

BadWAM: When World-Action Models Dream Right but Act Wrong

Qi Li¹ Xingyi Yang² Xinchao Wang^{1†}

¹National University of Singapore ²The Hong Kong Polytechnic University

liqi@u.nus.edu xinchao@nus.edu.sg

[†]Corresponding author.



GitHub



Homepage



Hugging Face

Abstract

World-action models (WAMs) are emerging as a promising foundation for embodied control: rather than predicting actions alone, they learn representations that couple action generation with future world prediction. This coupling is often viewed as a source of robustness, interpretability, and safety, as a robot’s action can in principle be checked against its imagined future. In this paper, we show that this assumption is fragile. We introduce *BadWAM*, a unified framework for modeling and evaluating *World-Action Drift Attacks*: a new class of WAM-specific adversarial attacks that use small visual perturbations to break the alignment between what a WAM imagines and what it executes. *BadWAM* characterizes this attack surface along two natural criteria: attack strength and stealthiness. When the adversary prioritizes disruption, *BadWAM* instantiates an *action-only adversarial attack*, which directly drives the model toward task-failing actions. When the adversary additionally prioritizes stealth, *BadWAM* instantiates an *imagination-preserving adversarial attack*, which seeks to induce harmful action shifts while keeping the model’s predicted future close to its clean imagination. Together, these two attacks capture a spectrum of WAM-specific failures: from overt action hijacking to stealthier cases where the model appears to imagine a plausible future but executes a desynchronized action. We evaluate *BadWAM* across different variants of WAMs. Results show that our attacks substantially reduce task success rates under closed-loop execution. For example, our action-only attack reduces the model performance from 96.5% to 43.1% success. The results of our imagination-preserving attack further exposes a WAM-specific vulnerability: moderate future-preserving regularization can maintain strong attack performance while reducing future imagination drift.

1 Introduction

Robotic foundation models are rapidly moving from action prediction toward *world-action modeling* [1, 12, 13, 22, 27, 29].

A world-action model (WAM) does not merely map observations and language instructions to low-level controls; it also learns representations that couple action generation with future world prediction [22, 27]. This design is attractive for embodied AI. Future prediction can provide richer physical representations, enable planning or verification, and offer an intuitive safety signal: if a robot can imagine the consequences of its actions, then unsafe or inconsistent behavior might be detected before execution [14, 22, 27].

Recent WAMs instantiate this idea in different ways. Some models explicitly generate future observations or latent futures before action prediction, while others use video modeling as an auxiliary training objective and skip explicit future generation at test time [27]. Newer WAM variants further extend this paradigm toward object-addressable manipulation, mobile manipulation, and visual-tactile contact-rich control [5, 18, 25]. Across these designs, the central promise is the same: coupling action with world prediction should make robot behavior more physically grounded. This promise also motivates emerging WAM safety mechanisms that inspect imagined futures, latent rollouts, or visual trajectories to detect unsafe plans [3, 17].

This paper begins with a basic security question: *are WAMs robust to adversarially induced execution failures?* We find that the answer is no. Small visual perturbations can substantially reduce task success across WAM variants. The deeper issue is specific to WAMs: *action generation and future imagination can fail asynchronously*. Even when the imagined future remains plausible, the action output need not remain aligned with it. We show that an attacker can desynchronize a WAM’s action from its future prediction. In other words, the model may still produce plausible future imaginations, yet execute actions that cause the task to fail. This failure mode is qualitatively different from standard adversarial attacks on image classifiers, visual world models, or reactive robot policies [9, 19, 21, 24]. The attacker is not merely changing a label or shifting an action vector; it is exploiting the interface between two coupled capabilities: imagining and acting.

We call this vulnerability *World-Action Drift Attack*. Rather

than treating action and imagination as independent outputs, BadWAM focuses on their alignment. A small visual perturbation can push the model toward a task-failing action while leaving its imagined future visually plausible. This creates a dangerous gap: a robot may appear to imagine a reasonable future, yet execute an action that is no longer aligned with that imagination. In such cases, future prediction alone is not a reliable safety signal.

We instantiate this idea in *BadWAM*, a unified framework for modeling and evaluating world-action drift attacks under black-box access to a deployed WAM. BadWAM characterizes the attack surface along two natural criteria: attack strength and stealthiness. When the adversary prioritizes disruption, BadWAM instantiates an *action-only adversarial attack*, which drives the model toward task-failing actions using only queryable model outputs. This attack captures the overt end of the threat spectrum, where the adversary aims to maximize execution failure. When the adversary additionally prioritizes stealth, BadWAM instantiates an *imagination-preserving adversarial attack*. This attack also operates through black-box queries, but optimizes for harmful action shifts while keeping the model’s predicted future close to its clean imagination. Intuitively, this creates a stealthier failure: the robot appears to imagine a plausible future, but the action it executes has been desynchronized from that future. Together, they expose a spectrum of black-box WAM failures, from overt action hijacking to stealthy imagination-preserving failure.

We evaluate BadWAM on LIBERO [16] and RoboTwin [6]. Our experiments show that WAMs are vulnerable across variants. The action-only attack substantially reduces closed-loop task success, and the imagination-preserving attack exposes a stronger WAM-specific risk: moderate future-preserving regularization can maintain strong attack performance while reducing imagination drift. In our experiments, the action-only attack reduces task success from 96.5% to 43.1%. The imagination-preserving attack reduces task success while keeping future predictions close to clean rollouts. Our analysis further reveals where the attacks operate. The perturbations primarily affect continuous action channels and later portions of the action horizon, rather than merely flipping gripper commands. Episode-level statistics show that failed episodes tend to have larger action shifts, while future-imagination shifts can remain bounded. These findings suggest that WAM failures are not simply caused by visually destroying the model’s imagination. Instead, the attack exploits a decoupling between what the model imagines and what it executes.

This work makes the following contributions:

- We identify *World-Action Drift Attack* as a WAM-specific attack surface, where action outputs can be adversarially hijacked while the model’s predicted futures remain visually plausible.
- We design *BadWAM*, a unified framework with two

attack instantiations: *action-only adversarial attacks* for maximizing execution disruption, and *imagination-preserving adversarial attacks* for causing failures while preserving plausible predicted futures.

- We evaluate BadWAM on closed-loop robot manipulation tasks, showing substantial task degradation under bounded visual perturbations. We also provide detailed analyses of attack dynamics, action-channel shifts, horizon-level effects, imagination preservation, and future-weight trade-offs, offering insight into why WAMs fail under world-action drift attacks.

2 Related Work

2.1 World Action Models

World-action models have recently emerged as a predictive-action paradigm for embodied AI [22]. Unlike conventional visuomotor policies that directly map observations and instructions to actions, WAMs expose future prediction as part of the action-generation process. Let o_t denote the current visual observation, g denote a language instruction or goal, and $a_{t:t+H-1}$ denote an action chunk over horizon H . A conventional reactive policy can be abstracted as

$$a_{t:t+H-1} \sim \pi_{\theta}(o_t, g). \quad (1)$$

By contrast, a WAM additionally models future world states, such as latent future states $z_{t+1:t+K}$ or decoded future frames $v_{t+1:t+K}$. Abstractly, its world-prediction module W_{θ} first imagines a future, and its action module A_{θ} then produces an action sequence conditioned on that imagination:

$$z_{t+1:t+K} \sim W_{\theta}(o_t, g), \quad a_{t:t+H-1} \sim A_{\theta}(o_t, g, z_{t+1:t+K}). \quad (2)$$

In fully joint formulations, the model may instead learn a joint distribution over future states and actions:

$$(z_{t+1:t+K}, a_{t:t+H-1}) \sim p_{\theta}(z_{t+1:t+K}, a_{t:t+H-1} \mid o_t, g). \quad (3)$$

In all cases, the defining feature is that action prediction is coupled, explicitly or implicitly, with a learned model of how the world may evolve.

This coupling can appear in multiple architectural forms. Some WAMs first imagine a future and then infer actions from that imagined future. Others jointly predict actions and future representations in a shared sequence model. Still others use future prediction primarily as a training signal, so that the model benefits from world modeling even if explicit future generation is not used at inference time [27]. Fast-WAM studies this design space and shows that video modeling during training can provide substantial benefits even when test-time future generation is removed [27]. This observation is important for security: even when a model does not expose

future videos at deployment time, the learned action representation may still inherit vulnerabilities from the underlying world-action coupling.

Recent WAM variants further broaden the design space. OA-WAM introduces object-addressable world representations for robust manipulation [18]; ABot-M0.5 extends WAMs to unified mobility-and-manipulation control with action-space disentanglement and train-test consistency mechanisms [5]; and VT-WAM incorporates visual-tactile prediction for contact-rich manipulation [25]. These systems illustrate the rapid expansion of WAMs from visual manipulation to richer embodied settings. As WAMs become more capable and more deeply integrated into robot control stacks, understanding their security properties becomes increasingly urgent and meaningful.

Prior WAM work usually treats future prediction as a beneficial representation, planning interface, or safety signal [3, 17]. A natural hope is that an imagined future can be used to check whether a planned action is physically plausible or safe. Formally, one might imagine a monitor M that inspects the predicted future:

$$\text{safe} = M(z_{t+1:t+K}, g). \quad (4)$$

However, such a monitor only observes the imagined future, not necessarily the alignment between the future and the action that will be executed. This leaves open a WAM-specific security question: can an adversary perturb the observation so that the action changes substantially while the imagined future remains apparently plausible? BadWAM formalizes and evaluates this gap in Section 4.

