PASA-KV: ATTACKING ACTION-FACING VISUAL ROUTES IN WORLD ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World action models (WAMs) increasingly couple internal visual or world representations with action generation, making their world-to-action pathways a critical yet poorly understood source of robustness. We investigate when perturbations to these internal representations actually become behaviorally consequential, and uncover a pronounced mismatch between representation perturbability and closed-loop vulnerability: attack effectiveness varies sharply across visual K/V routes, while larger representation displacement does not reliably imply stronger control failure. Our analysis attributes this mismatch to the action-conditioned readout, where keys determine where action queries read and values determine what content they receive. This motivates an action-facing view of vulnerability, in which the effect of a perturbation depends on how it is routed into action generation rather than on representation-space magnitude alone. We instantiate this view with Tail-AV, a short-probe signal for identifying vulnerable routes, and operationalize it in PASA-KV, which selects action-facing K/V targets before training a fresh task-shared attack. Experiments across multiple WAM systems and a physical robot validate readout-informed target selection, demonstrate strong closed-loop attacks, and reveal substantial system-level differences in resistance.

1 INTRODUCTION

World action models (WAMs) are emerging as a promising paradigm for robot control by coupling visual world representations and dynamics with action generation (Ha & Schmidhuber, 2018; Hafner et al., 2019; Li et al., 2026; Kim et al., 2026; Yuan et al., 2026); latent-imagination methods provide an earlier route from learned world models to control (Hafner et al., 2020; 2021). Generalist robot policies include RT-1, RT-2, and Octo (Brohan et al., 2023; Zitkovich et al., 2023; Octo Model Team et al., 2024), while MimicGen scales demonstration generation for robot learning (Mandlekar et al., 2023). Across direct, cascaded, and joint WAM architectures, action generation ultimately depends on internal visual or world representations. This coupling introduces a security question that cannot be answered by studying input perturbations alone: *when does a change in an internal visual representation actually propagate to the action generator and alter closed-loop behavior?* Existing attacks on robot policies expose vulnerabilities at the input, representation, attention, denoising, or action level (Chen et al., 2024; Wang et al., 2025; Lu et al., 2026; Tae & Lee, 2026; Zhang et al., 2026), but they do not reveal which world-to-action routes make WAMs behaviorally vulnerable.

A natural view is that routes whose representations can be perturbed more strongly should provide more effective attack targets. Our layerwise study shows that this view is incomplete. Under the same attack objective and training budget, perturbing different visual K/V layers of FastWAM leads to sharply different closed-loop outcomes (Figure 1(a,c)). More strikingly, stronger or broader internal perturbation does not necessarily produce stronger behavioral failure. Jointly attacking K and V broadens the set of effective targets, yet does not dominate either route at every layer; at L20, for example, K-only reduces policy success to 0.0%, whereas joint K/V leaves it at 91.5%. Across layers, the magnitude of K/V displacement likewise fails to determine eventual attack effectiveness. These results reveal a fundamental mismatch: **representation disruption is not action disruption**.

The missing information lies in how a perturbation interacts with the *action-conditioned readout*. In attention-based WAMs, visual keys determine where action queries read, whereas visual values determine what content they receive. A large K displacement can therefore have little effect when it

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

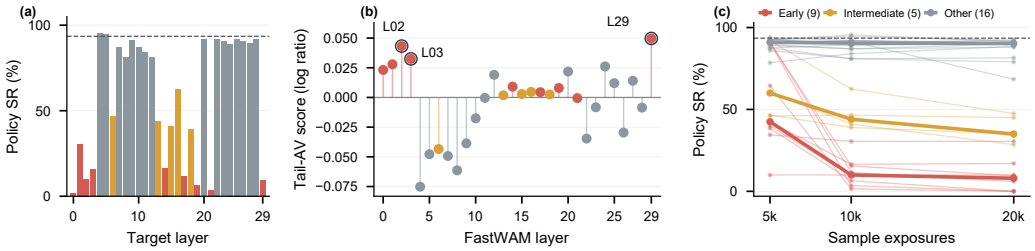


Figure 1: **Layerwise evidence for action-facing vulnerability.** All panels use the evaluated FastWAM checkpoint. **(a)** Policy SR for 30 singleton joint K/V targets after 10k exposures; dashed: clean reference. **(b)** Tail-AV scores from short probes; rings mark Top-3. **(c)** Budget–SR trajectories at 5k/10k/20k exposures. Colors denote retrospective response groups, unused by the selector.

poorly aligns with the directions to which action queries are sensitive, while a large V displacement may remain inconsequential when the altered content receives little attention. K- and V-induced changes can also reinforce or partially oppose one another, providing a mechanism for the non-monotonic behavior above. This suggests that vulnerability should not be characterized solely in representation space, but through the interaction between an internal perturbation and the action-facing pathway that consumes it. We refer to this view as *action-facing vulnerability*: the behavioral relevance of a perturbation depends not only on how much the representation changes, but also on how that change is routed into action generation.

This perspective changes how vulnerable routes should be identified. Rather than ranking layers by how far their representations move, we measure how a short perturbation changes the visual contribution received by action queries. This motivates Tail-AV, which combines action-query routing with the magnitude of visual content being read and, from short probes, identifies layers that later become strong attack targets (Figure 1(b)). Importantly, this signal is computed without closed-loop rollout outcomes. The resulting characterization naturally leads to the Probe-and-Select Action-facing K/V Attack (PASA-KV): probe candidate routes cheaply, select those with the strongest action-facing responses, and train the final task-shared patch only on the selected routes. In this way, PASA-KV operationalizes the action-facing view without exhaustively training and evaluating full attacks at every candidate layer.

Our experiments support this interpretation at both the mechanism and system levels. Under matched FastWAM controls with the same final K/V objective, target count, geometry, and training budget, Tail-AV-selected layers achieve 1.5% policy success, compared with 47.5% when layers are selected directly by probe K/V displacement and 18.5–86.5% for evenly spaced or random selections (Section 5.3). The resulting PASA-KV Top-5 attack reduces FastWAM success from 92.4% to 0.0%. Yet the same construction produces substantially smaller degradation on LingBot-VA and Cosmos Policy, reaching 87.2% and 89.2% SR, respectively, exposing pronounced architecture-dependent resistance rather than a universal vulnerability (Table 1). Finally, rerunning the complete procedure on a separately trained Piper FastWAM checkpoint yields a task-shared Top-5 patch with 0/10 successes on each of the three evaluated tasks with nonzero clean success (Section 5.6).

Our contributions are:

- We characterize visual K/V vulnerability in WAMs and reveal a systematic mismatch between representation perturbability and closed-loop behavioral vulnerability: attack effectiveness varies sharply across layers, and stronger internal perturbation does not necessarily imply stronger control failure.
- We identify action-facing readout as the missing link between internal perturbation and behavioral consequence. This view explains the distinct roles of K and V and motivates Tail-AV as a short-probe signal for locating vulnerable visual routes.
- We operationalize this understanding through PASA-KV, whose readout-informed selection substantially outperforms matched representation-based and uninformed controls. Cross-architecture and real-robot evaluations further reveal large differences in how WAMs expose and resist action-facing attacks.

2 Related Work

World Action Models and Their Architectural Paradigms. WAMs differ in how visual dynamics modeling enters action generation at inference time (Ye et al., 2026). Direct-policy FastWAM encodes observations into visual K/V that action denoising reads without generating future video (Yuan et al., 2026). In cascaded LingBot-VA, visual-state prediction precedes action denoising, which accesses video-side representations through cached K/V (Li et al., 2026). Joint Cosmos Policy instead represents actions and future states as latent slots within a shared denoising process, so visual–action interactions occur within the joint transformer rather than between sequential prediction stages (Kim et al., 2026).

Adversarial manipulation of visual attention and representations. Localized token and patch attacks can disrupt ViTs (Joshi et al., 2021; Fu et al., 2022), while patch-aware optimization improves attack transfer (Wei et al., 2022). At the attention level, adversarial patches can concentrate many queries on patch-controlled keys (Lovisotto et al., 2022). Related work studies visual jailbreaks (Qi et al., 2024), cross-attention attacks on customized diffusion models (Xu et al., 2024), attention backdoors (Lyu et al., 2022), and attention-head path-deviation diagnostics (Jeong et al., 2026). UPA-RFAS combines feature, attention, and semantic objectives (Lu et al., 2026); its attention objective targets text-to-vision aggregation, whereas PASA-KV targets visual keys and values consumed by action queries. AGSD studies physical action-to-vision attention hijacking, and SARF mitigates it through visual-encoder fine-tuning (Zhang et al., 2026).

Attacks on robot policies and vision–language–action models. OpenVLA provides a representative action-generating platform (Kim et al., 2024). DP-Attacker targets diffusion policies (Chen et al., 2024); Wang et al. (2025) study spatially grounded untargeted and targeted trajectory objectives; DRIFT attacks the first velocity-field step of flow-matching VLAs (Tae & Lee, 2026). Other work uses trajectory prefixes (Wang et al., 2026), targets CoT reasoning (Huang et al., 2026) or visual proprioception (Fu et al., 2026), steers visual-token attention (Prasad et al., 2026), or studies textual jailbreaks and training-time backdoors (Jones et al., 2025; Zhou et al., 2025).

3 From Representation Perturbability to Action Vulnerability

We use FastWAM as a controlled testbed to study how perturbations to internal visual representations become behaviorally consequential. We define *action-facing visual K/V* as the visual keys and values consumed by action queries, and optimize a single RGB patch against individual layers while keeping the victim and attack protocol fixed. This controlled setting lets us separate three quantities that are easily conflated: how much an internal representation can be changed, how much that change alters the action-facing readout, and how much the resulting policy behavior degrades. We first establish that vulnerability varies sharply across internal routes, then ask what actually explains this variation.

3.1 Behavioral vulnerability is highly route-dependent

Under the same joint K/V objective and training budget, different FastWAM layers exhibit sharply different closed-loop vulnerability (Figure 1(a,c)). For example, L03 drops from 91.0% Policy SR at 5k exposures to 15.5% and 9.5% at 10k and 20k exposures, whereas L28 remains close to clean performance over the same sweep. We retrospectively summarize these responses as Early, Intermediate, and Other layers only for analysis; they are not used for attack construction or selection.

This heterogeneity rules out a simple picture in which all internal visual routes become similarly vulnerable once they are exposed to the same attack objective. It instead suggests that some routes are intrinsically more consequential for control than others. The most direct explanation would be that these routes are simply easier to perturb in representation space: if so, the layers with larger K/V displacement should also induce stronger behavioral failures. We test this possibility next.

3.2 REPRESENTATION DISPLACEMENT DOES NOT EXPLAIN BEHAVIORAL VULNERABILITY

Representation-space change captures part of the attack response, but it does not explain the layerwise variation above. Across the 30 layers, normalized K and V displacement have only limited association with attack strength, $\rho(D_K, \mathrm{AttackStrength}) = 0.316$ and $\rho(D_V, \mathrm{AttackStrength}) = 0.528$. Layers with larger representation changes therefore do not reliably produce larger closed-loop failures. Complementary footprint and Rescue/Transfer analyses lead to the same conclusion and are reported in Appendix B.3.

The failure of this representation-space diagnostic changes the question. The issue is no longer simply *how much* a layer can be perturbed, but *which part of that perturbation is actually exposed to action generation*. A visual representation may move substantially while the action policy barely consumes the changed content; conversely, a smaller perturbation may matter if it lies directly on the route used by action queries. This motivates us to examine the interface through which visual K/V are read by the action generator.

3.3 VULNERABILITY EMERGES THROUGH THE ACTION-CONDITIONED READOUT

Consider how action query i reads visual token j:

$$A_{ij} = \mathrm{softmax}_j\left(\frac{Q_i K_j^\top}{\sqrt{d}}\right), \qquad y_i = \sum_j A_{ij} V_j. \tag{1}$$

Keys determine *where* action queries read, whereas values determine *what* content they receive (Vaswani et al., 2017). Prior work questions attention weights as standalone explanations (Jain & Wallace, 2019), while relevance-propagation methods extend analysis beyond attention visualization (Chefer et al., 2021); causal mediation and model editing provide complementary probes of internal computations (Vig et al., 2020; Meng et al., 2022). Representation displacement therefore becomes behaviorally relevant only through its interaction with this action-conditioned readout.

For $q_i = Q_{\mathrm{action}}$, a first-order expansion under $k_j \mapsto k_j + \delta k_j$ and $v_j \mapsto v_j + \delta v_j$ gives

$$\delta y_i \approx \underbrace{\frac{1}{\sqrt{d}} \sum_{j \in \mathcal{V}} a_{ij} (q_i^\top \delta k_j)(v_j - y_i)}_{u_K} + \underbrace{\sum_{j \in \mathcal{V}} a_{ij} \delta v_j}_{u_V}. \tag{2}$$

The two terms reveal why raw displacement alone is insufficient. A large K perturbation may induce little readout change when it poorly aligns with the action query, i.e., when $q_i^\top \delta k_j$ is small. Likewise, a large V perturbation may contribute little when the affected content receives negligible attention. The relevant quantity is therefore not representation-space magnitude alone, but how the perturbation is gated by action-query routing.

