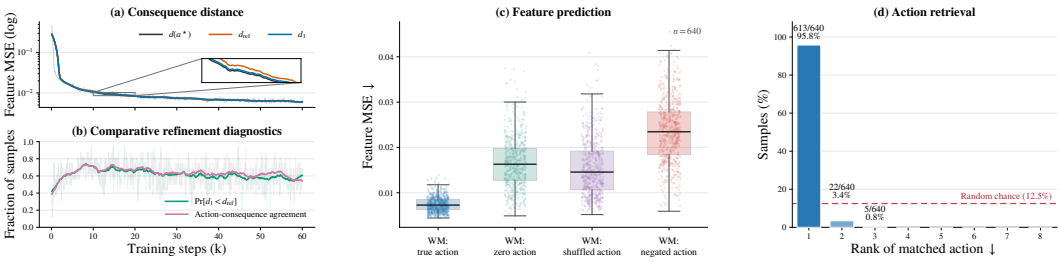


Figure 2: **Analysis of Consequence Feedback.** (a) Consequence distances during training. (b) Comparative refinement diagnostics. (c) Future prediction error under action interventions. (d) Matched-action retrieval among 8 candidates.

5.4 Analysis of the Consequence Feedback

We answer Q3 on LIBERO by examining how comparative consequence refinement evolves during training and whether the learned world model captures action-sensitive transition signals. Full evaluation details are provided in Appendix C.3.

As shown in Fig. 2 (a), the demonstrated-action distance $d(a^\star)$ decreases from 0.039 to 0.006, while the proposal-conditioned distances decrease alongside it, indicating that the world model learns the demonstrated transition dynamics as the policy improves. The refined proposal also remains closer to the demonstrated consequence than its auxiliary reference, with the distances exhibiting $d(a^\star) < d_1 < d_{\text{ref}}$ over most of training. Figure 2(b) further shows that $\Pr[d_1 < d_{\text{ref}}]$ increases rapidly during training, indicating that the refined proposal increasingly improves upon the best auxiliary consequence. After the initial rise, the fraction stabilizes and slightly decreases toward the end of training as all proposals continue to improve under direct action supervision and their consequence discrepancies gradually converge. Meanwhile, consequence-space and action-space comparisons agree on 63.8% of samples, showing that the two signals are correlated but not equivalent. This is consistent with Proposition 1: $d(\cdot)$ measures action deviations through their induced state transitions rather than action-space distance alone. **These results show that the world model becomes increasingly accurate, while its relative consequence signal supports progressive policy refinement.**

We further evaluate the trained world model under action substitutions while keeping the observation fixed. As shown in Fig. 2 (c), the action that generated the observed transition produces less than half the future-feature error of all tested alternatives, including persistence prediction, a zero action, a shuffled action from another sample, and a negated action. We further reverse the evaluation in Fig. 2 (d): given an observed transition and 8 candidate actions, ranking candidates by predicted consequence places the matched action first in 95.8% of samples, compared with 12.5% random chance. **These results show that the learned world model captures action-sensitive consequence signals that distinguish alternative actions and provide informative feedback for policy refinement.**

6 Conclusion

We introduced ActEffect, a robot policy that shifts the role of world modeling from predictive foresight to training-time consequence feedback. A task-decoupled DINOv3 consequence space isolates action-induced state changes from task semantics, while comparative consequence refinement converts relative consequence differences across policy proposals into feedback for refining the final action. Since the world model is removed at deployment, ActEffect combines state-of-the-art control performance with efficient inference, reaching 99.0% on LIBERO, 67.5% on RoboCasa-GR1, and 91.5% in real-world manipulation tasks. Our analyses further show that the learned world model captures action-sensitive dynamics and provides proposal-specific consequence signals for policy refinement. Future work will scale ActEffect through large-scale embodied pretraining and evaluate it on more complex real-world manipulation tasks.

AI USE STATEMENT

During the preparation of this manuscript, we used AI-assisted tools for assistance in writing and programming. Specifically, we used AI tools to improve language clarity, grammar, and manuscript presentation, and to assist with selected aspects of experimental code implementation and debugging. All research ideas, algorithmic designs, theoretical analyses, experimental protocols, results interpretation, and scientific conclusions were developed and verified by us. We take full responsibility for the final content of this manuscript.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the ActEffect model architecture, training details, optimization procedures, and experimental settings in Appendix B and C. The main implementation code is provided in the supplementary materials. Upon acceptance, we will release the source code, configuration files, and trained model checkpoints.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Jialei Chen, Kai Wang, Kang Chen, Shuaihang Chen, Feng Gao, Wenhao Tang, Zhiyuan Li, Weilin Liu, Zhuyu Yao, Boxun Li, et al. Lawam: Latent world action models for efficient dynamics-aware robot policies. *arXiv preprint arXiv:2606.15768*, 2026.