2.2 Attacks against Embodied AI Systems

Adversarial examples were first studied extensively in image classification, where small input perturbations can cause large changes in model predictions [9, 19, 24]. Subsequent work extended these attacks to black-box and query-limited settings, including gradient-free and score-based estimation methods [4, 11, 26]. Physical-world attacks further showed that adversarial patterns can remain effective under real-world transformations, for example through adversarial patches or robust physical perturbations [2, 7]. BadWAM builds on this broader adversarial robustness literature but targets a different object: not a classifier label or static perception output, but the closed-loop action behavior of a WAM.

Attacks on embodied AI systems are particularly concerning because model failures can translate into physical consequences. Prior work has studied attacks on perception modules, reinforcement-learning agents, and embodied policies, including perturbations that alter observations, timing, or environmental cues [8, 15]. These attacks typically target reactive decision-making. For a reactive policy π_θ , an

observation attack can be written as

$$\max_{\|\delta\|_\infty \leq \epsilon} D(\pi_\theta(o_t + \delta, g), \pi_\theta(o_t, g)), \quad (5)$$

or, in closed-loop evaluation, as minimizing task reward or success. BadWAM naturally generalizes this view to predictive-action models whose outputs include both actions and imagined futures. The relevant question is no longer only whether the action changes, but whether the action changes while the imagination remains plausible.

Recent work has begun to examine attacks and safety risks for world models and WAMs. BadWorld attacks visual world models by perturbing context images to degrade future roll-outs under uncertain controls [21]. Physical-conditioned attacks study how perturbing condition channels in generative world models can induce semantic or decision-level distortions [10]. JailWAM proposes a safety evaluation framework for WAM jailbreaks in robot control, focusing on unsafe or destructive behaviors represented through visual trajectories [17]. These works highlight that predictive models in embodied systems introduce new attack surfaces beyond conventional action policies.

Closest to our setting is recent work on oracle-level integrity attacks against imagine-then-act world models [3]. That work studies VLA-style and world-model-based systems, where an imagined future is consumed by downstream oracles such as safety gates, MPC planners, or imagine-then-check verifiers. It shows that corrupting the trusted imagination can compromise systems that rely on future predictions. BadWAM studies a complementary failure mode. Rather than attacking the imagined future as the primary object, we target the alignment between imagination and action in deployed WAM controllers. We show that action and imagination can be adversarially desynchronized: the predicted future may remain close to the clean prediction while the executed action changes enough to cause task failure.

This distinction is central to our threat model. If a safety monitor trusts plausible imagined futures, then a purely imagination-based check may miss an attack that preserves future predictions while hijacking actions. BadWAM therefore treats the alignment between action and imagination as the security-critical object. Our action-only adversarial attack prioritizes task disruption, while our imagination-preserving adversarial attack explicitly studies the trade-off between attack strength and stealthiness.

3 Threat Model

We study inference-time attacks against deployed WAM-based robot policies. Following the notation introduced above, the robot observes o_t and receives an instruction or goal g at each replanning step. The WAM then outputs an action chunk $a_{t:t+H-1}$ and, depending on the model interface, may also expose an imagined future in latent form $z_{t+1:t+K}$ or decoded

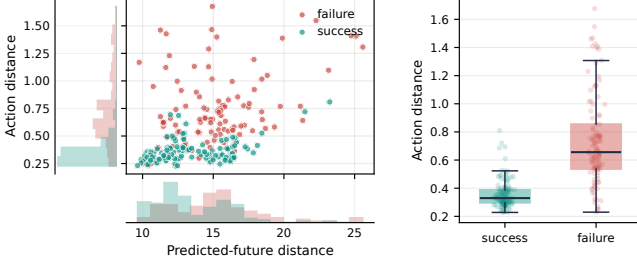


Figure 1: Empirical motivation for world-action adversarial attacks. Failed episodes tend to have larger action shifts, while predicted-future shifts overlap across successful and failed executions. This motivates attacking the alignment between action and imagination.

video form $v_{t+1:t+K}$. The robot executes part of the predicted action chunk, observes the environment again, and repeats this process in closed loop.

Adversarial capability. The adversary can perturb the visual observation before it is processed by the WAM. We write the perturbed observation as

$$\tilde{o}_t = \text{clip}(o_t + \delta_t), \quad \|\delta_t\|_\infty \leq \epsilon. \quad (6)$$

The perturbation is bounded so that the attacked observation remains visually close to the clean observation. For multi-camera inputs, we treat o_t as the concatenation of all camera observations and apply the same norm constraint to the full visual input. The adversary does not modify the language instruction g , the robot state, the environment dynamics, the model parameters, or the training data.

We consider two access levels. In the *action-only* setting, the adversary can query the WAM and observe only its output action chunk. This captures black-box access to a deployed policy API or robot inference stack. In the *imagination-visible* setting, the adversary can additionally observe the model’s imagined future, either as latent rollouts or decoded future video. This captures systems where future prediction is exposed for planning, debugging, monitoring, or imagine-then-check safety verification [3, 28]. In both settings, the adversary does not require gradients or white-box access to model weights.

Adversarial objective. The adversary’s primary goal is untargeted task failure: the robot should fail to complete the instructed task under closed-loop execution. The attack is successful if a perturbation causes the robot to execute a sequence of actions that reduces task success, even though the instruction and environment remain unchanged.

BadWAM additionally studies stealthiness. A non-stealthy attack may simply maximize action disruption. A stealthier attack should degrade execution while keeping the WAM’s imagined future close to its clean prediction. This is important because future prediction is often treated as a safety signal: a monitor may inspect the imagined future and accept the action if the future appears plausible [3]. BadWAM targets

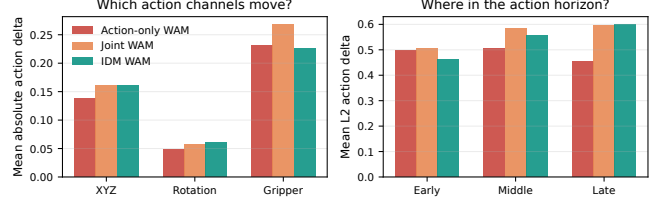


Figure 2: Action-only adversarial attack produces structured action shifts. Across three WAM variants, the attack primarily perturbs continuous action channels and specific portions of the action horizon, rather than acting as uniform output noise.

precisely this gap between what the WAM imagines and what it executes.

Non-goals. We do not study training-time poisoning, model extraction, reward hacking, prompt injection, or attacks that physically modify the robot’s environment. We also do not assume that the attacker can directly command the robot or alter the controller after actions are produced. Our focus is narrower: small bounded observation perturbations that exploit the world-action interface of WAMs at inference time.

4 BadWAM

BadWAM is a unified framework for modeling and evaluating world-action drift attacks. As shown in Figure 3. The framework treats a WAM as a queryable predictive-action system and optimizes bounded visual perturbations at each replanning step. Its two attack instantiations correspond to two points on the attack-strength/stealthiness spectrum: an *action-only adversarial attack* that prioritizes execution failure, and an *imagination-preserving adversarial attack* that additionally constrains predicted-future drift.

4.1 Design Motivation

BadWAM is motivated by a simple empirical observation: action and imagination are coupled in WAMs, but they are not inseparable. In our closed-loop traces, task failures tend to exhibit larger action shifts than successful episodes, while the corresponding imagination shifts can substantially overlap. This suggests that the security-critical object is not only the action or the imagination alone, but their alignment. Figure 1 visualizes this phenomenon under our imagination-preserving attack. This motivation is also consistent with recent diagnostic evidence that WAM action heads can rely on future-predictive latents in a task-phase-dependent manner [20]. BadWAM asks a complementary security question: whether this coupling can be adversarially utilized.

We therefore formulate BadWAM around two distances. D_{act} measures the deviation between clean and attacked action chunks, and D_{img} measures the drift between clean and attacked imagination. These distances are interface-level quantities: they do not require a particular WAM architecture,

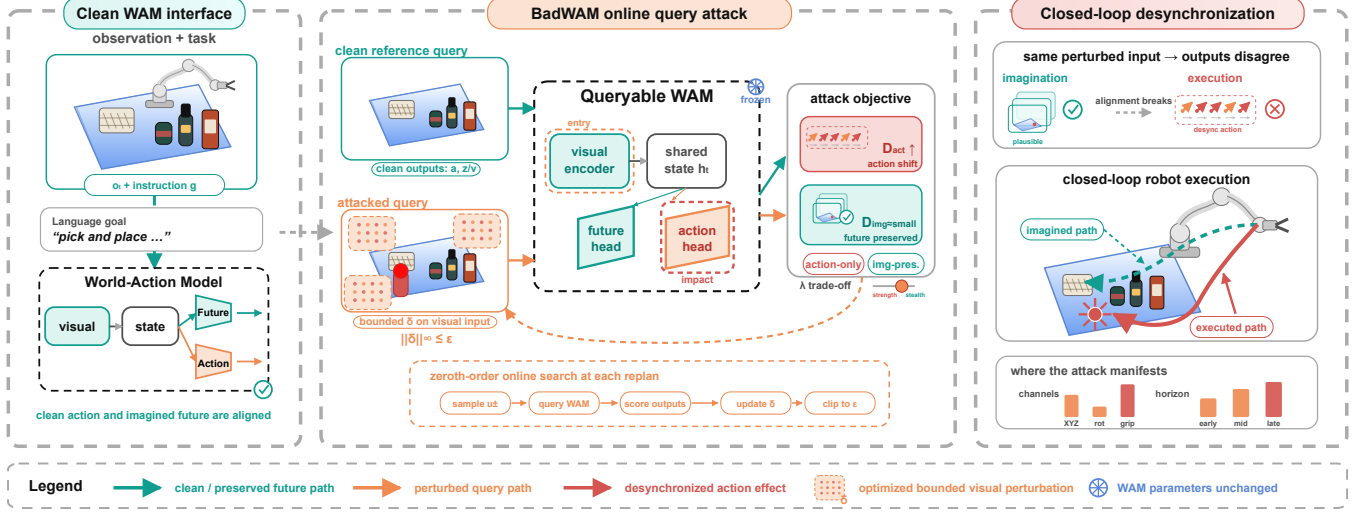


Figure 3: Overview of BadWAM. BadWAM injects a small visual perturbation into model observations and performs query-based online search over a frozen WAM. The optimized trigger disrupts the action prediction pathway while preserving or minimally altering visual rollout predictions, leading to world-action adversarial attacks during closed-loop execution.

only access to the corresponding outputs. Conceptually, a stealthy adversarial attack seeks large action deviation under bounded visual perturbations while keeping imagination drift below a monitorable threshold:

$$\max_{\|\delta\|_{\infty} \leq \epsilon} D_{\text{act}}(a^{\delta}, a) \quad \text{s.t. } D_{\text{img}}(z^{\delta}, z) \leq \tau_{\text{img}}. \quad (7)$$

Equation 7 defines the core BadWAM attack surface. The two attacks below instantiate different relaxations of this objective.