This view also explains why K and V need not combine monotonically. Since

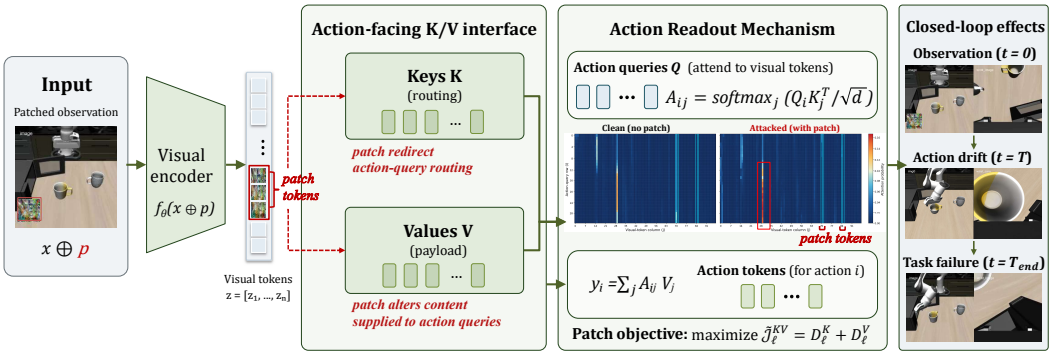
$$\|u_K + u_V\|_2^2 = \|u_K\|_2^2 + \|u_V\|_2^2 + 2\langle u_K, u_V \rangle,$$

K- and V-induced readout changes may reinforce or partially oppose one another. The controlled component attacks exhibit this behavior. Under the same FastWAM protocol and 10k-exposure budget, joint K/V optimization yields effective attacks ($\mathrm{SR} \leq 50\%$) on 13/30 layers, compared with 7/30 for K-only and 11/30 for V-only (Table 20; complete pooled results appear in Appendix Table 16). Joint optimization thus broadens effective-layer coverage overall, but does not dominate every layer: at L20, K-only reaches 0.0% SR whereas joint K/V remains at 91.5%.

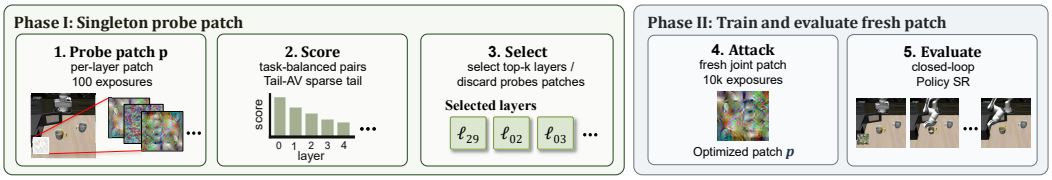
Together, these results suggest a more precise account of the heterogeneity observed in Section 3.1. Vulnerability is governed not simply by whether an internal representation can be moved, but by how that perturbation interacts with the action-conditioned pathway that consumes it. We refer to this distinction as *action-facing vulnerability*. Importantly, this is still a mechanistic account rather than a claim that readout change perfectly determines closed-loop failure. The stronger test is whether action-facing information can identify vulnerable routes *before* a full attack is trained.

3.4 CAN SHORT ACTION-FACING RESPONSES ANTICIPATE LATER VULNERABILITY?

The action-facing view makes a concrete prediction: if the readout captures information about vulnerability beyond raw representation displacement, then an early readout-aware response should



(a) PASA-KV attack surface: action-facing keys and values



(b) PASA-KV pipeline: locate, select, and train a fresh attack

Figure 2: **PASA-KV construction pipeline.** **(a)** Action-facing visual K/V interface. **(b)** Candidate routes are first probed with a small optimization budget and ranked by Tail-AV; a fresh task-shared patch is then trained only on the selected Top-k routes. Victim weights remain fixed throughout, and rollout outcomes are excluded from layer selection.

help distinguish routes that later become effective attack targets. We test this prediction with short probes rather than full attack training.

For visual token j, we define its action-facing contribution as

$$I_{ij} = A_{ij}\|V_j\|_2, \tag{3}$$

which combines whether the action query reads a visual token with the magnitude of the content delivered through that route. Tail-AV summarizes the probe-induced growth of the sparse upper tail of these contributions. Under the FastWAM protocol, Probe100 Tail-AV prioritizes L29/L02/L03, which later become strong attack targets under longer optimization (Figure 1(b)), without using rollout outcomes or retrospective response labels.

This prospective behavior is not merely a consequence of one favorable probe run. Repeated Probe100 runs produce highly stable rankings, while comparisons with representation- and component-based signals further support the utility of action-facing responses; complete statistics are reported in Appendix A.7 and Appendix A.4. We emphasize that Tail-AV is a short-budget ranking signal rather than a calibrated surrogate for final behavioral damage.

The remaining question is whether this ranking advantage survives the step that ultimately matters: training a full attack under a matched budget. Section 5.3 therefore compares Tail-AV-selected routes against representation-based and uninformed selections while holding the final objective, target count, geometry, and training budget fixed. If action-facing vulnerability captures something genuinely useful beyond representation perturbability, its selected routes should support stronger final attacks. Section 4 turns exactly this criterion into PASA-KV.

4 From Characterization to Attack: PASA-KV

Section 3 shows that internal routes differ substantially in behavioral vulnerability, while representation displacement alone provides an incomplete basis for deciding where to attack. The practical difficulty is that directly resolving this uncertainty is expensive: fully training and evaluating an

attack at every candidate layer would spend most of the budget merely discovering which routes matter. PASA-KV addresses this problem by *decoupling route identification from full attack optimization*. As illustrated in Figure 2, it first uses inexpensive singleton probes to expose the short-budget action-facing response of each candidate route, selects promising layers with Tail-AV, and only then allocates the full optimization budget to a fresh multi-route attack.

4.1 Attack setting and route probing

We consider a known-task white-box attack (Goodfellow et al., 2015; Madry et al., 2018) with a frozen victim and ten LIBERO-10 tasks (Liu et al., 2023a). A single RGB patch p is shared across tasks, drawing on localized patch and physical-world attack formulations (Brown et al., 2017; Athalye et al., 2018; Eykholt et al., 2018). During training, the patch modifies clip frame 0; at deployment, it is applied to the current primary observation at every replan without rewriting history. Primary WAM protocols use current-view patching without transformation sampling. Architecture-specific interfaces, geometries, and deployment details appear in Appendix A.1.

The first stage is designed only to reveal how each candidate route responds under a small, common perturbation budget. For a candidate layer ℓ, we optimize a singleton probe patch against the relative displacement of its visual K/V:

$$D_\ell^Z = \frac{\|Z_\ell(x \oplus p) - Z_\ell(x)\|_F}{\max(\|Z_\ell(x)\|_F, \epsilon)}, \quad Z \in \{K, V\}, \qquad J_\ell^{KV} = \tfrac{1}{2}\mathbb{E}[D_\ell^K + D_\ell^V]. \tag{4}$$

Targeting layer ℓ specifies where the optimization loss is applied, rather than assuming that the induced perturbation remains localized to that layer. FastWAM uses $\epsilon = 10^{-8}$ and 100 exposures per candidate layer (Probe100). Only RGB patch pixels are optimized through the frozen victim, without unrolling action denoising (Chi et al., 2025; Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Liu et al., 2023b); optimization and capture details appear in Appendix A.2.

Importantly, the probe objective itself is *not* used to rank layers. Section 3 shows that large K/V displacement does not reliably imply strong behavioral vulnerability. Instead, the short probe exposes how the induced perturbation enters the action-facing readout, measured next.

4.2 Action-facing route selection

Following Equation 3, we characterize each probed layer through the contribution $I = A\|V\|_2$, which combines action-query routing with the magnitude of visual content being read. For task t and candidate layer ℓ, we flatten I over H heads, Q action queries, and T visual tokens. Let $\mu_q(I)$ denote the mean of its largest $\lceil qHQT \rceil$ entries, with $q = 0.005$. Tail-AV measures the relative growth of this sparse upper tail between the patched and clean forwards:

$$s_\ell^{\mathrm{Tail}} = \frac{1}{10}\sum_{t=1}^{10} \log\left[\max\left(\frac{\mu_q(I_{\ell t}^p)}{\max(\mu_q(I_{\ell t}^0), \delta)}, \delta\right)\right], \qquad \delta = 10^{-12}. \tag{5}$$

The score is averaged across one held-out scoring observation from each of the ten known tasks, so selection favors routes whose action-facing response is consistently exposed across the task set rather than routes dominated by a single task instance. These observations are excluded from patch optimization, and no closed-loop rollout outcome enters the score. Candidate layers are ranked by descending s_ℓ^{Tail}, with ascending layer ID as the tie-breaker, and the Top-k set S is retained. Full scoring details and layerwise rankings are reported in Appendix A.3.

This stage separates two roles that would otherwise be conflated. The singleton probe provides a controlled perturbation with which to interrogate each route, while Tail-AV determines whether the resulting change is strongly exposed to action queries. The former creates the diagnostic signal; the latter decides whether the route merits the substantially larger final-training budget.

Table 1: **Fixed-patch comparisons with WAM-adapted baselines.** Policy SR (%, lower is stronger). All conditions use LIBERO-10 Eval500 (10 tasks $\times$ 50 trials). Each PASA-KV Top-k condition uses a separately trained, task-shared patch (10k exposures); bold marks the lowest observed SR per architecture. GPU-h reports recorded construction cost; FastWAM Top-k includes Probe100 (Appendix A.8).

	FastWAM		LingBot-VA		Cosmos Policy		
Method	SR ↓	GPU-h ↓	SR ↓	GPU-h ↓	SR ↓	GPU-h ↓	Protocol
Shared clean reference	92.4	–	92.4	–	96.8	–	Shared Eval500 reference
DP-Attacker	81.0	4.23	94.2	1.97	95.2	2.88	Diffusion
DRIFT	53.0	4.78	92.4	2.46	**89.2**	3.33	Flow/EDM
UPA-RFAS	90.8	10.75	91.0	8.94	**89.2**	23.74	No PSM
UPA	77.4	6.45	94.2	8.65	89.6	3.14	Action
TMA	81.2	6.12	93.0	8.49	90.2	3.13	Action
UADA	71.4	6.25	95.0	8.64	**89.2**	3.09	Action
PASA-KV (Tail-AV Top-3)	1.20	2.33	88.0	0.96	89.6	2.98	Probe–select
PASA-KV (Tail-AV Top-4)	4.40	2.40	90.2	1.10	89.6	3.39	Probe–select
PASA-KV (Tail-AV Top-5)	**0.00**	2.35	**87.2**	1.55	**89.2**	3.40	Probe–select

4.3 Fresh multi-route attack training

After selection, we deliberately discard all singleton probe patches and their optimizer states. A fresh task-shared patch is initialized and jointly optimized over the selected routes:

$$J_{\text{joint}}(p; S) = |S|^{-1} \sum_{\ell \in S} J_\ell^{KV}(p). \tag{6}$$

The final patch is trained for 10k exposures with the victim fixed throughout. For L candidate layers, the complete construction therefore requires $100L$ probe exposures followed by one 10k-exposure final optimization, in addition to the lightweight scoring forwards.

Training a fresh patch is important for separating *selection* from *attack optimization*. If the optimized singleton probe were directly reused, performance could reflect the quality of a particular probe solution rather than the quality of the selected route itself. Resetting the patch ensures that Tail-AV contributes only the selected layer set S; all compared final attacks must learn their perturbation anew under the same full-training objective and budget. This separation also makes the matched controls in Section 5.3 a direct test of route selection rather than probe reuse.

The resulting procedure is therefore intentionally simple: Probe100 asks how each candidate route responds under a common small budget, Tail-AV identifies which responses are most exposed to action generation, and fresh joint training concentrates the full budget on those routes. In this way, PASA-KV turns the action-facing characterization of Section 3 into a concrete attack without using closed-loop outcomes to search over target layers.

5 Experiments

5.1 Experimental protocol

We evaluate direct FastWAM, cascaded LingBot-VA, and joint Cosmos Policy on LIBERO-10 under their documented inference settings; clean controls appear in Appendix D.1. FastWAM uses our frozen checkpoint and released inference configuration, whereas only LingBot-VA and Cosmos Policy use reduced-step solvers. Method-specific geometries and protocols are detailed in captions and appendices.

5.2 Main results: closed-loop attacks and construction costs

Table 1 shows that Top-5 reaches the lowest observed SR on FastWAM, LingBot-VA, and Cosmos (0.0/87.2/89.2%; tied on Cosmos). Recorded PASA-KV construction costs are lower than the listed

Table 2: **Matched FastWAM selection controls.** Same joint K/V loss, three targets, geometry, and 10k exposures; Eval200 (10 tasks × 20 trials). Dots denote retrospective response classes.