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

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

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.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 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 Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025b.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

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

Wei Li, Peijin Jia, Yuan Ma, Xuefeng Jiang, Titong Jiang, Sheng Sun, Yujian Li, Xin Wen, Han Hong, Zhikang Liu, et al. Fomovla: Bridging visual foresight and motion guidance for vision-language-action models. *arXiv preprint arXiv:2607.14739*, 2026b.

Xiang Li, Yupeng Zheng, Songen Gu, Huailiang Ma, Feng Yu, Xian Nie, Shanshuai Yuan, Yujie Zang, Weize Li, Shuai Tian, et al. Latent action as intention enables efficient future imagination for world action models. *arXiv preprint arXiv:2608.24882*, 2026c.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. Dit4dit: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

Chaoyi Pan, Giri Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Matthew Boffi, Guanya Shi, and Max Simchowitz. Much ado about noising: Dispelling the myths of generative robotic control. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LzWKuxTKuW.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. Vla-jepa: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-stream diffusion for world-model augmented vision-language-action model. In *Forty-third International Conference on Machine Learning*, 2026.

Haodong Yan, Junfeng Li, Junjie He, Zhide Zhong, MingMing Yu, Wenxuan Song, Jiaguan Zhu, Yangyang Zheng, Yuqiao Du, Jiadi You, et al. Robust-wam: Bridging generative pretraining and semantic foresight in world-action models. *arXiv preprint arXiv:2608.05903*, 2026.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

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

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

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026a.

Shanglin Yuan, Weiheng Zhao, Xin Shi, Haoyi Jiang, Xianda Guo, Liu Liu, Wenyu Liu, Wei Sui, and Xinggang Wang. Dreamwam: Beyond rgb future prediction for world action models. *arXiv preprint arXiv:2608.04996*, 2026b.

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*, 2026c.

Tianyi Zhang, Likun Wang, Guojian Zhan, Feihong Zhang, Yang Guan, Yao Lyu, and Shengbo Eben Li. Langevin rollout optimization for modelic reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026a.

Yuyang Zhang, Wenyao Zhang, Zekun Qi, He Zhang, Haitao Lin, Jingbo Zhang, Yao Mu, Xiaokang Yang, Wenjun Zeng, and Xin Jin. Imagewam: Do world action models really need video generation, or just image editing? *arXiv preprint arXiv:2606.19531*, 2026b.

Ruiteng Zhao, Zhengshen Zhang, Yue Su, Wenshuo Wang, Jiahui Li, Zhiyuan Yang, Francis EH 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.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loïc Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot learning with implicit world modeling. In *9th Annual Conference on Robot Learning*, 2025.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Si Liu, and Guanghui Ren. Acot-vla: Action chain-of-thought for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8152–8162, 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. *arXiv preprint arXiv:2504.02792*, 2025.

A Theoretical Analysis

A.1 Consequence Feedback Induces a Dynamics-Aware Action Metric

Action imitation treats every coordinate of the action space alike: an ℓ_2 regression loss penalizes a deviation of a given magnitude equally regardless of the direction in which it points. The premise of this work is that the physical world does not. We now make that statement precise by showing that, to second order around the demonstrated action, the consequence discrepancy of equation 5 is a quadratic form whose metric is set by the transition itself.

Write $N = KD$ for the number of predicted feature entries, let $a^\star$ denote the demonstrated action chunk, and let

$$r \;=\; P_\phi(z_t, a^\star) - z^v_{t+H} \tag{9}$$

be the residual that the world model leaves on the demonstrated transition, which is exactly the quantity minimized by $\mathcal{L}_{\mathrm{wm}}$.