4.2 Action-Only Adversarial Attack

The action-only attack captures the high-strength end of BadWAM. It treats the WAM as a queryable controller and focuses purely on execution disruption. Rather than enforcing any constraint on the imagined future, it searches for a bounded visual perturbation that maximizes the deviation between clean and attacked action chunks:

$$\delta_t^* = \arg \max_{\|\delta_t\|_{\infty} \leq \epsilon} D_{\text{act}}(a_{t:t+H-1}^{\delta}, a_{t:t+H-1}). \quad (8)$$

The clean action chunk is used only as a reference; the attacker does not need ground-truth expert actions or task success labels during optimization. This makes the attack applicable to deployed policy interfaces where the observable output is the action command sent to the robot controller.

Although simple, this objective is already highly disruptive. In our closed-loop experiments, the action-only attack reduces one WAM variant from 96.5% to 43.1% task success. We therefore use it as the high-strength endpoint of BadWAM. At the same time, the attack is not merely unstructured output corruption: the induced shifts are unevenly distributed across both action channels and temporal horizon segments.

For example, Figure 2 shows that the perturbations concentrate on task-relevant channels such as translation and gripper commands, while their temporal concentration varies across WAM variants. This motivates logging channel-level and horizon-level action statistics in our evaluation.

4.3 Imagination-Preserving Adversarial Attack

The action-only objective provides no control over what happens to the imagined future. For WAM deployments that expose imagined futures to planners, monitors, or imagine-then-check safety gates, this is insufficient: a stealthier attack should degrade execution while preserving the model’s clean imagination. BadWAM therefore instantiates an imagination-preserving attack by solving the Lagrangian relaxation of Equation 7:

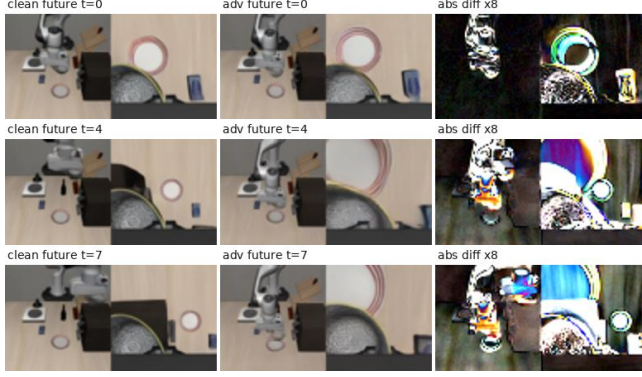
$$\delta_t^* = \arg \max_{\|\delta_t\|_{\infty} \leq \epsilon} D_{\text{act}}(a_{t:t+H-1}^{\delta}, a_{t:t+H-1}) - \lambda D_{\text{img}}(z_{t+1:t+K}^{\delta}, z_{t+1:t+K}). \quad (9)$$

The coefficient λ controls the trade-off between attack-strength and stealthiness. When $\lambda = 0$, the objective reduces to pure action disruption. As λ increases, the optimizer is discouraged from changing the imagined future, making the attack more stealthy but potentially less damaging.

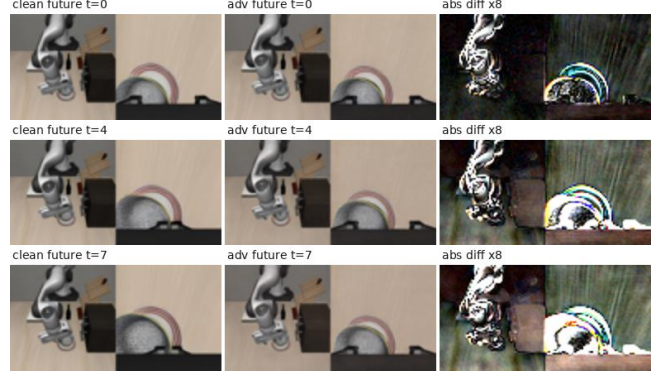
When decoded future videos are available, we instantiate the imagination distance as an average frame-level distance:

$$D_{\text{img}}(v_{t+1:t+K}^{\delta}, v_{t+1:t+K}) = \frac{1}{K} \sum_{k=1}^K d_{\text{frame}}(v_{t+k}^{\delta}, v_{t+k}). \quad (10)$$

The same objective can be applied to latent futures by replacing the frame-level distance with a latent-space distance. In



(a) Imagined future without preservation.



(b) Imagined future with preservation.

Figure 4: Qualitative illustration of the imagination-preserving objective on IDM WAM. Each panel compares predicted futures under clean and adversarial observations at selected future steps. *abs diff* denotes the absolute pixel difference between clean and adversarial predictions, amplified by $8\times$ for visibility. Both variants induce action-space failure, but the preservation term keeps the adversarial future more consistent with the clean imagination.

either case, the attack targets a WAM-specific failure: the robot may appear to imagine a plausible future while executing a desynchronized action.

4.4 Query-Based Online Optimization

BadWAM does not backpropagate through the WAM or access model parameters. At each replanning step, the attacker treats the WAM as a queryable input-output system and optimizes the attack objective with zeroth-order finite-difference queries. Concretely, for a current perturbation δ_t , the attacker samples random directions u_i and evaluates the scalar objective under positively and negatively perturbed observations. This gives the estimator

$$\widehat{\nabla}J(\delta_t) = \frac{1}{m} \sum_{i=1}^m \frac{J(\delta_t + cu_i) - J(\delta_t - cu_i)}{2c} u_i, \quad (11)$$

where J denotes either the action-only objective or the imagination-preserving objective, c is the finite-difference radius, and m is the number of sampled directions. The perturbation is then updated and projected back into the feasible ℓ_∞ ball:

$$\delta_t \leftarrow \Pi_{[-\varepsilon, \varepsilon]} \left(\delta_t + \eta \widehat{\nabla}J(\delta_t) \right). \quad (12)$$

This optimization is zeroth-order because it only uses objective values computed from WAM outputs; it does not require gradients, model weights, or training data. In our implementation, the random directions follow a simultaneous perturbation strategy [23, 26]. The same optimizer supports both BadWAM instantiations by changing only the scalar objective J : the action-only attack scores action deviation, while the imagination-preserving attack scores action deviation together with future-prediction consistency.

For each replanning step, BadWAM keeps the best perturbation found within the query budget. The final adversarial observation is then used to produce the action chunk executed

by the robot. This online design avoids training a separate attack model and naturally adapts to the current observation, instruction, and robot state.

Qualitative Example Figure 4 qualitatively illustrates the role of the preservation term. Both variants can induce action failure, but adding preservation keeps the predicted future closer to the clean imagination. This is precisely the WAM-specific failure mode BadWAM targets: the model’s imagination remains plausible while the executed action is desynchronized.

4.5 Closed-Loop Execution and Measurements

BadWAM is evaluated under closed-loop execution. The attack is recomputed at each replanning step, the WAM outputs an attacked action chunk, and the robot executes the selected actions before the next replan. This setting is more faithful than single-step action comparison because small action shifts may compound over time and cause task failure only after multiple replans.

We report three groups of measurements. First, we measure task-level attack effectiveness using closed-loop success rate and induced failures relative to clean execution. Second, we measure action disruption using action distance, action-channel statistics, and horizon-level action shifts. Third, for imagination-preserving attacks, we measure stealthiness using predicted-future distance and adversarial score. Together, these metrics separate two questions: whether the robot fails, and whether the failure is visible through the WAM’s imagined future.

5 Evaluation

We organize the evaluation around six research questions:

Table 1: Main closed-loop attack results on full sweeps with $\epsilon = 0.06$ and a search budget of 8 optimization iterations per replanning step. Gray values indicate the absolute reduction from clean task success.