Selection rule	Layer set	Class	Successes	Policy SR (%)
Probe-J^{KV}	L27/L23/L28	●/●/●	95/200	47.5
Evenly spaced	L00/L15/L29	●/●/●	37/200	18.5
Random-1	L27/L18/L16	●/●/●	173/200	86.5
Random-2	L12/L23/L00	●/●/●	127/200	63.5
Random-3	L20/L22/L00	●/●/●	50/200	25.0
Tail-AV	L29/L02/L03	●/●/●	3/200	**1.5**

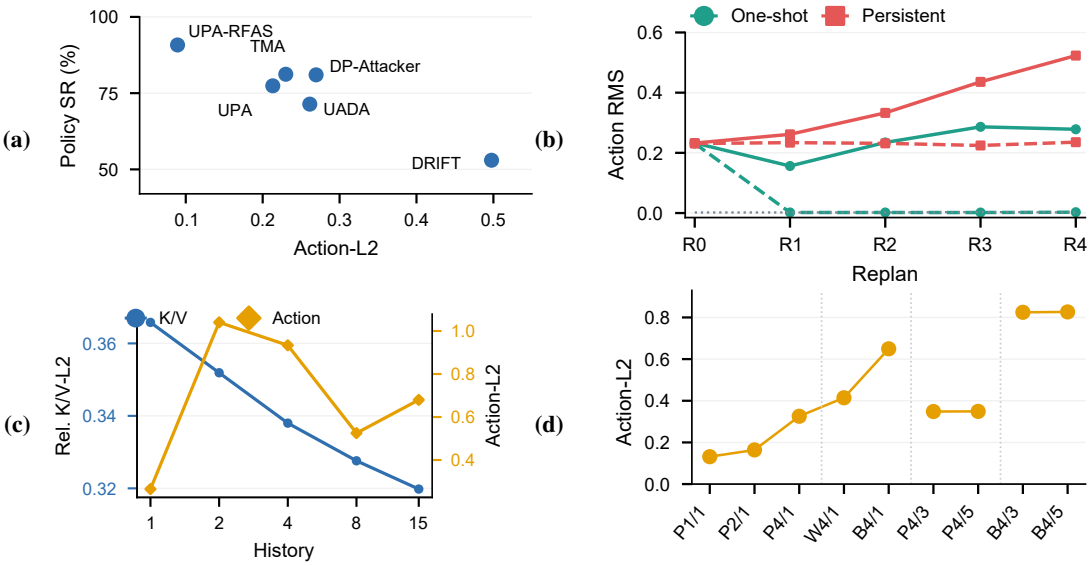


Figure 3: **Action-response diagnostics. (a)** FastWAM Action-L2 versus SR across six patches ($r = -0.956$, $\rho = -0.714$). **(b)** One-shot and persistent exposure; dashed/solid denote local/trajectory responses. **(c)** LingBot-VA response over history length (log scale). **(d)** Cosmos σ-normalized cumulative Action-L2. P/W/B denote primary/wrist/both views; suffixes give patched-frame and joint-step counts. Protocols: Appendices C.1, D.3–D.5.

adaptations on FastWAM and LingBot-VA; on Cosmos, DP-Attacker's 2.88 GPU-h is marginally lower than PASA-KV Top-3's 2.98 GPU-h. The smaller effects on LingBot-VA and Cosmos Policy highlight architecture-dependent resistance.

5.3 Matched controls validate readout-informed selection

The matched controls fix the victim, final K/V loss, target count, geometry, and 10k budget. Tail-AV obtains 1.5% SR, compared with 47.5% for Probe-J^{KV}, 18.5% for evenly spaced selection, and 25.0–86.5% for fixed random sets (Table 2). The 46-point gap to Probe-J^{KV} supports readout-informed selection over optimized K/V displacement alone.

With the layer sets and Probe100 ranking held fixed, three final-training reruns give Tail-AV 3/200 successes in every run. The lowest control SR is 18.5%, 9.5%, and 5.5% across the respective runs; the best-performing control differs by run (Appendix Table 10). Across eight additional Probe100 runs with fixed scoring pairs, the complete layer rankings remain highly stable (Spearman 0.9942–0.9987); Top-4/5 membership is unchanged and Top-3 changes once (Appendix Table 11).

5.4 Closed-loop diagnostics contextualize attack effectiveness

Action discrepancy is informative but does not fully rank attacks. Across 30 matched BS=1, 20k layers, Action-L2 is negatively associated with Eval200 SR ($r = -0.912$; $\rho = -0.880$; Ap-

pendix Table 27). This complements the six-baseline diagnostic, where 3/15 pairwise orders reverse (Figure 3(a)); together, the results show that action discrepancy is informative but does not replace closed-loop evaluation.

Repeated exposure sustains disruption beyond the initial action. For the FastWAM Top-3 patch, removing a one-shot patch largely restores same-state actions but not the resulting trajectory; persistent exposure increases trajectory RMS (Figure 3(b)). Changed actions and observations can therefore sustain divergence without an internal history cache (Appendix D.3).

Broader internal loss targets do not consistently strengthen attacks. Broader LingBot/Cosmos loss targets show no consistent singleton-attack gain (Appendix Table 29). LingBot history changes K/V and action responses differently, while Cosmos Action-L2 varies with patched input extent and solver settings (Figure 3(c,d); Appendices D.4 and D.5).

5.5 Selective K/V Guard mitigates fixed Tail-AV attacks

Whereas PatchGuard and PatchCleanser address input-level patch defenses (Xiang et al., 2021; 2022), Guard is an internal-route intervention independently selected by action sensitivity rather than Tail-AV. Against a separate Top-3/4/5 patch family, its five targeted layers raise pooled SR from 1.33% to 87.50% while retaining 93.5% clean SR; all-layer Guard reduces clean SR to 53.0%. Targeted suppression of attention-dominant and large weighted-value contributions therefore recovers fixed-attack performance without a clean test-time cache. Protocol, equations, and full results appear in Appendix E.

5.6 PASA-KV disrupts real-robot closed-loop control

On a separately trained Piper FastWAM checkpoint, we reran the full probe–select–train pipeline. As in the primary experiments, each Top-k uses a fresh task-shared patch trained across the four-task Piper suite and digitally overlaid on the head camera at every replan (Appendix F).

Table 3: **Piper real-robot closed-loop success over 10 trials per task.** Top-3 and Top-5 are separately trained patches.

Task	Clean	Tail-AV Top-3	Tail-AV Top-5
Stack three bowls	10/10	0/10	0/10
Place carrot in bowl	4/10	2/10	0/10
Place pen in cup	6/10	4/10	0/10

On bowl stacking, Top-3 reaches 0/10, matched only by DP-Attacker and UADA among the six baselines (Appendix Table 36). Top-5 yields 0/10 on all three evaluated tasks (Table 3).

6 Discussion

Our evidence is bounded by comparison scope. External comparisons cover the implemented WAM adaptations and one checkpoint per architecture, supporting system-level rather than architecture-family claims. Real-robot evaluation covers three tasks; the full baseline comparison is limited to bowl stacking, and Top-4 was not physically evaluated.

7 Conclusion

We introduce PASA-KV, which uses short action-facing probes to select visual K/V routes before training a fresh attack. Matched controls validate readout-informed selection. Across architectures, Top-5 reaches 0.0%, 87.2%, and 89.2% SR on FastWAM, LingBot-VA, and Cosmos Policy; recorded construction costs are lower on FastWAM and LingBot-VA. On a separate Piper checkpoint, one task-shared Top-5 patch yields 0/10 success on three clean-capable tasks. Together, the results distinguish representation perturbability, action-facing response, and closed-loop vulnerability.

AI USE STATEMENT

Generative AI tools were used to assist with literature search and synthesis, language editing and writing refinement, and the creation or refinement of scientific figures. AI-generated suggestions and materials were reviewed, revised, and verified by the authors before inclusion in the manuscript. The authors take responsibility for the final content of this work, including text and figures produced with the aid of generative AI.

ETHICS STATEMENT

This work studies adversarial attacks to characterize and reduce security risks in robot-control models. The evaluation includes benchmark simulation and controlled tabletop rollouts on a 6-DoF Piper robot. Physical trials were limited to the predefined task suite. No human-subject study was conducted. The work is intended to support robustness research. Because attack methods can be dual-use, we report threat-model constraints, defense failures, and adaptive-attack limitations rather than presenting the mitigation as complete. Any public release of attack artifacts should be reviewed for responsible disclosure and misuse risk.

REPRODUCIBILITY STATEMENT

Appendix A.1 provides the recovered FastWAM victim-training configuration and exact checkpoint identity. Construction budgets, known-task scoring definitions, and complete rankings appear in Appendices A.2–A.3; the K/V mechanism decomposition, counterfactual protocol, full heatmaps, and Probe100 predictive and repeat ledgers appear in Appendices A.5–A.7; matched selection controls in Appendix A.4; baseline adaptations and numerical diagnostics in Appendix C. Complete layer-wise attack and footprint results are in Appendix B; cross-architecture diagnostics in Appendix D; and Guard selection, equations, and mitigation results in Appendix E. Appendix F documents the Piper platform, real-robot patch construction and deployment, evaluated task coverage, and shared-task baseline results. The primary attack protocol is indexed by label 42, while singleton training configurations record seed 20260801. These are reported as distinct run identifiers; analyses do not assume matched random initializations across conditions, and each result is linked to its own victim, patch, and evaluation records. We will release the trained victim checkpoint (`step_020000.pt`) after acceptance; code and machine-readable result ledgers are also planned for release.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 284–293, 2018.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H. Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.025. URL `https://roboticsproceedings.org/rss19/p025.html`.

Tom B. Brown, Dandelion Mané, Aurko Roy, Martin Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791, 2021. doi: 10.1109/CVPR46437.2021.00084.

Yipu Chen, Haotian Xue, and Yongxin Chen. Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies. In *Advances in Neural Information Processing Systems*, volume 37, pp. 119614–119637, 2024. doi: 10.52202/079017-3800.

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. doi: 10.1177/02783649241273668.

Kevin Eykholt, Ivan Evtimov, Earlence 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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018. doi: 10.1109/CVPR.2018.00175.

Jiyuan Fu, Kaixun Jiang, Jingkai Jia, Zhaoyu Chen, Xueyao Chen, Lingyi Hong, Shuyong Gao, Chenzhi Tan, Dingkang Yang, and Wenqiang Zhang. Vla-hijack: A transferable patch attack against vision-language-action models via visual proprioception hijacking. arXiv preprint arXiv:2605.28083, 2026.

Yonggan Fu, Shunyao Zhang, Shang Wu, Cheng Wan, and Yingyan Celine Lin. Patch-Fool: Are vision transformers always robust against adversarial perturbations? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=28ib9tf6zhr.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems 31*, pp. 2451–2463. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2555–2565, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Zhengxian Huang, Wenjun Zhu, Haoxuan Qiu, Xiaoyu Ji, and Wenyuan Xu. TRAP: Hijacking VLA cot-reasoning via adversarial patches. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zLUPTgV7BI.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3543–3556, 2019. doi: 10.18653/v1/N19-1357.

Jaehwan Jeong, Evelyn Zhu, Jinying Lin, Emmanuel Jaimes, Tuan-Anh Vu, Jungseock Joo, Sangpil Kim, and M. Khalid Jawed. Your vision-language-action model already has attention heads for path deviation detection. arXiv preprint arXiv:2603.13782, 2026.

Eliot Krzysztof Jones, Alexander Robey, Andy Zou, Zachary Ravichandran, George J. Pappas, Hamed Hassani, Matt Fredrikson, and J. Zico Kolter. Adversarial attacks on robotic vision language action models. arXiv preprint arXiv:2506.03350, 2025.

Ameya Joshi, Gauri Jagatap, and Chinmay Hegde. Adversarial token attacks on vision transformers. *arXiv preprint arXiv:2110.04337*, 2021. URL https://arxiv.org/abs/2110.04337.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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. arXiv preprint arXiv:2406.09246, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. doi: 10.48550/arXiv.2601.21998. URL https://arxiv.org/abs/2601.21998v2.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791, 2023a. doi: 10.52202/075280-1939.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023b.

Giulio Lovisotto, Nicole Finnie, Mauricio Munoz, Chaithanya Kumar Mummadi, and Jan Hendrik Metzen. Give me your attention: Dot-product attention considered harmful for adversarial patch robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15213–15222, 2022. doi: 10.1109/CVPR52688.2022.01480.

Hui Lu, Yi Yu, Yiming Yang, Chenyu Yi, Qixin Zhang, Bingquan Shen, Alex C. Kot, and Xudong Jiang. When robots obey the patch: Universal transferable patch attacks on vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22867–22878, 2026.

Weimin Lyu, Songzhu Zheng, Tengfei Ma, Haibin Ling, and Chao Chen. Attention hijacking in trojan transformers. arXiv preprint arXiv:2208.04946, 2022.

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.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1262.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

Darshan Nagendra Prasad, Lars Ullrich, and Knut Graichen. Inference-time attention steering for vision-language-action driving models. In *Proceedings of the DriveX Workshop on Foundation Models for Autonomous Driving, held in conjunction with ECCV*, 2026. URL https://arxiv.org/abs/2608.17095. Poster.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 21527–21536, 2024. doi: 10.1609/aaai.v38i19.30150.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Hoseong Tae and Jong-Seok Lee. DRIFT: Derailing denoising trajectories of flow-matching VLAs with adversarial patch attack. *arXiv preprint arXiv:2608.03207*, 2026. URL https://arxiv.org/abs/2608.03207.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html.

Taowen Wang, Cheng Han, James Liang, Wenhao Yang, Dongfang Liu, Luna Xinyu Zhang, Qifan Wang, Jiebo Luo, and Ruixiang Tang. Exploring the adversarial vulnerabilities of vision-language-action models in robotics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6948–6958, 2025. doi: 10.1109/ICCV51701.2025.00653.

Xiaofei Wang, Mingliang Han, Tianyu Hao, Yi Yang, Yun-Bo Zhao, and Keke Tang. Partially observable adversarial patch attacks on vision-language-action models in robotics. *IEEE Robotics and Automation Letters*, 11(8):9351–9358, 2026. doi: 10.1109/LRA.2026.3703262.