Proposition 1 (Dynamics-aware consequence metric) *Suppose $P_\phi(z_t, \cdot)$ is differentiable at the demonstrated action $a^\star$ with Jacobian $J = \partial P_\phi(z_t, a)/\partial a\big|_{a=a^\star} \in \mathbb{R}^{N\times d_a}$. Then for $a = a^\star + \Delta a$, the local Gauss–Newton approximation of the consequence discrepancy is*

$$d(a) \approx \frac{1}{N}\Big[\Delta a^\top J^\top J\,\Delta a + 2\,r^\top J\,\Delta a + \|r\|_2^2\Big]. \tag{10}$$

When the demonstrated transition is well fitted such that $r \approx 0$,

$$d(a) \approx \Delta a^\top M\,\Delta a, \qquad M = \tfrac{1}{N} J^\top J \succeq 0. \tag{11}$$

Proof. A first-order expansion of the world model around $a^\star$ gives $P_\phi(z_t, a) \approx P_\phi(z_t, a^\star) + J\Delta a + O(\|\Delta a\|^2)$. Subtracting z^v_{t+H} and using equation 9 yields $P_\phi(z_t, a) - z^v_{t+H} \approx J\Delta a + r$. Expanding the squared norm of equation 5 under the Gauss–Newton approximation and ignoring higher-order curvature terms gives equation 10. Since $J^\top J$ is a Gram matrix, $M \succeq 0$. □

Proposition 1 shows that consequence feedback induces a transition-dependent metric in action space. Directions associated with larger singular values of J produce larger predicted changes in the future and therefore receive larger consequence penalties, whereas directions with weak influence on the predicted transition are penalized less strongly. This differs from standard action regression, which measures deviations directly in the original action coordinates. The world-model objective $\mathcal{L}_{\mathrm{wm}}$ reduces the demonstrated-transition residual r, anchoring this local consequence geometry to observed state transitions.

A.2 Optimization Dynamics of Comparative Consequence Refinement

We next analyze how comparative consequence refinement converts relative consequence differences into optimization signals. Define

$$u_t = \frac{d_1 - d_{\mathrm{ref}} + m}{T},$$

and $w_t = \sigma(u_t)$, where $\sigma(\cdot)$ denotes the sigmoid function.

Proposition 2 (Comparative consequence weighting) *For the CCR objective $\mathcal{L}_{\mathrm{CCR}} = T\,\mathrm{softplus}(u_t)$, with d_{ref} stop-gradient, its gradients with respect to the policy parameters θ and world-model parameters ϕ are*

$$\nabla_\theta \mathcal{L}_{\mathrm{CCR}} = w_t \nabla_\theta d_1, \quad \textit{and} \quad \nabla_\phi \mathcal{L}_{\mathrm{CCR}} = w_t \nabla_\phi d_1.$$

Proof. Since $\frac{\partial}{\partial x}\,\mathrm{softplus}(x) = \sigma(x)$, and d_{ref} is stop-gradient, $\frac{\partial \mathcal{L}_{\mathrm{CCR}}}{\partial d_1} = \sigma(u_t) = w_t$. Applying the chain rule with respect to θ and ϕ gives the stated gradients. □

The weighting factor w_t is determined by the relative consequence gap. When the refined proposal has a larger discrepancy than its reference, CCR assigns a stronger update; as the refined

proposal approaches or surpasses the reference, the update is attenuated. CCR therefore preserves the consequence-improvement direction ∇d_1 while allocating feedback according to the current need for refinement. The same weighting also updates the world model through the refined consequence branch. This allows consequence prediction to co-evolve with the policy's changing action proposals, while $\mathcal{L}_{\mathrm{wm}}$ continuously grounds the world model in observed demonstrated transitions. The benefit of this joint optimization is examined empirically in Appendix D.3.

B Implementation and Training Details

B.1 Model Architecture

We initialize the vision–language backbone from Qwen3-VL-4B (Bai et al., 2025), which couples a SigLIP-2 vision encoder of 24 layers and hidden width 1,024 with a 36-layer Qwen3 language model of hidden width 2,560 through an MLP-based merger. Given an action horizon H, we extend the instruction with H action queries and map their final hidden states through a two-layer MLP to obtain the feedforward proposal a_t^{ff}, while the full token sequence serves as cross-attention context for the action head. The consequence space is a frozen DINOv3 ViT-S/16 distilled from a ViT-7B teacher trained on LVD-1689M (Siméoni et al., 2025). Each view is resized to 224×224 and encoded into 14×14 patch tokens of dimension 384; we drop the class and register tokens so that only spatially grounded features define the space in which consequences are measured. The patch tokens of all views are projected to the backbone width and appended to the token sequence, and proprioception is embedded by a two-layer MLP, which together give the policy condition c_t.

The action head is a 16-layer diffusion transformer of width 768 with 12 heads, alternating cross-attention over c_t with self-attention over the action sequence. A three-layer MLP embeds actions and 32 learned queries decode the action chunk. We set $\tau = 0.9$, weight the coarse and refined regression terms of equation 2 equally, and follow equation 3 at test time, so that producing an action chunk costs exactly two forward passes.