Model	LIBERO (%)			RobotWin (%)		
	Clean	Action-only	Img-pres.	Clean	Action-only	Img-pres.
Action-only WAM	96.5	43.1 ($\downarrow 53.4$)	—	92.1	84.4 ($\downarrow 7.7$)	—
Joint WAM	98.1	61.5 ($\downarrow 36.6$)	63.0 ($\downarrow 35.1$)	90.9	84.4 ($\downarrow 6.5$)	85.2 ($\downarrow 5.7$)
IDM WAM	98.4	66.1 ($\downarrow 32.3$)	68.1 ($\downarrow 30.3$)	91.4	83.7 ($\downarrow 7.7$)	85.1 ($\downarrow 6.3$)

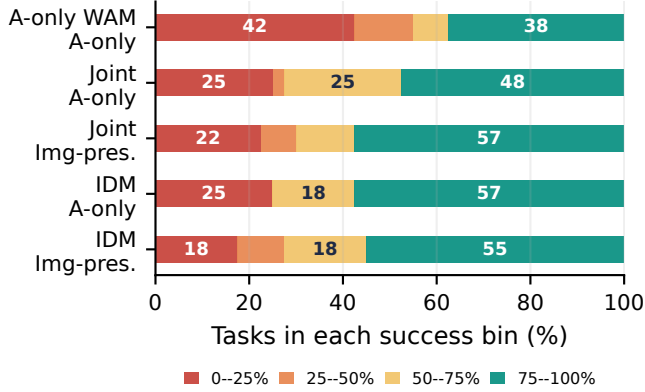


Figure 5: Task-level failure profile under attack. Unlike Table 1, which reports aggregate success, this figure shows how attacked tasks distribute across success-rate bins. Lower bins indicate tasks that are consistently broken rather than merely slightly degraded.

- RQ1 Attack effectiveness.** How much can bounded observation perturbations reduce closed-loop WAM reliability?
- RQ2 Systematicity.** Are the failures concentrated in a few tasks, or do they persist across suites and repeated trials?
- RQ3 Failure mechanism.** What do BadWAM failures look like during execution, and how does the query-based search produce them?
- RQ4 Stealth under matched strength.** Under matched attack strength, what does imagination preservation buy?
- RQ5 Sensitivity and efficiency.** How do future preservation, perturbation size, and query budget trade off?
- RQ6 Transferability and defenses.** Do the perturbations transfer across WAM variants, and how far do simple non-adaptive defenses go?

5.1 Experimental Setup

Benchmarks. Our main experiments use two language-conditioned robot manipulation benchmarks: LIBERO [16] and RoboTwin [6]. LIBERO provides four suites-Spatial, Object, Goal, and Long-horizon-for closed-loop tabletop manipulation. We evaluate all four LIBERO suites. RoboTwin

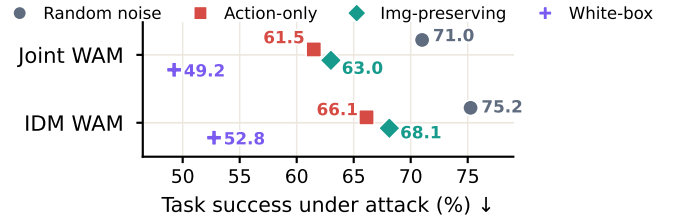


Figure 6: Comparison with random perturbations on full LIBERO dataset. All methods use the same ℓ_∞ budget. Lower task success indicates a stronger attack.

provides a larger-scale dual-arm manipulation benchmark with diverse long-horizon tasks and richer visual observations. Unless otherwise stated, each LIBERO main-result cell uses the full LIBERO sweep with 20 trials per task, and each RoboTwin main-result cell uses the corresponding RoboTwin full evaluation protocol. For more expensive studies, such as transfer, defenses, and ablations, we use a balanced LIBERO subset with three tasks per suite and 10 trials per task.

WAM policies. We evaluate three WAM-style controllers that expose different interfaces [22, 27]. The action-only WAM maps the current observation and instruction directly to an action sequence. The joint WAM predicts future visual states and actions jointly. The IDM WAM first constructs a future-imagination representation and then decodes actions from it. This model set separates action-only policies from policies whose imagined futures are also query-visible. All evaluations are closed-loop: the policy is repeatedly queried during execution, and BadWAM may perturb the current observation at each replan. For the action-only variant on LIBERO and RobotWin, we use the officially released checkpoints. For the other two LIBERO variants, we train them on 8 H100 GPUs with the default settings [27]. For the RobotWin variants, we train them on 8 H100 GPUs for 50k steps.

Attacks. We evaluate two objectives. The *action-only adversarial attack* maximizes the shift in the predicted action sequence under an ℓ_∞ -bounded observation perturbation. It requires only action outputs and therefore applies to all three policies. The *imagination-preserving adversarial attack* additionally penalizes changes in the predicted future, so it applies to WAMs whose future predictions are query-visible. Unless otherwise stated, both attacks perturb the full image with $\epsilon = 0.06$. The default search budget is 8 optimization iterations per replan; with the clean reference query, this gives 17

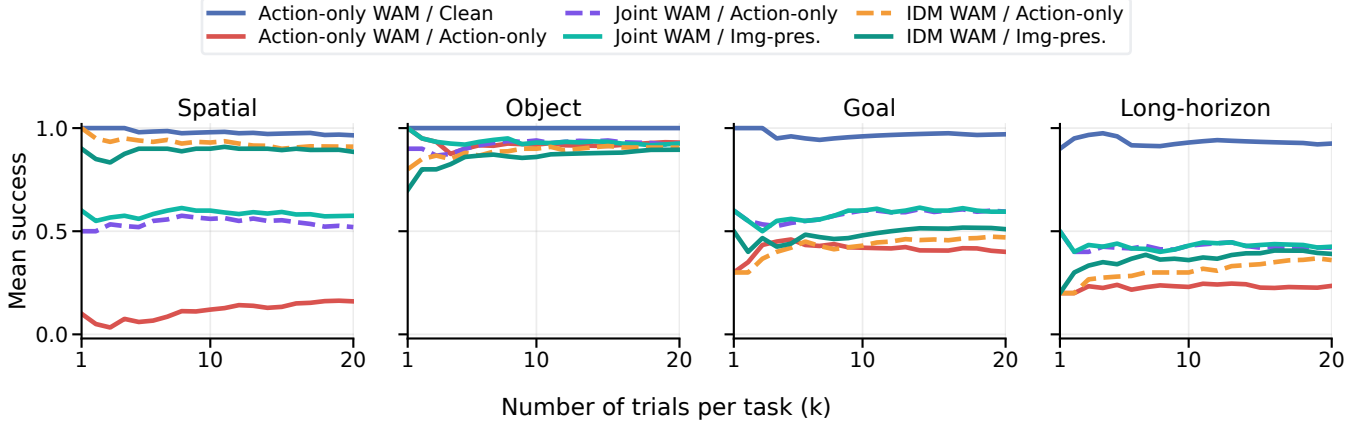


Figure 7: Mean pass@ k across trials on different LIBERO suites. BadWAM lowers success throughout the trial budget instead of only causing isolated unlucky failures.

WAM forward queries per attacked replan.

Metrics. Our primary metric is closed-loop task success rate; lower success under attack indicates a stronger attack. For a benchmark with T tasks and N_i evaluation episodes for task i , we compute

$$\text{Success} = \frac{1}{T} \sum_{i=1}^T \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbb{I}[s_{i,j} = 1], \quad (13)$$

where $s_{i,j}$ indicates whether episode j of task i succeeds. We also report the induced success drop. For mechanism analysis, we use action distance D_{act} , predicted-future distance D_{img} , and the normalized decoupling score from Section 4. For repeated-trial analysis, we report pass@ k : for each task, pass@ k is the fraction of successful executions among its first k trials, averaged across tasks.

5.2 BadWAM Reliably Induces Task Failures

To answer **RQ1**, Table 1 reports the main closed-loop results on LIBERO and RoboTwin. Clean WAM policies are strong. For example, in LIBERO, the action-only WAM reaches 96.5% task success, while the joint and IDM WAMs reach 96.7% and 100.0%. BadWAM substantially reduces these rates. On the action-only WAM, the action-only attack lowers success to 43.1%, a 53.4% drop. On the joint WAM, action-only attacks lowers success to 61.5%, while imagination-preserving attacks lowers success to 63.0%. On the IDM WAM, the corresponding attacked success rates are 66.1% and 67.0%. The main message is that WAMs are not protected simply because action generation is coupled with future prediction. BadWAM causes high-performing systems to fail under small, bounded perturbations. Moreover, the imagination-preserving attack remains close to the action-only attack even though it explicitly suppresses predicted-future drift. This indicates that the imagination pathway can be preserved while the action pathway is still shifted toward task failure.

Action-only Clean	96	100	97	92
Action-only Atk.	16	93	40	24
Joint A-only	52	92	60	42
Joint Img-pres.	57	92	60	42
IDM A-only	91	90	47	36
IDM Img-pres.	88	90	51	39

Spatial Object Goal Long-horizon

Success (%)

Figure 8: Per-suite success rates on LIBERO. The attack is especially damaging on spatial and long-horizon tasks, while object-centric tasks remain comparatively more robust.

Figure 5 complements the aggregate table with a task-level failure profile. For the action-only WAM, 42.5% of tasks fall into the 0-25% success bin under attack. For the joint and IDM WAMs, 17.5-25.0% of tasks fall into this lowest-success bin, depending on the objective. This distributional view is important for safety: a moderate average can hide tasks that become almost unusable.