Zhipeng Wei, Jingjing Chen, Micah Goldblum, Zuxuan Wu, Tom Goldstein, and Yu-Gang Jiang. Towards transferable adversarial attacks on vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2668–2676, 2022. doi: 10.1609/aaai.v36i3.20169.

Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. PatchGuard: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium*, pp. 2237–2254, 2021. URL https://www.usenix.org/conference/usenixsecurity21/presentation/xiang.

Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. PatchCleanser: Certifiably robust defense against adversarial patches for any image classifier. In *31st USENIX Security Symposium*, pp. 2065–2082, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/xiang.

Jingyao Xu, Yuetong Lu, Yandong Li, Siyang Lu, Dongdong Wang, and Xiang Wei. Perturbing attention gives you more bang for the buck: Subtle imaging perturbations that efficiently fool customized diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24534–24543, 2024. doi: 10.1109/CVPR52733.2024.02316.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. doi: 10.48550/arXiv.2603.16666. URL https://arxiv.org/abs/2603.16666v2.

Jinquan Zhang, Dongfu Yin, Run Yang, Yufeng Yan, Zhen Tian, and F. Richard Yu. Structure-aware robust fine-tuning: Defending vision-language-action robots against physical attention hijacking. arXiv preprint arXiv:2608.03231, 2026. URL https://arxiv.org/abs/2608.03231v1.

Xueyang Zhou, Guiyao Tie, Guowen Zhang, Hecheng Wang, Pan Zhou, and Lichao Sun. BadVLA: Towards backdoor attacks on vision-language-action models via objective-decoupled optimization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 127496–127523, 2025. doi: 10.52202/085713-4247.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Reproducibility details

A.1 Documented threat models

All attacks use frozen victims on known tasks, with checkpoint selection preceding patch training. Table 4 records patch lineage and evaluation denominators.

The frozen FastWAM victim uses precomputed text embeddings and FP32 training; attacks use BF16 inference. Periodic training evaluation and simulator-rollout Policy SR are separate, and construction costs exclude victim training.

Inference and deployment. FastWAM encodes video once at time 0 before ten action steps. Main patches lie outside manipulation regions, patch clip frame 0 during training, and patch each current observation at deployment. Table 6 gives Eval500 references; reduced-step controls and alternate geometries are documented separately below.

A.2 Patch construction and evaluation

Probe100 optimizes a singleton K/V loss for 100 exposures per candidate layer. Final training resets to the designated initial patch and optimizes the mean selected-layer loss for 10,000 BS1 updates; probe pixels and optimizer state are discarded. The primary suite uses protocol/run label 42, while singleton configurations use seed 20260801. Thus, L candidates require $100L + 10{,}000$ exposures plus scoring forwards. Table 7 gives the optimizer.

Patch composition and strength. With RGB input $x \in [0, 1]$, binary rectangle mask M, and embedded patch $P(p)$, $x \oplus p = (1 - M) \odot x + M \odot P(p)$. Patch values are converted to $[-1, 1]$ before replacement in the normalized video tensor; all other pixels and frames are unchanged. This recipe has no blending, transformation sampling, or pixel-distance penalty. Tail-AV scores each update-100 probe at full strength against the unpatched input.

Loss tensors and gradient path. FastWAM encodes fixed clips at video timestep zero. Equation 4 computes FP32 relative visual-cache norms (denominator floor 10^{-8}) and averages K/V components, selected layers, and batch samples equally. The clean cache has no gradients; the attacked path backpropagates only to the FP32 patch, with the victim frozen. Runs reset their designated initial pixels and optimizer states, clamp pixels after every update, and evaluate update 10,000. Headline Top-3/4/5 patches use Eval500 (trials 0–49 per task); evaluation seeds are 42 for FastWAM/LingBot and 195 for Cosmos extensions.

Table 4: **Experiment map.** Patch lineage is listed for each evidence block. Eval200/500 denote 20/50 online trials per task.

Evidence	Patch relationship	Endpoint and role
Primary comparison	Separately trained Top-3, Top-4, and Top-5 patches per model	Eval500; baseline comparison and target-count tradeoff
Matched selection	Primary FastWAM Top-3 plus separately trained controls	Common Eval200; isolates layer-set choice
Training repeats	Same layer sets and initialization; new final training	Eval200 per run; fixed-set repeatability
Layerwise analysis	Separate singleton-target patches	SR trajectories and paired internal responses; target characterization
Input diagnostics	Patch specified per panel	Paired action/representation responses; FastWAM also has 30 diagnostic rollouts
Guard	Separate fixed Top-3/4/5 family	200 episodes per patch and deployment; mitigation
Real robot	Piper-specific fresh Top-3/5 patches	Three shared tasks, 10 trials per task; physical closed-loop control

Table 5: **FastWAM victim training configuration.** Values describe the frozen victim model used for the FastWAM experiments.

Item	Documented record
Hardware and distributed training	8 MetaX GPUs; DeepSpeed ZeRO-2; FP32; seed 42; per-GPU batch 8; gradient accumulation 2; global batch 128
Data	LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10; 40 tasks; 1,712 demonstrations; 277,713 training frames
Schedule	10 planned epochs; 21,700 planned optimizer updates; save/eval/log frequencies 2,000/200/10 updates
Optimizer and schedule	AdamW; learning rate 10^{-4}; betas $(0.9, 0.95)$; epsilon 10^{-8}; weight decay 0.01; gradient clipping 1.0; 1,085-update 5% warm-up; cosine decay to 10^{-6}; no EMA or early stopping
Model	`Wan-AI/Wan2.2-TI2V-5B`, ModelScope revision `c2272aadfec2d95cea869a4e748d5e28275621c6`; approximately 5.00B video-expert and 1.02B action-expert parameters; 30-layer MoT; video/action/proprioception trainable, VAE and other modules frozen; both loss weights 1.0
Input and inference	Two 512×512 views resized to 224×224 and concatenated to 224×448; 9 video frames; 32-step 7-D action chunk; 8-D proprioception; min/max normalization; no random image augmentation; attack inference BF16 with 10 action-denoising steps
Validation scope	`val_set_proportion` $= 0$; periodic evaluation reused the training set, while paper Policy SR uses separate simulator rollouts.

A.3 TAIL-AV DEFINITION AND IMPLEMENTATION SCOPE

Tail-AV uses ten fixed clean/patched scoring pairs, one per task, held out from patch optimization. Scoring forwards provide internal responses and simulator rollouts provide Policy SR. Equation 5 takes the largest $\lceil 0.005HQT \rceil$ clean and attacked influences independently, forms their floored log ratio, and averages tasks equally.

FastWAM scores fixed-clip visual caches at the first action-denoising step. After masked softmax over all keys, it selects current-frame visual columns without within-span renormalization. Influence has shape $[1, H, Q, T]$; clean/patched forwards share action noise. Full-strength FastWAM probes require $30 \times 10 = 300$ scoring pairs.

The primary FastWAM ranking, `tail_av_order`, sorts internal Probe100 scores descending with ascending layer-ID tie-breaking. Policy SR, footprint, Rescue, Transfer, and Action-L2 remain evaluation diagnostics; response groups are interpretive overlays.

Table 6: Architecture-specific main threat models. Historical frames and cached states are not retroactively modified.

Victim	Inference graph and visual route	Patch (px x, y)	@ Evaluation solver	Clean
FastWAM	Direct; current-primary action-facing video K/V	50×50 $(5, 160)$	at 10 action steps	462/500 (92.4%)
LingBot-VA	Cascaded; current agent-view video-to-action route	29×29 $(3, 91)$	at 1 video + 10 action steps	462/500 (92.4%)
Cosmos Policy	Joint; current-primary conditioning slot	50×50 $(5, 160)$	at 1 joint denoising step	484/500 (96.8%)



Figure 4: **FastWAM patch geometry on LIBERO-10.** Tasks T0–T9 share $(x, y, w, h) = (5, 160, 50, 50)$ on the 224×224 primary view (4.98% of that view; 2.49% of the two-view input). The secondary view is unpatched.

Architecture-specific implementation scope. Table 9 specifies each capture interface. LingBot-VA and Cosmos Policy use the same top-0.5% task-mean log ratio over $[1, H, Q, T]$ influence, with 30 and 28 candidate layers, respectively.

Development and selection scope. Rules and layer sets use the known LIBERO-10 task suite and are fixed before final-patch rollouts. Each Top-k prefix listed below is a fresh joint patch.

A.4 Selector efficacy controls

All six conditions in main-text Table 2 share three targets, the victim, final joint K/V loss, geometry, BS1, and 10k exposures. Controls use the common 200-episode grid (tasks 0–9; trials 0–19); deterministic rules fix random sets before training. The primary Tail-AV patch yields 3/200 successes on this grid and 6/500 on Eval500.

Fixed-layer final-training repeats. Table 10 adds seeds 43/44 to the seed-42 Eval200 records while fixing each method's layer set, final objective, and 10k budget. Tail-AV has lower SR than every control in every run. These runs vary final-patch training only; mean and sample SD summarize the three runs.

Probe-run selector reproducibility. The canonical FastWAM Probe100 run and eight additional runs (probe seeds 42–44) share 30 candidates, ten task-balanced scoring pairs, and the 100-exposure singleton budget. Across the eight complete rankings, correlation with the canonical ranking is 0.994216–0.998665 (mean 0.996941); Top-4/5 are identical and Top-3 differs once (L01 for L03). This tests scoring-run variation separately from final-patch training.

Table 7: FastWAM PASA-KV attack-construction recipe. These parameters train the RGB patch, not the victim. Probe and final training share the optimizer; only the loss-target set and update budget change.

Setting	Documented implementation
Trainable variable	FP32 RGB patch, $3 \times 50 \times 50$; frozen victim, BF16 forward
Optimizer	AdamW; learning rate 5×10^{-4}, betas $(0.9, 0.95)$, epsilon 10^{-8}, weight decay 0.01
Schedule and gradients	Constant learning rate; no warm-up, scheduler, gradient clipping or accumulation in this patch loop
Initialization	Uniform $[0, 1]$ initializer; primary attack suite indexed by protocol/run label 42; singleton training configurations record seed 20260801; final training starts from fresh pixels
Projection and loss	Clamp pixels to $[0, 1]$ after every update; minimize $-J^{KV}$ (singleton) or $-J_{\text{joint}}$; image-L1 weight zero
Updates and batch	BS1, one update per exposure; 100 per singleton probe, 10,000 per final patch
Training inputs	LIBERO-10 observation clips; documented scoring-input exclusion list; clip frame 0 primary view only is patched
Patch selection	Update 100 for probes; update 10,000 for final evaluation; no rollout-based checkpoint selection

Table 8: **Production Tail-AV selector scores and ranking on the evaluated FastWAM checkpoint.** Descending Eq. 5, with layer-ID tie-breaking; scores rounded to six decimals. Top-3/4/5 prefixes give the selected attack sets.

Rank	Layer	Score	Rank	Layer	Score	Rank	Layer	Score
1	L29	0.049694	11	L14	0.009106	21	L28	−0.008567
2	L02	0.043262	12	L19	0.007954	22	L10	−0.017586
3	L03	0.032456	13	L16	0.004701	23	L26	−0.029588
4	L01	0.027921	14	L17	0.004437	24	L22	−0.034651
5	L24	0.026162	15	L15	0.002879	25	L09	−0.038692
6	L00	0.023169	16	L18	0.002561	26	L06	−0.043447
7	L20	0.021821	17	L13	0.001807	27	L05	−0.047775
8	L12	0.019057	18	L11	−0.000352	28	L07	−0.049264
9	L27	0.014059	19	L21	−0.000550	29	L08	−0.061348
10	L25	0.012028	20	L23	−0.008298	30	L04	−0.075180

Component-score rankings. Table 12 and Figure 5 compare all 30 layers from the shared Probe100 component-ablation run. Attention-only-tail retains routing alone; Tail-AV additionally includes the magnitude of received visual content. The two are the strongest continuous signals, while Mean-AV and Value-only-tail are weaker. Tail-AV has the strongest Early-layer alignment (AUC 0.8624; 3/3 and 4/5 Early layers at Top-3/5), so it is retained as the routing–content implementation.

A.5 Action-facing K/V mechanism analysis

First-order decomposition. For an action query q_i and a visual token j, let

$$a_{ij} = \mathrm{softmax}_j\left(q_i^\top k_j/\sqrt{d}\right), \qquad y_i = \sum_j a_{ij} v_j. \tag{7}$$

Holding q_i fixed and perturbing only visual K/V gives the first-order expansion

$$\delta y_i \approx \underbrace{\frac{1}{\sqrt{d}} \sum_{j \in \mathcal{V}} a_{ij}(q_i^\top \delta k_j)(v_j - y_i)}_{u_K \text{ (K-mediated routing)}} + \underbrace{\sum_{j \in \mathcal{V}} a_{ij} \delta v_j}_{u_V \text{ (V-mediated content)}}. \tag{8}$$

The K term depends on query alignment and value contrast, while the V term is gated by action-facing attention; raw representation displacement alone is therefore insufficient. The joint contribu-

Table 9: Architecture-specific selector capture interfaces. The common influence reduction is specified in Appendix A.3.