The world model P_ϕ lifts each DINO token to width 768, adds a patch-position embedding together with a global action embedding obtained from the flattened action chunk, applies four residual self-attention blocks with eight heads, and projects back to the 384-dimensional DINO space. Because the chunk enters as a single conditioning vector, the model stays compact: at roughly 31M parameters it is two orders of magnitude smaller than the policy it supervises, and sharing it across proposals guarantees that all interventions are scored under one world model.

B.2 Optimization Dynamics

Table 6 summarizes the coefficients of the three objectives in equation 8 and the modules updated by each term. Direct action supervision $\mathcal{L}_{\text{policy}}$ updates the policy, while $\mathcal{L}_{\text{wm}}$ trains the world model on observed demonstrated transitions. Comparative consequence refinement $\mathcal{L}_{\text{CCR}}$ operates through the refined consequence branch: its auxiliary reference is stop-gradient, while gradients through d_1 jointly update the policy and world model. This allows consequence prediction to evolve together with the refined action distribution while remaining grounded by demonstrated-transition supervision. DINOv3 remains frozen throughout. We further evaluate the effect of this joint optimization in Appendix D.3.

B.3 Training Details

We train on 8 NVIDIA H100 GPUs with BF16 precision and ZeRO Stage 2 sharding, at a per-GPU batch size of 16 and a global batch size of 128. Optimization uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 10^{-8}, and gradient-norm clipping at 1.0. The pretrained backbone is fine-tuned at 10^{-5} and the randomly initialized modules at 10^{-4}, both decayed to 10^{-6} by a cosine schedule, while DINOv3 receives no gradients. The remaining settings follow each embodiment. Across all three suites, the world model predicts the terminal consequence after one complete action chunk, with the consequence horizon aligned to the action horizon.

LIBERO. The policy observes the primary and wrist views together with a 7-dimensional end-effector pose and gripper state, and emits 8-step chunks of 7-dimensional delta end-effector commands. We train for 60k steps with 5k warmup steps.

RoboCasa-GR1. This suite provides a single egocentric view and a 29-dimensional joint state covering the two arms, two hands, and waist. We expand the joint angles into 58 sine–cosine features to remove their wrap-around discontinuity, and predict 16-step chunks in the original joint space. The larger embodiment and broader task set call for a longer schedule of 120k steps with 15k warmup steps.

Table 6: **Optimization Configuration.** Coefficients of the three objectives in equation 8 and the modules each term updates.

Term	Coefficients	Policy	World model
Action supervision $\mathcal{L}_{\text{policy}}$	$\lambda_{\text{ff}} = 0.1$	✓	✗
World-model learning $\mathcal{L}_{\text{wm}}$	$\lambda_{\text{wm}} = 1.0$	✗	✓
Comparative consequence refinement $\mathcal{L}_{\text{CCR}}$	$\lambda_{\text{CCR}} = 1.0, T = 0.1, m = 0$	✓	✓

Real-world tasks. Here the policy reads the head, waist, and right-wrist views and a 15-dimensional state of right-arm joint positions, joint efforts, and gripper state. Each 50-step chunk spans one second at the 50 Hz control rate, and its seven arm commands are expressed relative to the current joint configuration, since insertion requires corrections far smaller than the absolute joint range; the gripper command stays absolute and binary. We train for 50k steps with 1k warmup steps.

C Experimental Details

C.1 Simulation Benchmarks

LIBERO. We follow the official protocol of LIBERO (Liu et al., 2023) and train one policy on the union of the Spatial, Object, Goal, and Long suites, drawing the four suites with equal probability; each contributes 10 tasks with 50 demonstrations per task. At evaluation time, both views are rendered at 256×256, and the simulator is stepped with 10 no-op actions so that objects settle before control begins. We run 50 rollouts per task with the built-in success predicate, and cap episodes at the official limits of 220, 280, 300, and 520 steps for Spatial, Object, Goal, and Long.

RoboCasa-GR1. We adopt the official tabletop split of RoboCasa-GR1 (Nasiriany et al., 2024), which pairs the Fourier GR-1 humanoid with 24 tasks: 18 pick-and-place tasks that transfer objects between containers such as trays, plates, and baskets, and 6 articulated tasks that additionally require closing a cabinet, drawer, or microwave after placement. A single policy is trained on all of them with equal task weighting and 1,000 demonstrations per task. Evaluation again uses 50 rollouts per task and the benchmark's own success predicate, and we report the mean over the 24 tasks.