Sanity check against white-box attacks and random perturbations. To verify that BadWAM is not merely exploiting generic visual fragility, we compare it with two references in Figure 6. First, we use a model-agnostic random uniform perturbation baseline under the same ℓ_∞ budget. Second, we include a stronger white-box reference that backpropagates through the WAM to optimize an ℓ_∞ -bounded input perturbation with a differentiable action-output surrogate; after the perturbation is found, the model is still evaluated with the standard closed-loop inference procedure. This white-box setting is outside BadWAM’s black-box threat model, but serves as an upper-bound reference for how much stronger

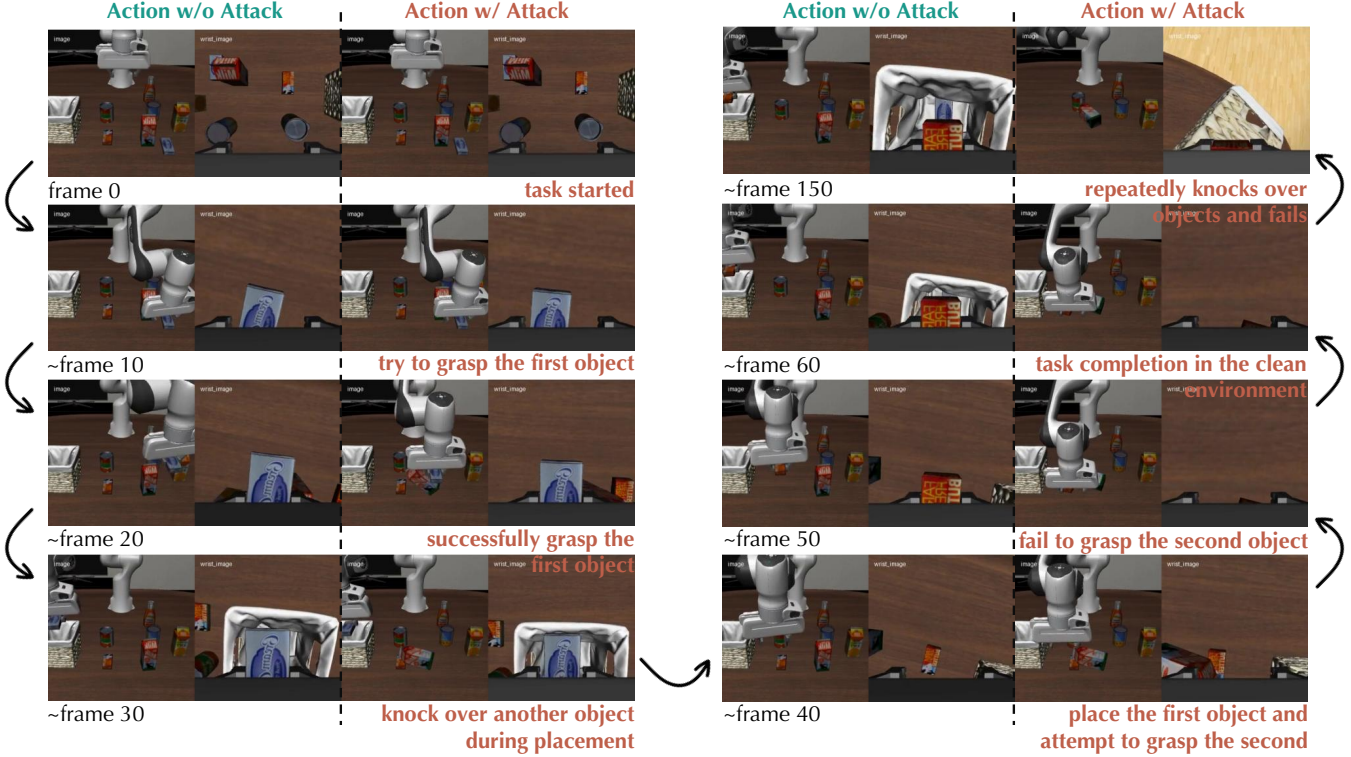


Figure 9: Qualitative comparison of clean and attacked rollouts on a LIBERO-10 task. Each row shows synchronized third-person and wrist-camera observations over time. The clean WAM completes the sequential manipulation task, while the attacked WAM initially behaves plausibly but gradually drifts, knocks over objects, fails to grasp the second object, and eventually fails.

attacks can become with gradient access.

Random perturbations are much weaker than targeted attacks: on Joint WAM and IDM WAM, random noise leaves task success at 71.0% and 75.2%, while BadWAM reduces success to 61.5%/63.0% and 66.1%/68.1% for action-only and imagination-preserving objectives, respectively. The white-box reference further lowers success to 49.2% and 52.8%, showing that stronger access can amplify the vulnerability. Together, these results suggest that BadWAM failures are caused by targeted world-action attacks rather than arbitrary image corruption, and that our black-box attacks leave room below the white-box upper bound.

Takeaway

BadWAM substantially reduces closed-loop success across WAM variants. The effect persists for imagination-based WAMs, showing that future prediction is not automatically a robustness mechanism.

5.3 Failures Are Systematic Across Suites and Trials

To answer **RQ2**, we next test whether the attack is concentrated in a few task types or persists across the benchmark. Figure 8 breaks success down by LIBERO suite. The largest

drops occur on tasks requiring precise spatial control or long-horizon execution. For the action-only WAM, Spatial success falls from 96.5% to 16.0%, Goal success falls from 97.0% to 40.0%, and Long-horizon success falls from 92.5% to 23.5%. Object tasks remain comparatively robust at 93.0%. The joint and IDM WAMs show the same qualitative pattern, although the degradation is less extreme. For example, the joint WAM under action-only attacks reaches 52.0% success on Spatial and 42.0% on Long-horizon, but remains at 92.5% on Object.

This suite-level structure suggests that BadWAM is not simply destroying visual perception uniformly. It is most effective when small action deviations can compound through geometry, contact timing, or sequential dependencies. Object-centric tasks often tolerate moderate drift as long as the relevant affordance remains visible and reachable. Spatial and long-horizon tasks leave less margin for a wrong approach direction, grasp position, or gripper timing.

We also test whether the success drop survives repeated trials. Figure 7 plots $\text{pass}@k$ as k grows from 1 to 20. If BadWAM only exploited a few unlucky seeds, the attacked curves would move back toward the clean curves as more trials are averaged. Instead, the separation persists across the full trial budget. The clean action-only WAM stays close to 1.0, while the attacked action-only WAM remains around 0.40-0.43. The joint and IDM WAMs also remain consistently below their clean references under both attack objectives.

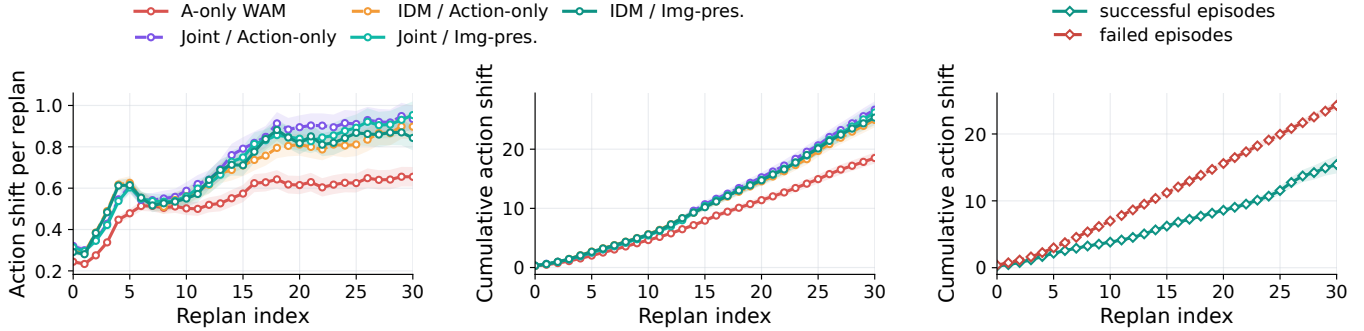


Figure 10: Per-replan action shifts persist across execution and accumulate over time, with failed episodes showing substantially larger cumulative shifts than successful ones. Action shift is measured between clean and attacked action chunks at each replan; shaded regions denote 95% confidence intervals.

Takeaway

BadWAM induces systematic reliability loss. The failures are strongest on tasks where small action errors accumulate over space and time, and they persist under repeated-trial evaluation.

5.4 How Do BadWAM Failures Emerge?

We next answer **RQ3** by examining representative executions and the optimization traces that produce them. Figure 9 shows a representative failure. The clean WAM completes the sequential manipulation task by grasping and placing the required objects in order. The attacked WAM, however, does not immediately collapse into random motion. It initially follows a plausible strategy and interacts with the right region, but its actions gradually drift away from the task intent. As the rollout progresses, the robot knocks over nearby objects, fails to stably grasp the second object, and eventually fails.

This example highlights why closed-loop evaluation is necessary. The perturbation need not make the observation obviously nonsensical or cause an obviously catastrophic first action. Instead, it can introduce structured deviations that remain locally plausible but become physically consequential over the horizon. Figure 10 confirms this behavior at the dataset level: per-replan action shifts persist across execution and accumulate over time. Failed episodes accumulate substantially larger action shifts than successful ones, suggesting that BadWAM failures often emerge from progressive closed-loop drift rather than a single isolated bad action.

Figure 11 analyzes the search process behind these failures. Across query-optimization iterations, the best objective score increases and the best action distance increases with it. The future-video distance changes more mildly relative to the action shift. Thus, the attack is not simply random visual noise; across replans, the optimizer repeatedly finds bounded perturbations that move the action output while keeping future-prediction drift under pressure from the preservation term.

We then answer **RQ4** by comparing the two objectives

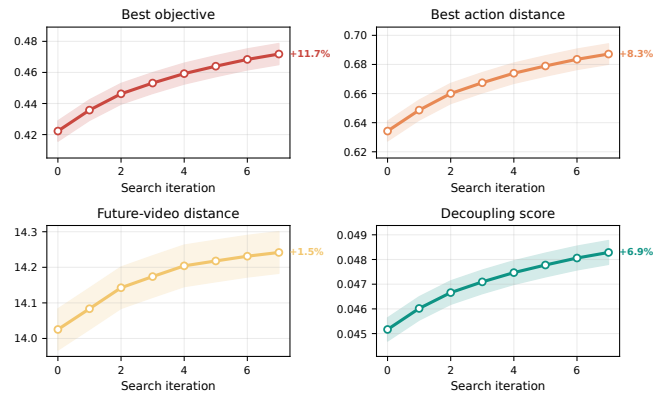


Figure 11: Search dynamics for the imagination-preserving attack. Each panel uses a metric-specific y-axis range and reports mean $\pm$ 95% confidence interval across replans. The query-based optimizer consistently improves the objective and increases action deviation, while the future-video distance changes much less in relative terms.

under matched perturbation and query budgets. Figure 12 isolates this trade-off under matched attack resources. The two objectives use the same perturbation bound, query budget, and closed-loop protocol. Their under-attack success rates are close, but the imagination-preserving objective reduces average predicted-future distance from 14.01 to 13.04 while keeping the input perturbation budget essentially matched. Across the 40 LIBERO tasks in this run, its predicted-future distance is lower on 39 tasks. This supports the core BadWAM claim: a WAM can be pushed toward failure while its imagined future remains closer to the clean prediction than a purely action-only objective would allow.