System	Scoring interface	Attention normalization and visual slice
FastWAM	Fixed-clip cache; first action-denoising step; 30 layers	Masked softmax over available keys, then current-frame visual columns; no within-slice renormalization.
LingBot-VA	One video step (endpoint 0.6), followed by the first of ten action steps; 30 layers	Action queries attend to full video-cache and action keys; retain current primary-view cache columns and values, conditional batch only.
Cosmos Policy	One joint-noise level; action-slot 4 queries; blocks 0–27	Softmax over all joint-slot keys; retain current-primary slot 3 columns and values. Slot indices are zero-based.

Each interface forms $I = A\|V\|_2$ without renormalizing attention inside the selected visual slice. The selected layer sets are listed in the appendix text; Top-k outcomes, denominators, and recorded construction costs appear in main-text Table 1.

Table 10: **Fixed-layer final-training repeats on FastWAM.** Success counts per 200 episodes; lower is stronger attack. Mean SR and sample SD summarize the three runs with equal weights. Here 42/43/44 are final-training seeds, while the singleton sweeps use protocol label 42. All five matched controls are retained, and the first Tail-AV column draws the primary Eval500 subset.

Method	Seed 42	Seed 43	Seed 44	Mean SR (%)	SD (pp)
Tail-AV Top-3	3	3	3	1.50	0.00
Probe-J^{KV}	95	94	89	46.33	1.61
Evenly spaced	37	53	43	22.17	4.04
Random-1	173	170	169	85.33	1.04
Random-2	127	131	11	44.83	34.08
Random-3	50	19	25	15.67	8.22

tion contains a cross-term:

$$\|u_K + u_V\|_2^2 = \|u_K\|_2^2 + \|u_V\|_2^2 + 2\langle u_K, u_V\rangle. \tag{9}$$

For reference, Cauchy–Schwarz gives the bounds

$$\begin{aligned} \|u_K\|_2 &\le \frac{\|q_i\|_2}{\sqrt{d}} \left(\sum_j a_{ij}\|\delta k_j\|_2^2\right)^{1/2} \left(\sum_j a_{ij}\|v_j - y_i\|_2^2\right)^{1/2}, \\ \|u_V\|_2 &\le \left(\sum_j a_{ij}\|\delta v_j\|_2^2\right)^{1/2}. \end{aligned} \tag{10}$$

Thus, global D_K and D_V do not alone characterize action-facing vulnerability. With $g_{KV} = g_K + \lambda g_V$, the cross-term is a theoretical account of layer-specific reinforcement or opposition.

A.6 Counterfactual readout protocol and complete mechanism results

The mechanism sweep covers 30 FastWAM layers, three optimization objectives, and 50 observations from ten task-balanced inputs. K-only and V-only denote optimization objectives, so their patches may alter both cache components. For every joint K+V patch, we fix Q^0_{action} and construct:

$$\begin{aligned} Y_{00} &= Y(Q^0, K^0, V^0), \quad Y_{K0} = Y(Q^0, K^p, V^0), \\ Y_{0V} &= Y(Q^0, K^0, V^p), \quad Y_{KV} = Y(Q^0, K^p, V^p). \end{aligned} \tag{11}$$

Let $\Delta_K = Y_{K0} - Y_{00}$, $\Delta_V = Y_{0V} - Y_{00}$, $\Delta_{KV} = Y_{KV} - Y_{00}$, and $\Delta_{\text{int}} = Y_{KV} - Y_{K0} - Y_{0V} + Y_{00}$. The identity $\Delta_{KV} = \Delta_K + \Delta_V + \Delta_{\text{int}}$ has maximum relative residual 1.273×10^{-7}. Attention

Table 11: **FastWAM Tail-AV probe-seed rank stability under fixed scoring pairs.** Spearman correlations compare each of eight additional complete 30-layer rankings with the canonical Tail-AV ranking. Top-k columns indicate exact set agreement with the canonical Top-k set. The only Top-3 exception is Seed44-B, which swaps L03 for L01; Top-4 and Top-5 membership remain unchanged.

Run	Probe seed	Spearman	Top-3	Top-4	Top-5
Canonical	42	1.000000	Ref.	Ref.	Ref.
Seed42-A	42	0.998665	Yes	Yes	Yes
Seed42-B	42	0.997775	Yes	Yes	Yes
Seed43-A	43	0.998665	Yes	Yes	Yes
Seed43-B	43	0.996885	Yes	Yes	Yes
Seed43-C	43	0.994216	Yes	Yes	Yes
Seed44-A	44	0.996440	Yes	Yes	Yes
Seed44-B	44	0.995996	No	Yes	Yes
Seed44-C	44	0.996885	Yes	Yes	Yes
Mean (8 additional)	–	0.996941	7/8	8/8	8/8

Table 12: **Selector-score component ablation on the evaluated FastWAM checkpoint.** All four score variants use the same Probe100 component-ablation run (30 layers and ten task-balanced scoring pairs). ρ_{10k} and ρ_{20k} are Spearman correlations with later layer AttackStrength; Early-group AUC and Early@Top-k are retrospective diagnostics using the nine Early layers as positives.

Score	Early-group AUC ↑	ρ_{10k}	ρ_{20k}	Early@Top-3	Early@Top-5
Tail-AV	**0.8624**	**0.521**	0.508	**3/3**	**4/5**
Attention-only-tail	0.7672	0.506	**0.537**	**3/3**	3/5
Mean-AV	0.7831	0.229	0.173	2/3	3/5
Value-only-tail	0.6931	0.100	0.143	2/3	3/5

uses the full key set; visual columns $[0, 98)$ are selected after softmax without renormalization. Clean/attacked forwards share noise, and values average within task then over tasks.

Table 13 gives the complete layerwise summary. For joint K+V, correlations with $\mathrm{AttackStrength} = 1 - \mathrm{SR}/100$ are 0.316 (D_K), 0.528 (D_V), 0.530 (D_A), and 0.332 ($\|\Delta_{KV}\|$). Figures 8 and 9 show where attention probabilities and attention-weighted visual-value contributions change across the four prespecified layers and three optimization objectives.

Prespecified mechanism cases. Four cases illustrate layer dependence. L21 is K-dominant (K/V SR 6.0/91.0%; mean $|\Delta A|$ 0.00511/0.00074); L29 is V-dominant (K/V/K+V SR 92.5/14.5/9.5%; contribution change 0.01003/0.08218/0.08192); and L17 is joint-beneficial (K/V/K+V SR 66.0/91.5/11.5%; mean $|\Delta A|$ 0.00207/0.00250/0.00534). L20 illustrates non-dominance: K-only reaches 0.0% SR ($\|\Delta K\| = 95.4$, $|\Delta A| = 0.00449$), whereas K+V reaches 91.5% ($\|\Delta K\| = 36.4$, $\|\Delta_{KV}\| = 117.2$, $|\Delta A| = 0.00251$). Thus, large internal displacement alone does not determine closed-loop failure.

Figures 6 and 7 show the pathway and cases. Case panels prioritize their relevant contrast: panel a omits K+V (3.5% SR; mean $|\Delta A| = 0.00390$), and panel d omits V-only (91.0% SR; mean $|\Delta A| = 4.75 \times 10^{-4}$). The omitted conditions are reported here and in the full spatial maps.

A.7 Probe100 continuous prediction and repeat stability

The shared component-ablation Probe100 Tail-AV scores correlate with later 5k/10k/20k AttackStrength at 0.344/0.521/0.508. Eight additional runs retain the same 30 candidates, task-balanced pairs, and 100-exposure budget; Table 14 reports predictive summaries and Top-k overlaps.

A.8 Construction cost

Costs cover probing, selection, and final training, excluding victim training, smoke tests, queueing, and rollout evaluation. FastWAM Top-3/4/5 cost 2.3297/2.3961/2.3540 GPU-h, each with the re-

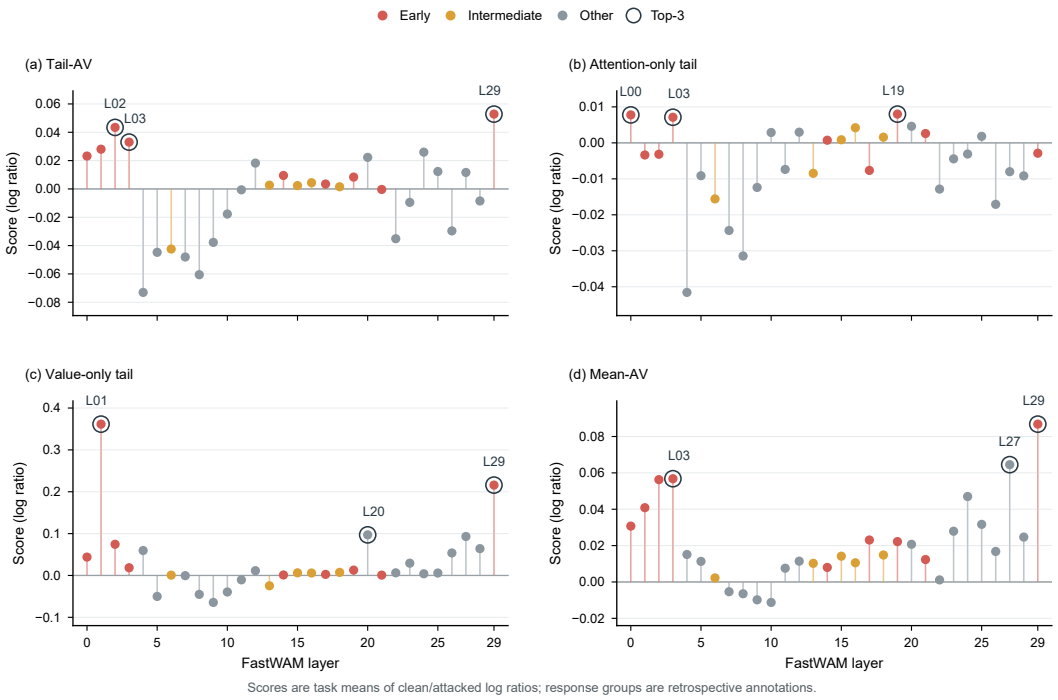


Figure 5: **Complete selector-score component rankings on the evaluated FastWAM checkpoint.** Thirty-layer rankings for Tail-AV, Attention-only-tail, Value-only-tail, and Mean-AV from the shared Probe100 component-ablation run. Colors denote retrospective Early, Intermediate, and Other groups; rings mark each ranking's Top-3.

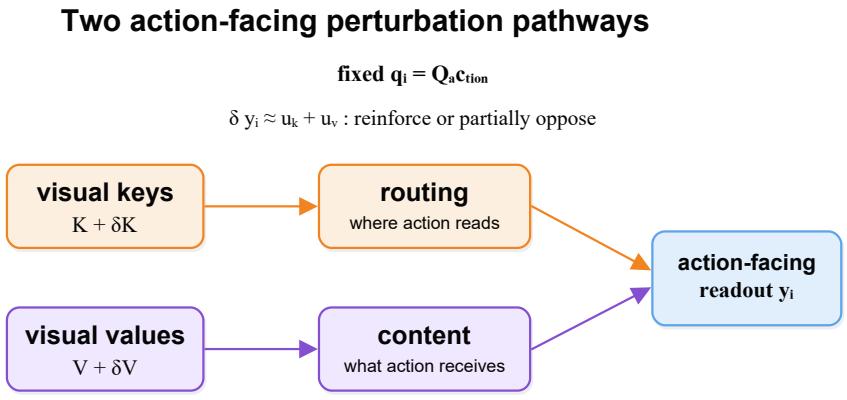


Figure 6: **Action-facing visual K/V pathways.** A fixed action query reads visual keys through attention and aggregates visual values into the action-facing readout. Key and value perturbations can alter routing and received content, respectively.

tained probe charge (Table 15). Against the largest baseline total of 10.746293 GPU-h (no-PSM UPA-RFAS), these are 77.7–78.3% lower. These are offline construction costs, not application latency.

Table 13: **Complete 30-layer K/V mechanism summary.** All rows use the joint K+V sweep with 50 observations from ten task-balanced tasks. D_K, D_V, and D_A are representation/readout displacement summaries; AS is $1-\mathrm{SR}/100$. The decomposition and heatmaps provide per-observation detail.