C.2 Real-World Experiments

Data collection. We collect demonstrations on the wheeled dual-arm humanoid robot platform shown in Fig. 3 through teleoperation. In each demonstration, the operator removes a USB plug from one of three receptacle positions on the left and inserts it into the first receptacle on the right. We collect 200 demonstrations for each source position, resulting in 600 demonstrations and approximately 2.5 hours of robot data. Each trajectory records the head, waist, and right-wrist RGB streams, right-arm proprioception, and commanded actions at 50 Hz, and is stored in the LeRobot format for training.

Evaluation conditions. Fig. 4 illustrates the four conditions, each evaluated over 50 rollouts with binary task success.

- **Fixed** initial states: every rollout begins from the same calibrated right-arm pose.
- **Randomized** initial states: at reset the end-effector is displaced by $\Delta x \in [-0.05, 0]$ m and $\Delta y, \Delta z \in [-0.05, 0.05]$ m, with its orientation preserved.
- **Background** variations: both the scene behind the workspace and the color of the USB plug are changed.
- **Lighting** variations: the room lights are switched off and the workspace is lit by a handheld flashlight.

C.3 Consequence Signal Evaluation Details

This section details the protocol behind Fig. 2. Panels (a) and (b) are logged online during LIBERO training on the minibatches used for optimization; faint traces are individual logged values and solid traces are centered moving averages over 5% of the logs. Panels (c) and (d) evaluate a fixed 60k-step checkpoint on 640 current–future pairs, obtained by sampling 20 episodes from each of the four suites and eight temporally separated timesteps per episode.

Comparative refinement diagnostics. Panel (b) tracks two quantities. The first is $\Pr[d_1 < d_{\text{ref}}]$, the fraction of samples for which the refined proposal produces a predicted consequence closer to the demonstrated future than the best auxiliary reference. Here, $d_{\text{ref}} = \min\{d_0, d_{\text{ff}}\}$, and we denote the corresponding auxiliary proposal by $a^{j_{\text{ref}}}$, where $j_{\text{ref}} = \arg\min_{j \in \{0,\text{ff}\}} d(a^j)$. The second quantity measures action–consequence agreement. Specifically, we compare the consequence-space relation $d_1 < d_{\text{ref}}$ with the corresponding action-space relation $\|a^1 - a^\star\|_2^2 < \|a^{j_{\text{ref}}} - a^\star\|_2^2$, and report the fraction of samples for which the two comparisons agree. This measures whether consequence-space refinement and direct action-space proximity provide consistent preferences over the same pair of proposals. As training progresses, all proposals improve under direct action supervision and their differences gradually shrink, causing the comparative signal to stabilize and slightly weaken late in training even as their absolute consequence errors continue to decrease.

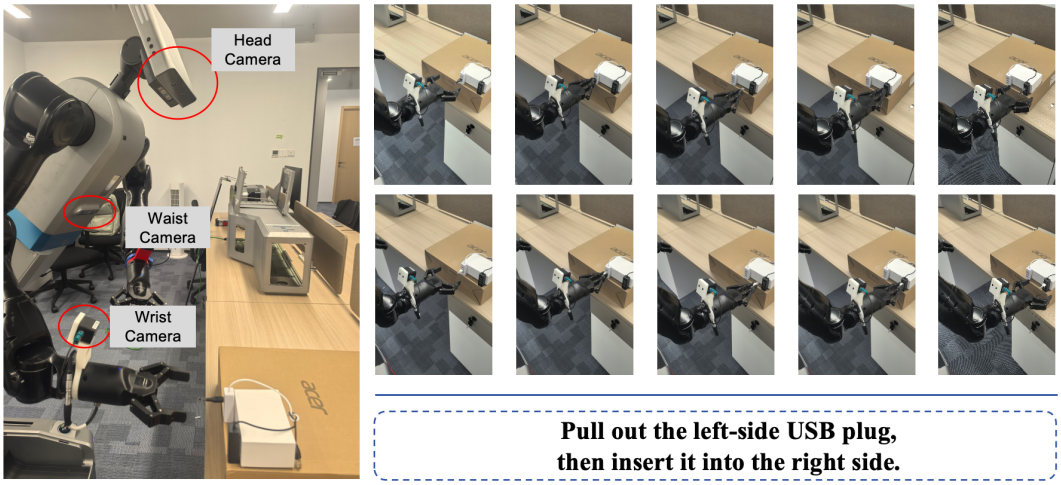
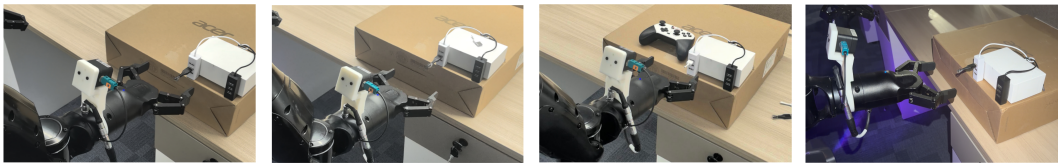