These dynamics are also useful diagnostically. If action distance rises together with a large future-distance increase, the attack becomes less stealthy. If future distance stays low but action distance does not improve, the objective is over-regularized. The default configuration lies between these regimes: it induces a damaging action shift while retaining a comparatively stable predicted future.

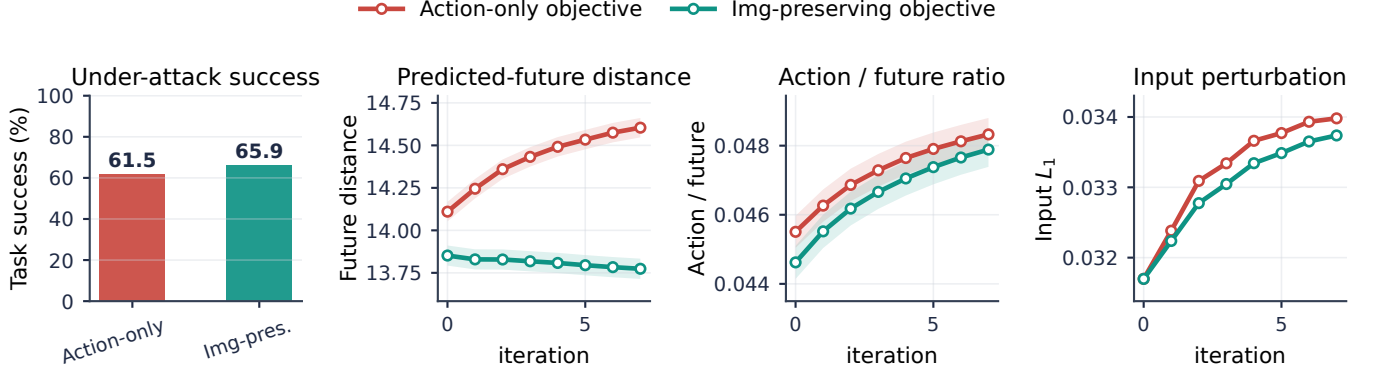


Figure 12: Matched-strength stealth trade-off. The imagination-preserving objective produces consistently smaller predicted-future shifts under a comparable input perturbation budget, revealing a stealthier failure mode than the action-only objective. Shaded regions denote 95% confidence intervals over replans.

Takeaway

BadWAM failures are structured and progressive. The attack consistently moves the action output in a damaging direction, rather than simply generic visual noise.

5.5 Ablations: Future Preservation, Perturbation Budget, and Runtime

To answer **RQ5**, we ablate the future-preserving weight, perturbation budget, and query budget. Figure 13 varies the future-preserving weight λ in the imagination-preserving objective. When $\lambda = 0$, the optimizer does not penalize predicted-future changes. As λ increases, future distance generally decreases, showing that the preservation term actively shapes the search rather than merely changing the reported metric. For the joint WAM, increasing λ from 0 to 0.015 reduces D_{img} from 14.70 to 14.34 and lowers task success from 61.7% to 56.7%. For the IDM WAM, the same setting reduces D_{img} from 15.36 to 15.13 and lowers success from 55.0% to 51.7%. At larger λ , future distance continues to trend downward, but action distance and attack strength begin to weaken. Thus, λ controls a real action-imagination trade-off.

Figure 14 studies the sensitivity of the imagination-preserving attack to perturbation size and query budget. The perturbing budget has the clearest effect. With $\epsilon = 0.01$, task success remains high for both models, at 95.8% for the joint WAM and 98.3% for the IDM WAM. At the default $\epsilon = 0.06$, success drops to 56.7% and 51.7% on the balanced subset. At $\epsilon = 0.20$, both models collapse to 0.0% success. Attack strength therefore scales strongly with the allowed visual perturbation budget.

On the other hand, the query-budget sweep exposes a non-monotonic runtime-effectiveness frontier. Moving from budget 1 to 32 increases mean per-replan optimization time from 2.54s to 27.84s on the joint WAM and from 2.71s to 30.57s on the IDM WAM. Increasing the budget from 1 to 16 im-

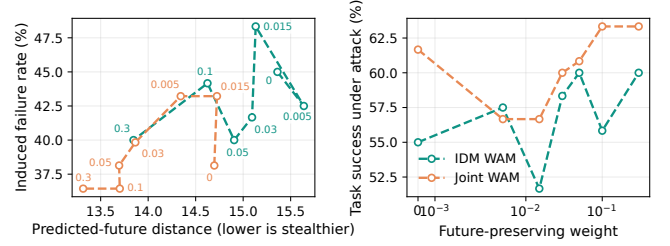


Figure 13: Ablation on future-preserving weight λ . Moderate future preservation can improve the action-future tradeoff, while excessive preservation weakens action manipulation.

proves attack effectiveness, reducing task success from 64.2% to 56.7% for the joint WAM and from 60.8% to 51.7% for the IDM WAM. However, budget 32 does not further reduce closed-loop success despite its higher per-replan optimization cost. This reflects the gap between the local per-replan objective optimized by BadWAM and the long-horizon closed-loop task outcome: larger local action shifts do not always translate into more task failures after environment feedback and subsequent replanning. We therefore use a moderate default budget, which captures most of the attack benefit while avoiding unnecessary runtime overhead. These measurements come from an analysis-oriented implementation that saves extensive statistics, so they should be read as prototype costs rather than optimized real-time numbers.

Takeaway

BadWAM exposes a tunable tradeoff. Larger perturbations increase attack strength, λ controls future preservation, and query budget controls the strength-runtime frontier.

5.6 Transferability Across WAM Variants

Here, we first answer the transfer part of **RQ6** by testing whether BadWAM perturbations are tied to a single source

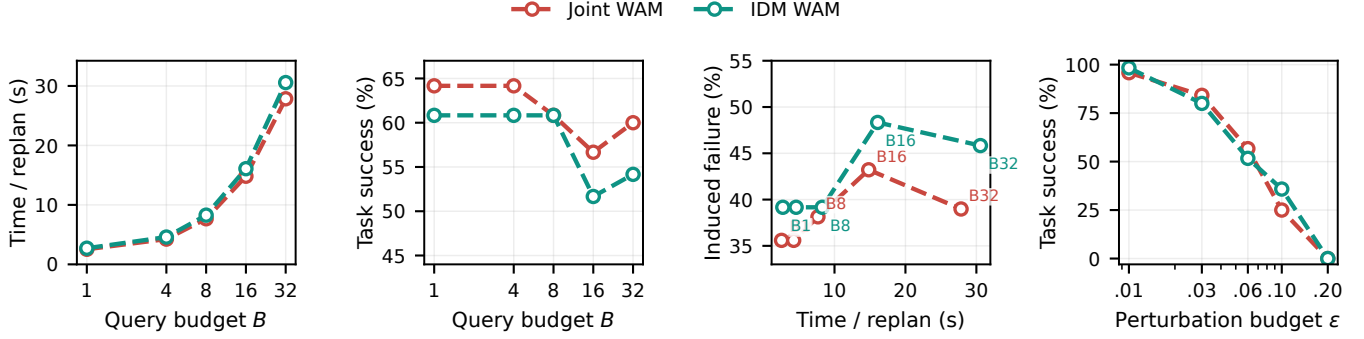


Figure 14: Efficiency and budget sensitivity of BadWAM. Increasing the query budget B raises the per-replan optimization cost, but also gives the optimizer more opportunity to find stronger perturbations. The perturbation-budget study further shows that larger ϵ consistently improves attack effectiveness. Together, the curves expose the practical tradeoff among attack strength, stealthiness, and runtime.

model. The defense part of **RQ6** is answered in Section 5.7. Table 2 optimizes perturbations on one WAM and evaluates the attacked closed-loop rollout on a different target WAM. The transferred attacks remain effective. For action-only attacks, transferred task success falls to 64.2% for Action-only WAM $\rightarrow$ Joint WAM, 59.2% for Joint WAM $\rightarrow$ IDM WAM, and 61.7% for IDM WAM $\rightarrow$ Joint WAM. These are 32.5%-40.8% drops relative to the target clean references.

Imagination-preserving attacks also transfer between the joint and IDM WAMs. Joint WAM $\rightarrow$ IDM WAM transfer lowers target success from 100.0% to 60.8%, while IDM WAM $\rightarrow$ Joint WAM transfer lowers target success from 96.7% to 63.3%. The source-side future distances remain moderate, with $D_{\text{img}}^{\text{src}} = 14.86$ and 14.32. Thus, BadWAM does not simply overfit to one action head. It exploits observation-space directions that affect action generation across related WAM variants.

Takeaway

BadWAM perturbations are not model-specific artifacts. They transfer across WAM variants, suggesting shared action-sensitivity directions in the observation space.