Layer	SR (%)	AS	D_K	D_V	D_A	$\|\Delta_K\|$	$\|\Delta_V\|$	$\|\Delta_{KV}\|$	Tail-AV
L00	1.5	0.985	0.377	1.433	0.429	9.4	105.8	112.9	0.425
L01	30.5	0.695	0.646	2.241	0.723	33.2	331.8	327.5	0.760
L02	10.0	0.900	0.553	1.967	0.566	37.5	324.7	310.8	0.474
L03	15.5	0.845	0.561	1.452	0.587	37.3	222.0	205.9	0.291
L04	95.5	0.045	0.565	1.357	0.687	58.3	214.6	208.1	0.187
L05	94.5	0.055	0.736	1.599	0.746	63.9	260.9	272.5	0.371
L06	46.5	0.535	0.363	1.088	0.432	39.7	178.5	178.6	-0.202
L07	87.0	0.130	0.401	0.877	0.370	41.4	172.1	171.4	-0.081
L08	81.0	0.190	0.437	1.011	0.525	50.3	201.1	203.5	-0.345
L09	91.0	0.090	0.521	0.926	0.353	42.5	192.4	194.4	-0.053
L10	87.0	0.130	0.412	1.054	0.351	45.2	236.8	238.7	-0.117
L11	84.0	0.160	0.339	0.776	0.395	54.3	160.3	165.5	-0.020
L12	81.0	0.190	0.429	0.786	0.448	59.6	149.1	153.4	0.013
L13	44.0	0.560	0.532	0.905	0.698	86.4	157.3	163.6	-0.138
L14	16.5	0.835	0.510	1.072	0.690	72.0	162.5	157.0	-0.115
L15	41.0	0.590	0.477	1.080	0.609	80.1	160.5	162.2	-0.032
L16	62.5	0.375	0.393	0.794	0.457	51.3	96.9	109.5	0.240
L17	11.5	0.885	0.643	1.038	0.906	73.9	164.4	160.9	0.062
L18	39.0	0.610	0.389	0.997	0.374	49.2	189.9	183.1	-0.114
L19	6.5	0.935	0.525	1.119	0.859	104.0	209.8	206.8	-0.037
L20	91.5	0.085	0.230	0.731	0.574	36.4	121.1	117.2	0.047
L21	3.5	0.965	0.483	0.809	0.571	74.8	142.1	153.4	0.036
L22	92.0	0.080	0.245	0.570	0.217	39.5	59.8	66.3	-0.010
L23	90.5	0.095	0.269	0.614	0.120	28.7	30.8	56.7	-0.006
L24	89.0	0.110	0.277	0.675	0.357	60.0	61.3	125.2	0.203
L25	92.0	0.080	0.139	0.695	0.108	21.0	18.8	40.9	0.020
L26	90.5	0.095	0.217	0.697	0.268	40.5	39.5	77.3	0.022
L27	89.5	0.105	0.237	0.907	0.210	26.3	33.4	55.3	0.100
L28	92.0	0.080	0.236	0.743	0.128	14.0	23.9	29.4	0.010
L29	9.5	0.905	0.213	2.810	0.793	30.6	318.6	310.9	2.063

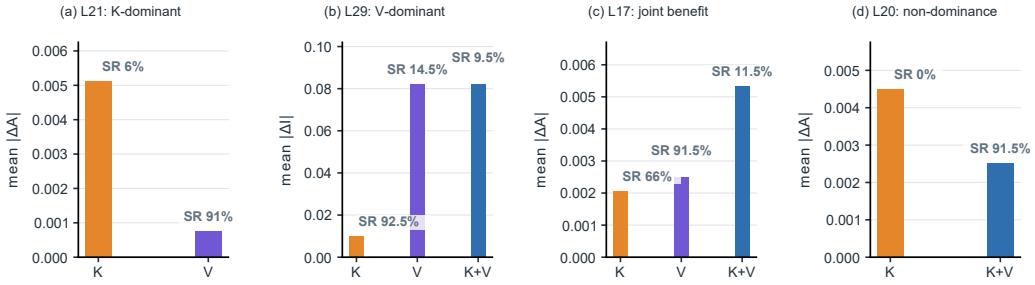


Figure 7: **Prespecified counterfactual mechanism cases on the evaluated FastWAM checkpoint.** **(a)** L21, K-dominant. **(b)** L29, V-dominant. **(c)** L17, joint-beneficial. **(d)** L20, joint non-dominance. Bars show case-specific objective contrasts, not all three objectives in every panel; omitted conditions are reported in the text. Labels report singleton Policy SR.

B Complete layerwise and PASA-KV results

B.1 Architecture-wide component sweeps

This section reports pooled and complete layerwise component results.

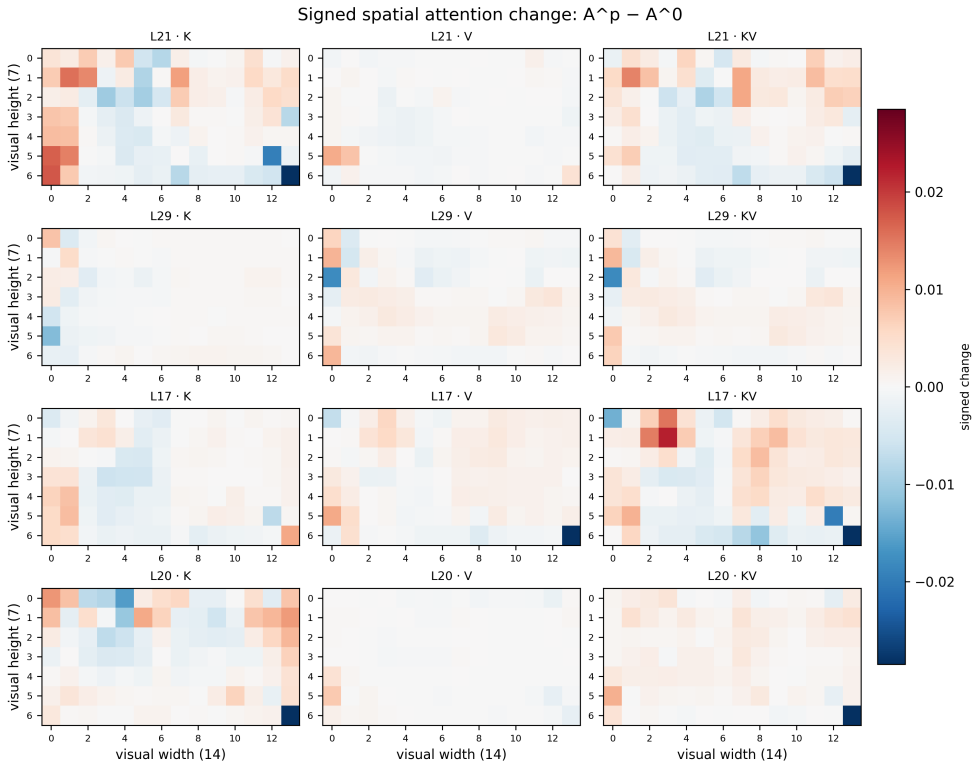


Figure 8: **Spatial changes in action-facing attention on the evaluated FastWAM checkpoint.** Each panel shows the task-macro signed mean of $\Delta A_{ij} = A^p_{ij} - A^0_{ij}$, averaged over action queries and heads and mapped to the 7×14 visual-token grid after full-key softmax. Rows show L21/L29/L17/L20; columns show patches trained with K-only, V-only, and joint K+V objectives. K-only and V-only denote training objectives, not isolated cache-tensor interventions. Red/blue indicate increases/decreases.

Table 14: **Probe100 probe-seed stability across eight additional runs on the evaluated FastWAM checkpoint.** Each repeat uses the fixed scoring pairs and a complete 30-layer ranking; values report mean ± sample SD and observed range. The canonical ranking is used only as the Top-k set reference.

Quantity	Mean ± SD	Observed range
Spearman ρ to 5k AttackStrength	0.3550 ± 0.0074	$[0.3465, 0.3655]$
Spearman ρ to 10k AttackStrength	0.5198 ± 0.0080	$[0.5067, 0.5297]$
Spearman ρ to 20k AttackStrength	0.5065 ± 0.0084	$[0.4953, 0.5194]$
Canonical Top-3 overlap	0.9583 ± 0.1179	$[0.6667, 1.0000]$
Canonical Top-5 overlap	1.0000 ± 0.0000	$[1.0000, 1.0000]$

Attention-map objectives. The attention-map columns are architecture-specific map-displacement controls. FastWAM normalizes the action-to-first-frame spatial map and maximizes its clean–attacked center-of-mass distance, $\|\sum_j (a^p_j - a^0_j) r_j\|_2$, with zero action latents and video/action timesteps. LingBot uses first-action-step attention over renormalized current-frame agent-view columns; Cosmos uses action-slot-4 queries over current-primary slot 3. Their score is $\|A^p - A^0\|_F / \max(\|A^0\|_F, 10^{-6})$. Each singleton objective uses detached clean targets.

The matched 10k sweep has 352 rows across four objectives and 30/30/28 FastWAM/LingBot/Cosmos layers. The 112 historical 5k C/A/W/J rows supplement the architecture-specific sweep. Pooling sums successes and trials over layers; Table 20 reports paired statistics and the descriptive 50% SR threshold.

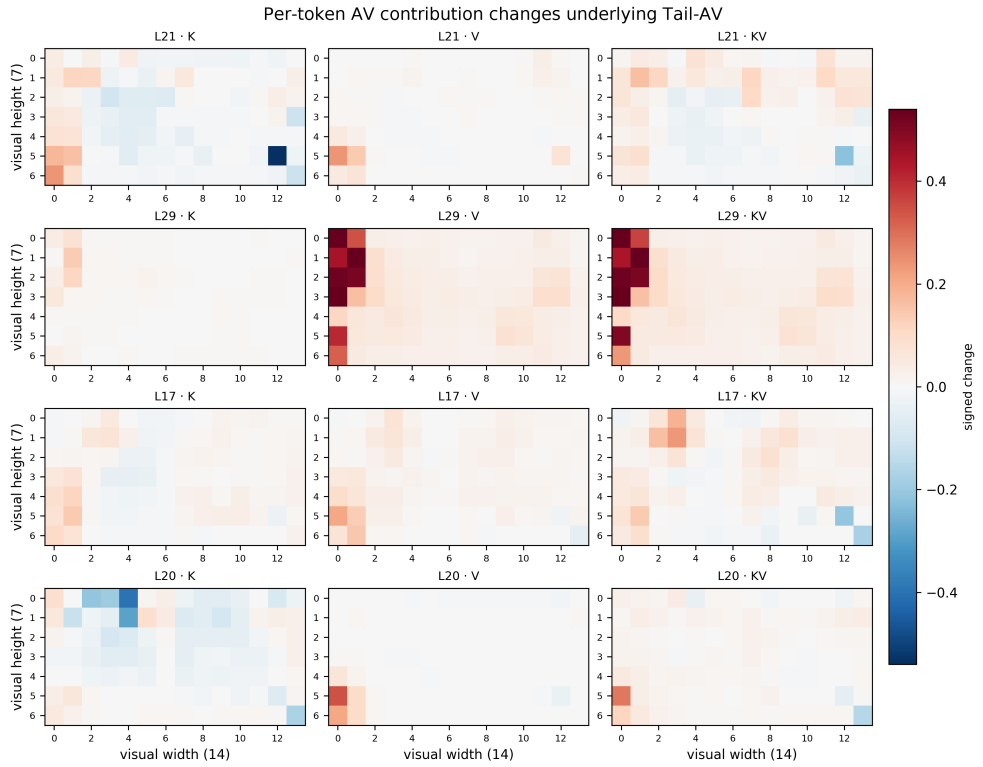


Figure 9: **Spatial changes in action-facing visual contribution on the evaluated FastWAM checkpoint.** Each panel shows the signed task-macro mean over 50 observations of $\Delta I_{ij} = A^p_{ij}\|V^p_j\|_2 - A^0_{ij}\|V^0_j\|_2$, averaged over action queries and heads on the 7×14 visual-token grid. Rows show L21/L29/L17/L20; columns show K-only, V-only, and joint K+V optimization objectives. K-only and V-only name training objectives, not isolated cache-tensor interventions. Red/blue indicate increases/decreases. These maps complement the quantitative case summary in Figure 7.

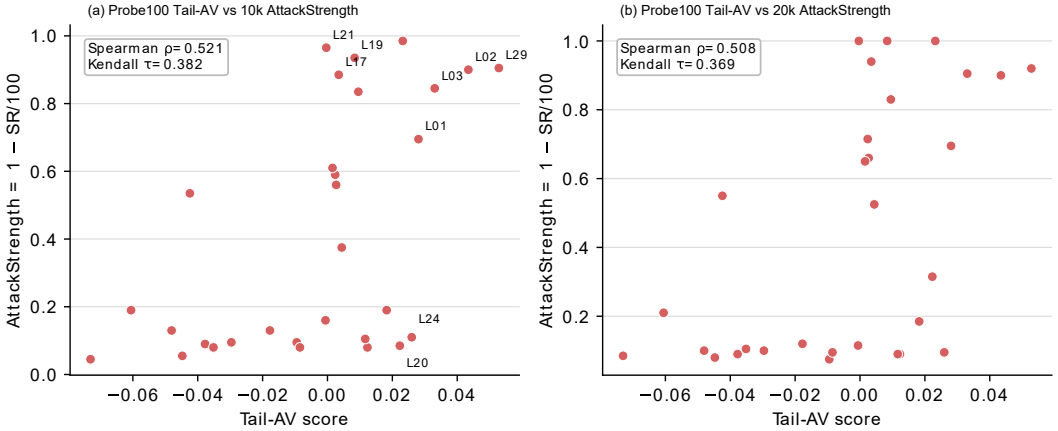


Figure 10: **Probe100 provides later-vulnerability ordering evidence.** Each point is a FastWAM layer: Tail-AV from the shared 100-exposure component-ablation score ledger versus later AttackStrength $= 1 - \mathrm{SR}/100$ from the matched BS=1 singleton patches at 10k or 20k exposures.

Table 15: FastWAM Tail-AV Top-3 construction cost (GPU-h). The total includes probing, selection, and final patch training.

Method	Probe	Full construction
Tail-AV Top-3	0.568632	2.3297

Table 16: **All-layer component-objective attacks on the evaluated FastWAM checkpoint.** Values are pooled closed-loop Policy SR (%; lower is stronger) over independently trained single-layer patches. K-only and V-only denote end-to-end patch optimization targets. Bold marks the lowest SR.