Figure 3: **Real-world experimental setting.** We use a wheeled dual-arm robot equipped with head-, waist-, and right-wrist-mounted RGB-D cameras. The task requires removing a USB plug from one of three receptacles on the left and inserting it into the first receptacle on the right. Only the right arm is controlled in this experiment.



(a) fixed initial states (b) randomized initial states (c) background variations (d) lighting variations

Figure 4: **Real-world evaluation conditions.** We evaluate robustness under fixed and randomized initial states, as well as background and lighting perturbations.

Action interventions. Each condition in panel (c) changes only the action passed to the world model, holding the current observation and the target future fixed. *Persistence* makes no prediction and reuses the current features as the future, measuring how much of the transition a static scene already explains. *True action* supplies the matched demonstration chunk. *Zero action* supplies an all-zero chunk, testing whether the current view alone accounts for the future. *Shuffled action* supplies a valid chunk drawn from another sample, testing state–action correspondence under in-distribution action statistics. *Negated action* negates every dimension of the matched chunk, a strong intervention on action direction.

Action retrieval. Panel (d) builds eight candidate chunks for each transition: the matched action, and seven actions obtained by cyclically shifting the action chunks of the evaluation set, which preserves in-distribution action statistics while breaking their correspondence with the current state. Scoring every candidate against the same observed future and ranking the resulting discrepancies gives the rank of the matched action, so rank one means that the action which actually occurred best explains the observed transition. Chance level is $1/8$, and the matched action is ranked first in 95.8% of samples, with a mean rank of 1.05.

C.4 Baselines

For simulation benchmarks, we directly use published results when the corresponding paper reports results under the same benchmark and evaluation protocol. Otherwise, we reproduce the method using its official open-source implementation and match our training data, number of optimization steps, observation inputs, and evaluation protocol as closely as possible. For the real-world task, all baselines are trained from their official implementations using the same 600 demonstrations, and

evaluation conditions as ActEffect. For Fast-WAM (Yuan et al., 2026c) and GigaWorld-Policy (Ye et al., 2026a), we follow their authors' recommended setting and decode actions without generating future video at deployment.

C.5 Inference Latency Protocol

We measure simulation latency on a single NVIDIA L40S GPU and real-world latency on a single NVIDIA RTX PRO 6000 GPU. All models run in BF16 with batch size one. After warm-up, we measure the wall-clock time of a complete action-chunk policy call and synchronize the GPU around the timed region. The reported latency includes input preprocessing, Qwen3-VL inference, current-frame DINO encoding and projection, action generation, and action postprocessing, while excluding model loading, network communication, and environment or robot stepping. For ActEffect, each timed call contains exactly two forward passes of the action head and does not execute the controlled world model.

D Additional Experimental Results

D.1 Results on LIBERO-PLUS

LIBERO-PLUS (Fei et al., 2025) extends LIBERO to systematically evaluate policy robustness under controlled distribution shifts. It introduces perturbations along seven dimensions, including camera viewpoint, robot initial state, language rephrasing, lighting, background texture, sensor noise, and object layout, while retaining the original LIBERO tasks and demonstrations. We evaluate ActEffect in a zero-shot setting by directly transferring the policy trained on standard LIBERO, without any further adaptation on LIBERO-PLUS.

As shown in Table 7, ActEffect achieves an overall success rate of 80.3%, outperforming all compared WAM baselines and ranking second among all methods. In particular, ActEffect shows strong robustness to appearance-related perturbations, reaching 96.7% under lighting changes and 95.8% under background variations, while also maintaining competitive performance across camera, language, noise, and layout shifts. Compared with Fast-WAM, ActEffect improves the overall success rate from 51.5% to 80.3%, with especially large gains under camera, background, and sensor-noise perturbations. These results suggest that ActEffect retains strong robustness under diverse distribution shifts, consistent with its use of action-sensitive consequence feedback rather than relying solely on predictive context for action generation.

Table 7: **LIBERO-PLUS Results.** Success rates (%) under seven controlled perturbation dimensions and the overall aggregate score.