5.7 Non-Adaptive Defenses and Detection

We then answer the defense part of **RQ6** with simple non-adaptive preprocessing and detection baselines. Table 3 reports preprocessing defenses against the imagination-preserving attack on the balanced LIBERO subset mentioned in Section 5.1. We test three lightweight transformations: Gaussian blur smooths the input image before querying the WAM, JPEG-noise ensemble averages predictions over JPEG-compressed/noisy views, and resize-crop ensemble averages predictions over random resized crops. Gaussian blur and JPEG-noise ensembles recover much of the attacked success while keeping clean performance high. For example, JPEG-noise raises attacked success to 89.2% on the joint WAM and

90.0% on the IDM WAM. By contrast, resize-crop ensembles are not operationally useful: they reduce clean success to nearly 50% and still leave the attacked policies mostly unusable.

These results should not be read as complete defenses. They mainly show that the current attack is non-adaptive to these transformations; an adaptive attacker could optimize through the same preprocessing or use expectation over transformations. We also evaluate an augmentation-consistency detector, which flags replans whose outputs are inconsistent under a JPEG-augmented view. Figure 15 shows moderate AUROC, but at 5% false-positive rate it detects only 13.4% of attacked joint-WAM replans and 21.4% of attacked IDM-WAM replans.

Takeaway

Simple preprocessing can weaken the current non-adaptive attack, but it is not a complete defense and may harm clean performance. Simple consistency detection is also insufficient at practical false-positive rates.

5.8 What Do These Results Imply for WAM Safety?

Across **RQ1-RQ6**, the results point to a subtle failure mode for imagination-based safety checks. A natural WAM monitor might inspect predicted futures and accept an action if the future looks plausible. BadWAM shows that this criterion is incomplete. The relevant security property is not plausibility of the imagined future in isolation, but synchronization between the imagined future and the action that will actually be executed. In the matched-strength run, the imagination-preserving objective lowers predicted-future drift on 39 out of 40 LIBERO tasks while still inducing substantial closed-loop failures. A monitor that only scores visual future plausibility could therefore be satisfied even after the action channel has

Table 2: Transferability of BadWAM attacks across WAM variants on the LIBERO subset. The attacker optimizes the perturbation on the source WAM and evaluates the closed-loop task success on the target WAM. Lower transferred task success indicates stronger transfer. $D_{\text{img}}^{\text{src}}$ reports the mean source-side imagination distance for imagination-preserving attacks. † denotes matched/subset clean references.

Attack objective	Transfer direction	Target clean (%)	Target under attack (%)	Drop (pp)	$D_{\text{img}}^{\text{src}}$
Action-only	Action-only WAM → Joint WAM	98.3 [†]	64.2	34.2	–
Action-only	Joint WAM → IDM WAM	100.0 [†]	59.2	40.8	–
Action-only	IDM WAM → Joint WAM	98.3 [†]	61.7	36.7	–
Img-pres.	Joint WAM → IDM WAM	100.0 [†]	60.8	39.2	14.86
Img-pres.	IDM WAM → Joint WAM	98.3 [†]	63.3	35.0	14.32

Table 3: Results of some non-adaptive defense baselines on LIBERO. Higher attack success indicates better robustness against the imagination-preserving attack. † denotes matched/subset clean references.

Model	Defense	Clean success (%)	Attack success (%)
Joint WAM	None	98.3 [†]	57.1
Joint WAM	Gaussian blur	98.3 [†]	85.0
Joint WAM	Resize-crop ens.	52.5 [†]	0.0
Joint WAM	JPEG-noise ens.	94.2 [†]	89.2
IDM WAM	None	100.0 [†]	54.6
IDM WAM	Gaussian blur	98.3 [†]	89.2
IDM WAM	Resize-crop ens.	54.2 [†]	8.3
IDM WAM	JPEG-noise ens.	93.3 [†]	90.0

shifted toward failure.

This also explains why the attack is WAM-specific. For a conventional action-only policy, an adversarial perturbation can only be assessed through its effect on the action or final task outcome. For a WAM, the attacker can exploit a gap between two interfaces: the future-imagination interface and the action interface. The action-only BadWAM objective targets the high-strength endpoint of this spectrum; the imagination-preserving objective targets a stealthier endpoint. The latter is especially relevant for systems that expose predicted futures to planners, runtime monitors, or human supervisors. Even when the predicted future remains visually close to the clean prediction, the executed action chunk may no longer be the action that would realize that future.

The defense experiments reinforce the same lesson. Pre-processing can be a useful diagnostic, but it is not a principled solution to WAM safety. A practical defense must preserve clean control while checking whether action and imagination remain mutually consistent over the closed-loop trajectory. The low-recall detector in Figure 15 further shows that moderate AUROC is insufficient for robot safety: runtime monitors must operate at very low false-positive rates, where recall is hardest to maintain.

Finally, the transfer results suggest that the vulnerability is not merely an implementation artifact. Perturbations optimized on one WAM variant remain damaging on another,

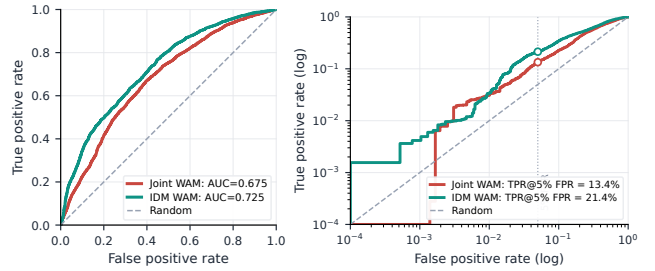


Figure 15: Augmentation-consistency detection is insufficient against imagination-preserving attacks.

indicating shared observation-space directions that affect action generation. A WAM safety evaluation should therefore go beyond clean and attacked success rates. It should also report action distance, predicted-future distance, input perceptibility, horizon-level shifts, and failure distributions across task families. Together, these metrics distinguish overt action hijacking from stealthier imagination-preserving failures, and test whether a defense restores action-imagination synchronization rather than masking one perturbation pattern.

5.9 Summary of Findings

The evaluation supports six conclusions. First, BadWAM substantially lowers closed-loop success across WAM variants, with around 30%-50% drops. Second, the failures are systematic: they persist across repeated trials and are strongest on spatial and long-horizon tasks. Third, the attack induces structured closed-loop failures rather than immediate random behavior. Fourth, imagination preservation creates a measurable stealth trade-off: it keeps predicted futures closer to clean rollouts while retaining comparable perturbation budgets and substantial attack strength. Fifth, BadWAM exposes a tunable sensitivity frontier: attack strength depends strongly on perturbation size, future-preserving weight, and query budget. Sixth, the vulnerability is not isolated to one model: attacks transfer across WAM variants, while simple detection baselines miss most attacks at practical false-positive rates.

Together, these results show that WAM safety cannot be

reduced to asking whether the imagined future looks plausible. The more security-relevant question is whether the model’s imagined future and selected action remain synchronized under adversarial observation perturbations.

6 Conclusion

World-action models promise a tighter connection between perception, prediction, and control by coupling action generation with imagined futures. This paper shows that this coupling can itself become an attack surface. We introduce BadWAM, a black-box framework for world-action drift attacks that induces task failures through small observation perturbations. Beyond action-only attacks, BadWAM exposes a more WAM-specific failure mode: action-imagination decoupling, where the executed actions are shifted while its predicted future remains comparatively stable. Across multiple WAM variants, BadWAM substantially reduces task success under bounded perturbations. Our analyses show that these failures are not captured by any single metric: task success, action shifts, predicted-future drift, perturbation budget, horizon structure, and transfer behavior each reveal different aspects of the vulnerability. The matched-strength comparison further shows that imagination-preserving attacks can maintain similar attack strength while reducing future drift, suggesting that future-based monitors alone may be overconfident. These results highlight the need to evaluate WAM safety as a synchronization problem between what a model imagines and what it executes. We hope BadWAM provides both a concrete attack framework and a diagnostic protocol for building safer world-action models.

References

- [1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. *pi_0*: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [2] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.
- [3] Linghan Chen, Kaiyan Ji, and Minyu Guo. Attacking the trusted imagination: Oracle-level integrity attacks on imagine-then-act world models. *arXiv preprint arXiv:2606.22966*, 2026.
- [4] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017.
- [5] Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Dekang Qi, et al. Abot-m0.5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026.
- [6] Tianxing Chen, Zanzin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.
- [7] Kevin Eykholt, Ivan Evtimov, Earlene Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.
- [8] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*, 2020.
- [9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.
- [10] Zhixiang Guo, Siyuan Liang, András Balogh, Noah Lunberry, Rong-Cheng Tu, Márk Jelasity, and Dacheng Tao. When world models dream wrong: Physical-conditioned adversarial attacks against world models. *arXiv preprint arXiv:2602.18739*, 2026.
- [11] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2137–2146. PMLR, 2018.
- [12] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.
- [13] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning*, pages 2679–2713. PMLR, 2025.