Model	Attn.-map	K only	V only	K+V
FastWAM	90.68	65.77	66.43	**58.87**

Protocol. All objectives use BS1 and 10k exposures per patch; each value pools 30 layers × 200 episodes (6,000). The complete three-model summary and layerwise values appear in Appendix B.1 (Table 17).

B.2 Exposure responses and selected multi-layer attacks

Post-hoc reporting groups. The three-point trajectories define Early by $\mathrm{SR}_{10k} \leq 35$ and $|\mathrm{SR}_{20k} - \mathrm{SR}_{10k}| \leq 8$ percentage points; delayed-drop by $\mathrm{SR}_{5k}, \mathrm{SR}_{10k} \geq 80$, $\mathrm{SR}_{10k} - \mathrm{SR}_{20k} \geq 15$, and $\mathrm{SR}_{20k} < 80$; and stable-high by a three-point minimum of 78 and $|\mathrm{SR}_{5k} - \mathrm{SR}_{20k}| \leq 12$. Remaining trajectories are Intermediate.

The display contains nine Early, five Intermediate (L06/L13/L15/L16/L18), and 16 Other targets; Other includes 15 stable-high trajectories and delayed-drop L20.

The Tail-AV-selected layer sets, listed in selected order for Top-3/4/5, are FastWAM: L29/L02/L03, L29/L02/L03/L01, and L29/L02/L03/L01/L24; LingBot-VA: L21/L02/L01, L21/L02/L01/L03, and L21/L02/L01/L03/L27; and Cosmos Policy: L14/L10/L01, L14/L10/L01/L26, and L14/L10/L01/L26/L19. Each set is trained as a separate fresh joint patch.

B.3 Footprint and visual-state replacement

Independent all-target remeasurement. Thirty targets use 5k/10k/15k/20k patches from one run per target (seed 20260801), giving $30 \times 4 \times 10 \times 30 = 36{,}000$ observed-layer records (1,200 paired conditions). BF16 forwards use tasks 0–9, frame 10, the documented frame-0 primary patch, and all 30 cache layers.

Footprint definition and aggregation. For target ℓ, training step s, task t, and observed layer m, let $d^K_{\ell stm}$ and $d^V_{\ell stm}$ be the clean-normalized tensor L2 displacements. Differences are evaluated in FP32 with clean norms clamped below at 10^{-12}. We compute

$$F_\ell(s) = \sum_{m=0}^{29} \frac{1}{10} \sum_{t=0}^{9} \frac{d^K_{\ell stm} + d^V_{\ell stm}}{2}, \qquad R_\ell(s) = \frac{F_\ell(s)}{F_\ell(5\mathrm{k})}. \tag{12}$$

Table 17: **All-layer component-objective attacks across three WAMs.** One evaluated checkpoint per architecture; values are pooled closed-loop Policy SR (%; lower is stronger) over independently trained single-layer patches. K-only and V-only denote end-to-end patch optimization targets. Bold marks the lowest SR among matched-10k cells in each architecture.

Architecture	Attn.-map	K only	V only	K+V
FastWAM	90.68	65.77	66.43	**58.87**
LingBot-VA	**92.45**	93.37	92.63	93.18
Cosmos Policy	91.34	89.91	**89.79**	90.04

Protocol. All objectives use BS1 and 10k exposures per independent patch. FastWAM/LingBot pool 30 layers × 200 episodes (6,000); Cosmos pools 28 × 200 (5,600), with K+V targeting current-primary slot 3 at joint=1. Objective definitions and full values: Appendix B.1; separate 5k C/A/W/J-KV tests: Appendix D.2.

Table 18: **Complete layerwise component attacks for the evaluated FastWAM and LingBot-VA checkpoints.** Each independently trained singleton uses BS1, 10k exposures and 200 evaluation episodes; values are Policy SR (%). The FastWAM attention-map controls record training seed 20260801; other cells retain the formal protocol label 42 where reported.

	FastWAM					LingBot-VA			
Layer	Attn.	K	V	K+V	Layer	Attn.	K	V	K+V
L00	89.5	83.5	31.5	1.5	L00	94.0	96.0	93.0	91.5
L01	87.5	31.5	28.0	30.5	L01	90.0	93.5	90.0	89.5
L02	91.0	9.5	10.0	10.0	L02	95.0	93.0	92.0	95.0
L03	91.5	87.5	6.0	15.5	L03	90.0	94.0	95.0	94.0
L04	91.5	67.5	91.0	95.5	L04	93.0	94.0	90.5	93.0
L05	93.0	94.0	89.5	94.5	L05	89.0	95.5	89.5	91.5
L06	90.5	79.0	46.5	46.5	L06	90.5	95.0	92.0	94.0
L07	92.0	89.5	92.5	87.0	L07	92.0	91.5	95.0	94.0
L08	90.5	73.0	94.0	81.0	L08	93.0	93.5	92.5	95.0
L09	92.0	91.0	91.5	91.0	L09	94.5	95.0	93.5	93.5
L10	88.0	88.0	89.5	87.0	L10	91.5	95.5	90.5	94.0
L11	92.0	80.5	82.0	84.0	L11	94.0	93.0	95.0	92.5
L12	91.5	85.5	88.0	81.0	L12	91.0	91.0	92.0	92.0
L13	89.5	80.0	25.5	44.0	L13	92.0	88.0	92.0	92.5
L14	91.0	68.0	22.0	16.5	L14	94.0	91.0	90.5	94.5
L15	87.0	64.0	39.0	41.0	L15	91.5	92.0	89.0	93.5
L16	93.0	68.5	87.5	62.5	L16	92.5	92.0	91.5	93.5
L17	90.0	66.0	91.5	11.5	L17	94.0	94.0	93.0	93.5
L18	90.5	7.0	38.0	39.0	L18	92.5	92.0	92.5	92.5
L19	94.0	9.5	10.5	6.5	L19	93.5	95.0	94.0	94.0
L20	89.0	0.0	91.0	91.5	L20	91.0	94.5	92.0	91.0
L21	93.0	6.0	91.0	3.5	L21	90.5	91.0	94.0	95.0
L22	89.0	88.5	93.0	92.0	L22	94.0	92.0	92.0	93.0
L23	94.0	10.5	91.5	90.5	L23	92.0	97.5	94.5	92.0
L24	87.0	91.0	90.5	89.0	L24	93.5	96.5	95.5	94.0
L25	88.0	90.0	95.0	92.0	L25	92.5	91.0	92.5	92.5
L26	90.0	90.5	88.0	90.5	L26	92.0	93.5	91.5	94.5
L27	91.5	91.0	92.5	89.5	L27	93.5	93.5	96.0	94.0
L28	92.5	90.0	92.0	92.0	L28	93.5	91.5	93.0	94.0
L29	91.0	92.5	14.5	9.5	L29	93.5	95.5	95.0	92.0
Pooled	90.68	65.77	66.43	58.87	Pooled	92.45	93.37	92.63	93.18

All 5k denominators are positive, and class means equally average target-normalized ratios. Tables 22 and 23 report all values; these checkpoints do not support a footprint–SR correlation claim.

Target coverage and descriptive groups. Figure 11(b) plots all targets and class means; L03/L13/L08 are closest to their group means. Mean 20k/5k ratios are 1.4390/1.1220/1.0975, spanning 0.9086–2.4966.

Response definition and aggregation. Rescue and Transfer measure normalized projected action change toward the clean and attacked action, respectively. Four fixed samples and then 30 targets are averaged to obtain g_ℓ, and $q_\ell = \max(0, g_\ell)/\sum_m \max(0, g_m)$. Each budget has 7,200 offline intervention records; their reference Action-L2 rises from 0.7467 to 1.2499 across 5k/20k.

Online oracle pilot. Table 24 uses 10k singleton patches and 50 episodes per condition. An oracle substitutes visual K/V from paired clean/attacked caches at the same simulator state, then executes the mixed-cache action. L25 reuses weak-attack set A (L17–L19); unlike Guard or offline Transfer, this pilot needs clean test-time access.

Table 19: **Complete layerwise component attacks for the evaluated Cosmos Policy checkpoint.** The first four columns are matched BS=1, formal 42-labeled, joint=1, 10k-exposure objectives; K+V targets current-primary slot 3. The last four columns report historical 5k C/A/W/J objectives. Every cell uses 200 episodes; Table 17 summarizes the matched 10k columns.

	Matched 10k component objectives				Historical 5k target-scope objectives			
Layer	Attn.-map	K only	V only	K+V	C-KV	A-KV	W-KV	J-KV
L00	89.5	92.0	92.0	93.0	92.0	90.5	92.5	91.5
L01	92.5	89.0	92.0	91.0	90.5	90.5	94.0	93.0
L02	95.0	88.5	88.0	88.5	89.5	91.0	91.5	89.5
L03	91.0	91.0	91.0	88.5	90.5	91.0	90.0	90.0
L04	90.5	90.0	89.5	89.0	87.5	92.5	90.5	91.0
L05	92.5	89.5	89.5	89.0	88.5	92.0	90.5	92.5
L06	93.5	90.0	89.5	90.5	90.0	87.0	87.0	88.5
L07	89.5	90.0	90.0	90.5	90.5	90.5	88.0	91.0
L08	90.0	91.5	91.0	91.5	90.5	91.0	89.5	92.5
L09	89.0	90.5	92.5	92.0	90.5	87.5	89.5	90.0
L10	87.5	91.5	91.0	91.0	90.5	91.5	91.0	88.5
L11	92.5	90.5	89.0	89.0	90.5	92.5	90.0	92.0
L12	92.0	89.0	90.0	91.0	90.0	91.5	92.5	92.0
L13	91.0	89.0	91.0	90.5	90.0	90.0	88.0	87.5
L14	94.0	90.5	91.5	90.0	89.5	92.0	93.0	90.5
L15	94.5	90.5	89.5	90.0	91.5	93.5	92.0	90.5
L16	92.5	87.0	87.0	87.5	90.5	90.0	91.0	86.5
L17	91.5	88.0	88.0	88.5	90.0	91.0	91.0	89.5
L18	89.5	90.0	90.0	90.0	89.5	88.0	89.0	89.5
L19	92.0	90.5	89.0	90.0	88.0	92.5	88.5	87.0
L20	91.5	89.0	89.0	90.5	89.0	87.5	88.5	90.5
L21	92.0	89.0	88.5	87.5	88.5	88.0	89.0	89.5
L22	88.5	90.0	93.0	92.5	89.0	92.0	88.5	91.5
L23	88.0	89.0	89.5	89.0	91.5	90.5	88.0	89.0
L24	95.0	91.0	87.5	87.0	90.0	92.0	91.5	89.0
L25	93.5	91.5	86.0	91.0	90.0	90.0	91.5	87.5
L26	89.0	90.5	89.5	91.5	90.5	90.5	90.0	89.0
L27	90.0	89.0	90.0	91.0	91.0	93.5	89.5	93.5
Pooled	91.34	89.91	89.79	90.04	89.98	90.71	90.20	90.09

Table 20: **Paired objective comparisons across the 30 layers of the evaluated FastWAM checkpoint.** Lower Policy SR is a stronger attack. "Strong attack" denotes the descriptive coverage threshold SR $\leq 50\%$. Percentile intervals resample the 30 evaluated layers 20,000 times with seed 20260905.

Paired layer statistic	Estimate	95% interval
K+V strictly stronger than K-only	17/30 (56.7%)	[40.0, 73.3]%
K+V strictly stronger than V-only	17/30 (56.7%)	[40.0, 73.3]%
Median $\mathrm{SR}_{K+V} - \mathrm{SR}_{K}$	-1.25 pp	$[-3.75, 0.50]$ pp
Median $\mathrm{SR}_{K+V} - \mathrm{SR}_{V}$	-1.00 pp	$[-3.50, 0.25]$ pp
Strong-attack layer coverage: K-only	7/30 (23.3%)	[10.0, 40.0]%
Strong-attack layer coverage: V-only	11/30 (36.7%)	[20.0, 53.3]%
Strong-attack layer coverage: K+V	13/30 (43.3%)	[26.7, 60.0]%
K+V retains strong attack in single-branch-asymmetric layers	8/10 (80.0%)	[50.0, 100.0]%

C BASELINE PROTOCOLS AND FIDELITY

Rationale and interpretation. Table 25 documents source-principle WAM adaptations, including geometry, training pools, and cost accounting; Table 2 isolates layer selection under a shared K/V objective and budget.

Table 21: Complete FastWAM singleton K/V exposure trajectories. Policy SR is in percent; labels are post-hoc reporting groups under the documented objective, exposure checkpoints, and rollout protocol.