Method	Camera	Robot	Language	Light	Bg.	Noise	Layout	Overall ↑
VLA								
UniVLA (Bu et al., 2025)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
π_0 (Black et al., 2024)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
OpenVLA-OFT (Kim et al., 2025a)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
GR00T-N1.6 (Bjorck et al., 2025)	–	–	–	–	–	–	–	72.6
$\pi_{0.5}$ (Intelligence et al., 2025)	**75.4**	**77.5**	85.6	**96.9**	<u>94.6</u>	**89.7**	**85.7**	**85.7**
WAM								
LDA-1B (Lyu et al., 2026)	–	–	–	–	–	–	–	45.5
Fast-WAM (Yuan et al., 2026c)	16.4	44.5	68.9	78.2	53.7	37.7	60.7	51.5
Robust-WAM (Yan et al., 2026)	28.7	43.4	73.6	90.2	62.6	56.8	67.8	58.9
Faster-WAM (Zhao et al., 2026b)	53.8	<u>71.6</u>	<u>94.7</u>	96.3	61.3	63.6	79.1	73.6
LAWA (Li et al., 2026c)	<u>69.2</u>	66.7	62.8	96.2	64.5	<u>85.5</u>	78.6	74.4
SG-WAM (Zhao et al., 2026a)	58.6	48.9	81.4	89.8	86.1	80.7	74.2	73.0
DreamWAM (Yuan et al., 2026b)	53.8	63.6	**94.8**	<u>96.7</u>	71.6	67.2	<u>80.7</u>	75.5
Ours								
ActEffect	66.4	55.2	87.1	<u>96.7</u>	**95.8**	82.6	78.8	<u>80.3</u>

D.2 Qualitative Visualization of Consequence Predictions

To complement the quantitative consequence-signal analysis in Sec. 5.4, we visualize the action-conditioned feature changes predicted by the world model under different action interventions. As shown in Fig. 5, conditioning on the demonstrated action recovers the dominant spatial structure of the observed feature transition from t to $t + 8$. The policy-predicted action induces a similar transition pattern. In contrast, negating the action produces a pronounced feature change in the opposite direction, while the zero action yields only weak changes, both of which are qualitatively consistent with the corresponding interventions. **These visualizations show that the world model responds meaningfully to action interventions rather than predicting the future from the current state alone, providing action-sensitive consequence signals that support comparative refinement.**

D.3 Additional Ablations

We further investigate three design choices in ActEffect: how future-modeling supervision interacts with policy learning, which future states are used as consequence targets, and how the world model is optimized during comparative consequence refinement. All additional ablations are conducted on LIBERO, with results summarized in Table 8.



Figure 5: **Qualitative consequence predictions.** Observed and predicted DINO feature changes under demonstrated, policy-predicted, negated, and zero action chunks from the same current state.

Table 8: **Additional Ablation Results.** Average success rates (%) on LIBERO under different consequence-modeling, temporal-target, and world-model optimization designs.

Consequence Modeling		*Temporal Consequence Target*		*World-Model Learning*	
Variant	Success (%) ↑	Variant	Success (%) ↑	Variant	Success (%) ↑
Implicit, VLM features	97.4	All future frames	98.2	Frozen during CCR	98.1
Implicit, DINO + language	97.7	Multi-step, stride 2	97.8	Trainable during CCR	**99.0**
Implicit, DINO features	98.0	Multi-step, stride 4	98.4		
Explicit evaluation (Ours)	**99.0**	Terminal $t + H$ (Ours)	**99.0**		

(1) Consequence modeling. We compare ActEffect's explicit proposal-level consequence evaluation with three implicit future-modeling variants inspired by FLARE (Zheng et al., 2025). Specifically, we introduce learned future tokens into the DiT action head, allowing them to interact with intermediate action representations through self-attention and supervising them to predict future visual features. We evaluate three variants: the first predicts VLM future features, corresponding to a task-oriented consequence space, while masking direct interaction between future tokens and task VLM tokens; the second replaces the prediction targets with DINO features while retaining the same attention mask; and the third also predicts DINO features but allows future tokens to attend directly to task VLM tokens through cross-attention, introducing task-semantic conditioning. All three variants use implicit future-modeling supervision without explicit proposal-level consequence evaluation or comparative consequence refinement.

Comparing the first two variants, replacing VLM future features with DINO features improves success from 97.4% to 98.0%, supporting the importance of a consequence space decoupled from task-oriented representations. Comparing the two DINO-based variants, allowing direct interaction with task VLM tokens reduces success from 98.0% to 97.7%, suggesting that excluding direct task-semantic conditioning benefits consequence modeling. Finally, full ActEffect improves success from 98.0% to 99.0% over the language-masked implicit variant, supporting the benefit of explicitly evaluating complete action proposals and converting their relative predicted consequences into feedback for policy refinement.