- [14] Qi Li, Bo Yin, Weiqi Huang, Ruhao Liu, Bojun Zou, Runpeng Yu, Jingwen Ye, Weihao Yu, and Xinchao Wang. Vision-language-action safety: Threats, challenges, evaluations, and mechanisms. *arXiv preprint arXiv:2604.23775*, 2026.
- [15] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3756–3762, 2017.
- [16] 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*, volume 36, 2023.
- [17] Hanqing Liu, Songping Wang, Jiahuan Long, Jiacheng Hou, Jialiang Sun, Chao Li, Yang Yang, Wei Peng, Xu Liu, Tingsong Jiang, et al. Jailwam: Jailbreaking world action models in robot control. *arXiv preprint arXiv:2604.05498*, 2026.
- [18] Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiaoping Zhang, and Wenbo Ding. Oa-wam: Object-addressable world action model for robust robot manipulation. *arXiv preprint arXiv:2605.06481*, 2026.
- [19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.
- [20] Utkarsh A. Mishra, Yongxin Chen, Danfei Xu, Yang Liu, Xi Chen, and Jiayuan Mao. Understanding and mitigating the video-action generalization gap via temporal ratio. *arXiv preprint arXiv:2607.08127*, 2026.
- [21] Linghui Shen, Mingyue Cui, and Xingyi Yang. Bad-world: Adversarial attacks on world models. *arXiv preprint arXiv:2606.16519*, 2026.
- [22] Qiuhong Shen, Shihua Zhang, Yue Liao, Qi Li, Zhenxiong Tan, Shizun Wang, Shuicheng Yan, and Xinchao Wang. World action models: A survey. *arXiv preprint arXiv:2606.20781*, 2026.
- [23] James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.
- [24] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.
- [25] Shuai Tian, Yupeng Zheng, Yuhang Zheng, Songen Gu, Yujie Zang, Yuxing Qin, Weize Li, Haoran Li, Wenchao Ding, and Dongbin Zhao. Vt-wam: Visual-tactile world action model for contact-rich manipulation. *arXiv preprint arXiv:2607.02503*, 2026.
- [26] Jonathan Uesato, Brendan O’Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5025–5034. PMLR, 2018.
- [27] 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.
- [28] Wentao Zhao, Jiaming Chen, Ziyu Meng, DonghuiMao, Ran Song, and Wei Zhang. VLMPC: Vision-Language Model Predictive Control for Robotic Manipulation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.
- [29] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

Appendix

A Details of Experiment Setup

A.1 Full and Subset Evaluation Protocols

Table 4 summarizes which results are currently reported on the full benchmark sweep and which results use the balanced LIBERO subset. The full LIBERO sweep contains all four suites—Spatial, Object, Goal, and Long-horizon—with 10 tasks per suite and 20 trials per task, resulting in 800 closed-loop episodes per model–attack pair. For expensive diagnostic studies, we use a balanced subset with three tasks per suite (task IDs 0, 4, and 9 in each suite) and 10 trials per task, resulting in 120 episodes per model–attack pair. We use the subset only for analysis-oriented experiments whose purpose is to compare trends under controlled settings. We plan to rerun these subset studies on the full LIBERO sweep and update the corresponding numbers in the final version.

The subset protocol is designed to preserve suite diversity while keeping expensive studies tractable. In particular, each subset experiment contains representatives from spatial reasoning, object manipulation, goal-conditioned manipulation, and long-horizon manipulation. We keep the same subset across future-preserving-weight ablations, perturbation-budget ablations, query-budget ablations, transfer experiments, and defense baselines, so that comparisons within these analyses are matched by task distribution.

A.2 Training Details

For the action-only WAM, we use the official FastWAM checkpoints [27]. The joint WAM and IDM WAM variants used in our LIBERO experiments are trained from the FastWAM release using the default training configuration. For RoboTwin, we use the same FastWAM training recipe and train the joint and IDM variants for 50,000 optimization steps on 8 H100 GPUs.

Table 5 lists the main training settings. For LIBERO, the model uses two cameras, namely the third-person view and the wrist view. Each camera is resized to 224×224 , and the two camera streams are concatenated horizontally. For RoboTwin, the model uses three cameras: a high camera and two wrist cameras. Each camera is resized to 240×320 , and the resulting video representation follows the FastWAM RoboTwin configuration. Both benchmarks use 33-frame training windows with an action-video frequency ratio of 4, corresponding to 32 action steps and 9 video frames.

The model backbone and tokenizer follow FastWAM. Specifically, the configuration uses Wan-AI/Wan2.2-TI2V-5B as the video backbone and Wan-AI/Wan2.1-T2V-1.3B as the tokenizer model. We keep the default FastWAM action and video schedulers, each with

1000 training timesteps and shift value 5.0. We do not modify the FastWAM architecture when training the joint and IDM variants.

A.3 Attack Hyperparameters and Query Accounting

Unless otherwise stated, BadWAM perturbs the full preprocessed visual input under an ℓ_∞ bound of $\epsilon = 0.06$. The perturbation is applied online at each replanning step during closed-loop execution. The default zeroth-order finite-difference search uses 16 perturbation queries plus one clean reference query per attacked replan, for a total of 17 WAM forward queries per attacked replan. Internally, the 16 perturbation queries are organized as 8 paired finite-difference updates. Each update samples one random Rademacher direction, evaluates positively and negatively perturbed inputs, and updates the perturbation by a projected signed step. We use a finite-difference radius of 0.02 and a search step size of 0.02.

For the action-only adversarial attack, the future-preserving coefficient is set to zero and the objective only maximizes clean–attacked action deviation. For the imagination-preserving adversarial attack, the objective additionally penalizes predicted-future drift. The default full-sweep setting uses $\lambda = 0.015$ for the future-preserving coefficient. The matched-strength analysis in Figure 12 uses a larger future-preserving coefficient when comparing the two objectives under similar attack strength, so that the effect of the preservation term is easier to isolate.

A.4 White-Box Gradient-Access Reference

Figure 6 includes a white-box gradient-access reference to contextualize the strength of our query-based attacks. This reference is not part of the main BadWAM threat model. BadWAM assumes query access and does not require gradients or model parameters. The white-box reference instead asks how much stronger the attack can become if the adversary can backpropagate through the WAM’s action output.

At the beginning of an episode, the attacker optimizes one bounded input perturbation using projected gradient ascent. The objective maximizes the deviation between the clean action chunk and the perturbed action chunk. The perturbation is initialized randomly inside the ℓ_∞ ball, updated for 16 gradient steps with step size 0.01, and projected back to $\|\delta\|_\infty \leq 0.06$ after each step. After the perturbation is found, the policy is evaluated with the standard closed-loop inference procedure. No model weights are modified, and the environment dynamics are not differentiated through.

This reference serves two purposes. First, it separates

Table 4: Evaluation scope of the reported experiments. Full LIBERO denotes 40 tasks with 20 trials per task. The balanced subset denotes 12 tasks with 10 trials per task.

Result	Current protocol	Planned update
Main closed-loop attack results in Table 1	Full LIBERO; RoboTwin full protocol with 50k-steps	RoboTwin on default 5 epochs (23.5k steps on 8× GPUs)
Random perturbation and white-box reference in Figure 6	Full LIBERO	No subset replacement needed
Pass@ k , suite-level analysis, and qualitative rollouts	Full LIBERO	No subset replacement needed
Closed-loop error accumulation and attack-search dynamics	Full LIBERO traces	No subset replacement needed
Matched-strength stealth trade-off	Full LIBERO for the selected WAM variant	Extend to additional variants
Future-preserving weight ablation in Figure 13	Balanced LIBERO subset	Rerun on full LIBERO
Perturbation-budget and query-budget ablations in Figure 14	Balanced LIBERO subset	Rerun on full LIBERO
Transferability in Table 2	Balanced LIBERO subset	Rerun on full LIBERO
Preprocessing defenses and consistency detection in Table 3	Balanced LIBERO subset	Rerun on full LIBERO

Table 5: Training configuration for the WAM variants trained in this work. We follow the FastWAM default configuration unless otherwise stated. Batch size is per GPU.

Setting	LIBERO joint/IDM WAMs	RoboTwin joint/IDM WAMs
Dataset configuration	libero_2cam	robotwin
Cameras	2 cameras	3 cameras
Image resize	224 × 224 per camera	240 × 320 per camera
Action dimension	7	14
State dimension	8	14
Training window	33 frames	33 frames
Action-video frequency ratio	4	4
Batch size per GPU	16	16
Number of GPUs	8 H100 GPUs	8 H100 GPUs
Optimizer schedule	cosine	cosine
Learning rate	1×10^{-4}	1×10^{-4}
Weight decay	1×10^{-2}	1×10^{-2}
Gradient accumulation	1	1
Number of workers	8	8
Training length	10 epochs	50k steps in this draft
Checkpoint interval	2000 steps	2500 steps
Evaluation interval	200 steps	500 steps

generic visual fragility from targeted action manipulation: random perturbations under the same budget are substantially weaker than targeted attacks. Second, it shows that gradient access can further amplify the vulnerability, which suggests that the query-based BadWAM attacks are not saturating the worst-case attack strength. Since this reference uses stronger access than BadWAM, we treat it as a diagnostic upper-bound reference rather than as the primary threat model.

A.5 Random Perturbation Baseline

The random perturbation baseline uses the same ℓ_∞ budget as BadWAM but removes the optimization objective. For each attacked observation, it samples an independent uniform perturbation from $[-\epsilon, \epsilon]$ and clips the perturbed input to the valid model input range. This baseline is model-agnostic and does not use action outputs, future predictions, gradients, or task feedback. Its role is to test whether task failures are caused merely by arbitrary visual corruption. The results in Figure 6 show that optimized world-action drift attacks are consistently stronger than random perturbations under the same visual budget.

A.6 Evaluation Metrics

Our primary metric is closed-loop task success. Lower success under attack indicates a stronger attack. In addition to success rate, we report action distance, predicted-future distance, and input perturbation statistics. The action distance is computed as the mean ℓ_2 deviation between clean and attacked action chunks. The predicted-future distance is computed analogously on the future representation or decoded future-video tensor available through the WAM interface. The decoupling score is the ratio between action distance and predicted-future distance, with a small numerical constant in the denominator for stability. A high decoupling score indicates that the attack obtains a larger action shift per unit of predicted-future drift.

For repeated-trial evaluation, we use pass@ k . For a fixed task, pass@ k is the fraction of successful executions among the first k trials. We then average this quantity across tasks. This metric shows whether an attack only causes isolated unlucky failures or consistently lowers reliability as more trials are considered.