(2) Temporal consequence target. We further examine whether predicting intermediate future states is necessary for consequence feedback. Using the eight-step action horizon on LIBERO, we compare predicting all eight future frames, four frames at a stride of 2, two frames at a stride of 4, and only the terminal frame at $t + 8$. The corresponding success rates are 98.2%, 97.8%, 98.4%, and 99.0%, respectively. The intermediate-state variants provide no consistent improvement over terminal-only prediction, while the terminal target achieves the best performance. These results suggest that, within the short action horizon considered here, the terminal state is sufficient to represent the consequence of the complete action chunk, supporting the terminal consequence formulation adopted in ActEffect.

(3) Joint world-model optimization. We further examine whether the world model should continue to adapt during comparative consequence refinement. In a frozen world-model variant, we first train the world model using demonstrated transitions and then freeze its parameters, such that $\mathcal{L}_{\mathrm{CCR}}$ updates only the policy. This variant achieves 98.1% success on LIBERO, compared with 99.0% for full ActEffect. The result suggests that jointly optimizing the world model with the policy provides an additional benefit, allowing consequence prediction to adapt alongside the policy's evolving action proposals while remaining grounded by demonstrated-transition supervision.

D.4 Corrective Behavior in Contact-Rich Manipulation

We further qualitatively examine how consequence feedback affects closed-loop behavior in contact-rich manipulation. We compare full *ActEffect* with *ActEffect (w/o Feedback)*, which removes $\mathcal{L}_{\mathrm{CCR}}$, so that the policy is trained only with direct action supervision and therefore reduces to a standard VLA. Importantly, the two variants use the same policy architecture at inference, with no world model or consequence evaluation involved. Their behavioral difference therefore arises entirely from the training-time consequence supervision.

Figure 6 presents a representative real-world rollout comparison in which both policies encounter a slight misalignment during USB insertion. In this case, the variant without consequence feedback continues pushing from the misaligned configuration and eventually fails. In contrast, ActEffect performs a corrective motion, re-aligns the USB plug with the receptacle, and successfully completes the insertion. Since the world model is absent at deployment, this correction is not produced by online model-based look-ahead planning. Instead, it suggests that training-time consequence feedback internalizes sensitivity to unfavorable action outcomes into the policy, resulting in feedback control-like closed-loop corrective behavior. Such behavior is particularly valuable in contact-rich manipulation, where small action or pose deviations can lead to substantially different physical outcomes. This example illustrates ActEffect's ability to recover from occasional insertion misalignment, rather than suggesting that corrective adjustment is required for every successful insertion.

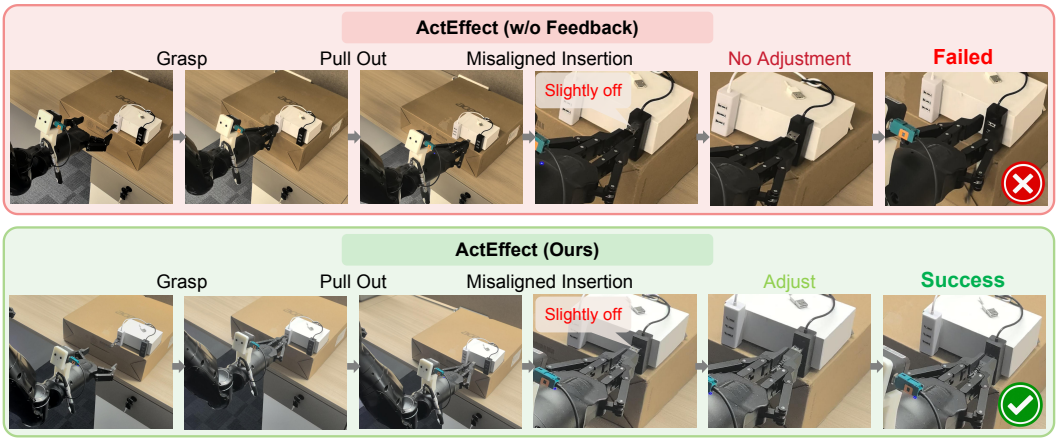


Figure 6: **Effect of consequence feedback on corrective behavior.** After a slight misalignment during USB insertion, ActEffect adjusts the insertion pose and succeeds, whereas the variant without consequence feedback continues without correction and fails.