Layer	5k	10k	20k	Response	Layer	5k	10k	20k	Response
L00	64.5	1.5	0.0	Early	L15	60.5	41.0	28.5	Intermediate
L01	34.5	30.5	30.5	Early	L16	92.0	62.5	47.5	Intermediate
L02	10.0	10.0	10.0	Early	L17	91.0	11.5	6.0	Early
L03	91.0	15.5	9.5	Early	L18	46.5	39.0	35.0	Intermediate
L04	92.5	95.5	91.5	Other	L19	38.5	6.5	0.0	Early
L05	92.5	94.5	92.0	Other	L20	90.0	91.5	68.5	Other
L06	46.0	46.5	45.0	Intermediate	L21	90.5	3.5	0.0	Early
L07	89.0	87.0	90.0	Other	L22	91.0	92.0	89.5	Other
L08	87.5	81.0	79.0	Other	L23	91.0	90.5	92.5	Other
L09	92.0	91.0	91.0	Other	L24	93.0	89.0	90.5	Other
L10	89.5	87.0	88.0	Other	L25	94.0	92.0	91.0	Other
L11	78.5	84.0	88.5	Other	L26	92.5	90.5	90.0	Other
L12	86.0	81.0	81.5	Other	L27	87.5	89.5	91.0	Other
L13	60.0	44.0	34.0	Intermediate	L28	91.0	92.0	90.5	Other
L14	42.5	16.5	17.0	Early	L29	40.0	9.5	8.0	Early

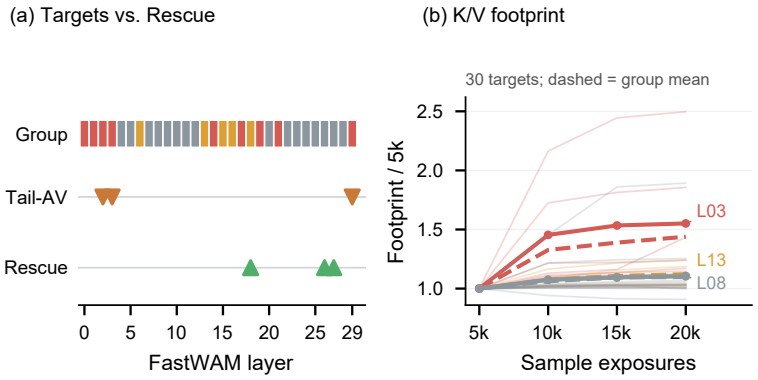


Figure 11: **Layerwise intervention diagnostics on the evaluated FastWAM checkpoint. (a)** Tail-AV-selected targets and Rescue peaks. **(b)** K/V-footprint growth relative to each target's 5k value. Colors denote retrospective response groups. Detailed protocols and analyses are provided in Appendices B.2 and B.3.

UPA-RFAS/UPA/TMA/UADA match PASA-KV's FastWAM geometry and retain 71.4–90.8% SR. Different objectives and observation pools make this a system-level comparison, not an isolated K/V-objective estimate. Costs are implementation-level workflow measurements (Appendix A.8).

LingBot DRIFT noise pairing and evaluation. During patch training, the clean and attacked velocity terms share noise per exposure. One-GPU BS1 training uses seed 42 and 10k updates (2.462206 GPU-h, excluding loading/evaluation); disjoint evaluation batches give $183/200 + 279/300 = 462/500$ successes.

UPA-RFAS contrastive reference set at batch size one. The WAM translation uses token positions, rather than different batch samples, as its contrastive reference set. For one image with P visual tokens, let c_u and a_v be unit-normalized clean (stop-gradient) and attacked features. The patch maximizes

$$J_{\text{tok}} = -\frac{1}{P}\sum_{u=1}^{P} \log \frac{\exp(c_u^\top a_u/\tau)}{\sum_{v=1}^{P}\exp(c_u^\top a_v/\tau)}. \tag{13}$$

Table 22: **Independent all-target exact-checkpoint K/V footprints.** Absolute task-mean footprint $F_\ell(s)$ and target-normalized 20k/5k ratio $R_\ell(20\text{k})$. All thirty targets use training seed 20260801, four checkpoints from the same run per target, ten paired tasks per point, and thirty observed layers. No historical SR is joined.

Target	Group	5k	10k	15k	20k	20k/5k
L00	Early	13.768	15.131	15.638	16.121	1.1709
L01	Early	15.310	15.577	15.678	15.722	1.0269
L02	Early	16.536	16.939	17.014	17.032	1.0300
L03	Early	10.294	14.971	15.792	15.969	1.5513
L04	Other	8.533	8.886	9.026	9.119	1.0687
L05	Other	7.459	9.079	9.279	9.353	1.2539
L06	Intermediate	13.619	13.996	14.155	14.197	1.0425
L07	Other	7.427	7.616	7.652	7.676	1.0336
L08	Other	10.386	11.161	11.363	11.479	1.1052
L09	Other	7.693	7.954	8.046	8.112	1.0544
L10	Other	8.723	9.536	9.737	9.836	1.1277
L11	Other	8.842	8.324	8.084	8.034	0.9086
L12	Other	7.834	8.044	8.057	7.811	0.9970
L13	Intermediate	9.555	10.266	10.482	10.567	1.1060
L14	Early	11.283	12.557	12.825	12.939	1.1468
L15	Intermediate	10.011	11.663	12.176	12.431	1.2418
L16	Intermediate	9.013	9.823	10.423	10.700	1.1872
L17	Early	9.596	16.557	17.405	17.807	1.8557
L18	Intermediate	7.536	7.699	7.757	7.782	1.0327
L19	Early	15.196	17.167	17.645	21.792	1.4341
L20	Other	7.645	11.105	14.222	14.471	1.8927
L21	Early	7.069	15.291	17.277	17.648	2.4966
L22	Other	6.736	6.777	6.752	6.762	1.0039
L23	Other	6.623	6.745	6.730	6.679	1.0085
L24	Other	6.371	6.415	6.417	6.417	1.0072
L25	Other	6.243	6.318	6.348	6.367	1.0200
L26	Other	6.291	6.398	6.487	6.531	1.0382
L27	Other	6.101	6.201	6.246	6.273	1.0283
L28	Other	6.291	6.321	6.355	6.364	1.0116
L29	Early	13.876	16.866	16.952	17.193	1.2390

Table 23: **Class summaries of target-normalized footprints.** Class means average $R_\ell(s)$ after normalization within each target. Groups are retrospective; SD uses $n - 1$ across target layers. All 5k ratios equal one.

		Class mean			20k across-layer distribution		
Group	n	10k	15k	20k	Median	SD	Range
Early	9	1.3269	1.3889	1.4390	1.2390	0.4795	[1.0269, 2.4966]
Intermediate	5	1.0758	1.1077	1.1220	1.1060	0.0910	[1.0327, 1.2418]
Other	16	1.0618	1.0937	1.0975	1.0309	0.2246	[0.9086, 1.8927]

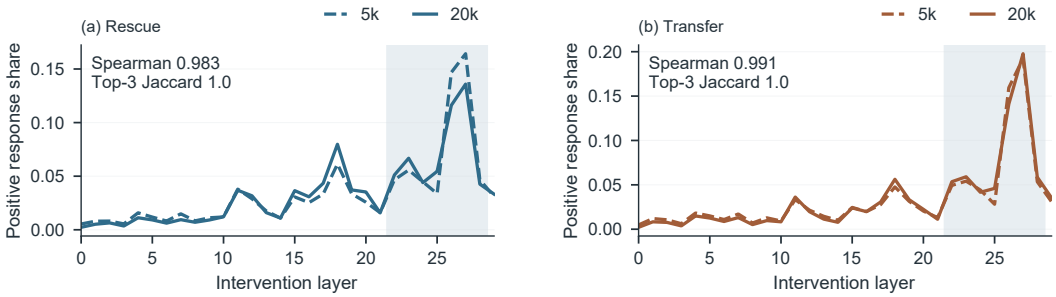


Figure 12: **Visual-state replacement profiles.** Each layer averages 30 target patches and four fixed samples (tasks 0/1; frames 0/10); negative gains are clipped before across-layer normalization. Shading marks L22–L28. Five-k/20k profile correlations are 0.983/0.991 with unchanged Top-3 sets.

Table 24: **Historical pilot of oracle K/V replacement.** Closed-loop Policy SR (%; higher means better control here), 50 episodes per cell. These 10k singleton patches provide supplementary mechanism evidence, separate from the main attack comparison and deployable Guard.

Attack target	Clean	Attack	Set A (3 layers)	Set B (3 layers)	A ∪ B (6)
L18	94	44	88	52	86
L29	96	16	26	32	58
L25 (weak control)	94	88	88	94	98

At every replan, selected visual K/V states are replaced by a clean pass at the same simulator state. Set A is L17–L19 for L18/L25 and L27–L29 for L29; B is L24–L26. This oracle uses clean observations and is not the deployable Guard. Each row uses its own paired task/trial grid (10 tasks × 5 trials).

Logits have shape $[B, P, P]$ with token-index labels, retaining P denominator terms at BS1. This token-reference contrast differs from sample-level contrast and uses action rather than source-model text queries.

C.1 Action discrepancy versus closed-loop success

Main-text Figure 3(a) and Table 26 remeasure the six primary baseline patches at a uniform runtime. Seed-42 patches share ten task-balanced observations with matched input, text, proprioception, and action noise; inference is BF16 with ten action steps and FP32 reduction. Two numerical repeats yield 120 records; repeat 0 is reported and Eval500 SR is joined by method and patch identity.

For normalized chunks $U_i^p, U_i^0 \in \mathbb{R}^{H\times d}$, $H = 32$, $d = 7$, the diagnostic is

$$D_{\text{chunk}} = \frac{1}{10}\sum_{i=1}^{10} \frac{\sum_{h=1}^{H} \|U_{i,h}^p - U_{i,h}^0\|_2}{\sqrt{Hd}}. \tag{14}$$

Cumulative sums within the predicted chunk, including the unexecuted suffix. Equation 15 uses RMS for the cross-replan diagnostic.

For the six heterogeneous baselines, full-precision values give Pearson $r = -0.956$ and Spearman $\rho = -0.714$; three of 15 method pairs invert Action-L2 and Policy-SR order. Thus, Action-L2 is informative but does not fully rank attacks across different objectives. Deployment and fixed-input checks appear in Appendices D.3 and D.5.

Table 25: **Baseline fidelity under our documented WAM adaptations.** Geometry and objectives are separated from budget accounting for readability. Areas refer to the affected resized view, not the complete input.

Method	Fidelity	Recorded geometry	Target / accessible representation
DP-Attacker	Source-principle adaptation	FastWAM: 45×45, main, 4.04%, EOT. Cosmos: 50×50, primary, 4.98%. LingBot: 29×29, agent, $\sim 5.13\%$.	Action-denoising target / diffusion surrogate; no explicit K/V intervention.
DRIFT	Source-principle adaptation; Cosmos EDM variant	FastWAM: 32×32, wrist, 2.04%, no EOT. Cosmos: 50×50, primary. LingBot: 18×18, wrist, $\sim 1.98\%$, at $(55, 110)$.	Initial flow/velocity for FastWAM and LingBot; joint denoised prediction for Cosmos.
UPA-RFAS	Conceptual adaptation; no PSM	FastWAM/Cosmos: 50×50, primary. LingBot: 29×29, agent.	Feature deviation, token contrastive repulsion, and action-query attention dominance; no explicit K/V intervention.
UPA	Conceptual adaptation	Same geometry as UPA-RFAS.	Continuous action-chunk objective.
TMA	Conceptual adaptation	Same geometry as UPA-RFAS.	Continuous action-chunk objective.
UADA	Conceptual adaptation	Same geometry as UPA-RFAS.	Continuous action-chunk objective.

Accounting field	Record and comparison boundary
Updates / exposures	All six: BS=1, seed 42, 10,000 optimizer updates and 10,000 sample exposures.
GPU-hour basis	GPU-active or card-wall construction records, excluding victim pretraining, smoke tests, queueing, and rollout evaluation. LingBot DRIFT records training-loop time only (2.462206 GPU-h), also excluding loading.
Observation pool	DRIFT uses its historical Spatial pool, whereas DP-Attacker uses a task-balanced pool.

FastWAM UPA-RFAS/UPA/TMA/UADA form the closer-geometry subset relative to PASA-KV; DP and DRIFT use different geometries. Table 1 gives SR and recorded costs.

Table 26: **Matched-noise Action-L2 for the six primary baseline patches.** Each D_{chunk} is the mean over ten fixed inputs using predefined numerical repeat 0. SR uses the same patch's full Eval500 outcomes from Table 1, not new rollouts. The diagnostic values belong to the independent uniform-runtime remeasurement, not the older diagnostic series.

Method	D_{chunk}	Success/episodes	Policy SR (%)
DP-Attacker	0.26948	405/500	81.0
DRIFT	0.49759	265/500	53.0
UPA-RFAS (no PSM)	0.08931	454/500	90.8
UPA	0.21314	387/500	77.4
TMA	0.23003	406/500	81.2
UADA	0.26127	357/500	71.4

Table 27: **Matched BS=1, 20k singleton Action-L2 and Policy SR across 30 FastWAM layers.** Each row is a separately trained single-layer joint K/V patch. Action-L2 is the mean unnormalized FP32 L2 norm of the full action-chunk difference over four fixed inputs (tasks 0/1, frames 0/10). Eval200 SR is matched by layer to Table 21.

Layer	Action-L2	Eval200 SR (%)	Layer	Action-L2	Eval200 SR (%)
L00	2.1042	0.0	L15	1.9247	28.5
L01	2.1661	30.5	L16	1.0540	47.5
L02	2.8538	10.0	L17	2.5732	6.0
L03	2.9605	9.5	L18	0.7030	35.0
L04	0.3385	91.5	L19	3.6645	0.0
L05	0.8464	92.0	L20	1.3762	68.5
L06	1.8924	45.0	L21	2.5344	0.0
L07	0.3268	90.0	L22	0.2111	89.5
L08	1.3173	79.0	L23	0.2490	92.5
L09	0.2765	91.0	L24	0.2514	90.5
L10	0.9184	88.0	L25	0.1863	91.0
L11	0.4562	88.5	L26	0.1964	90.0
L12	0.3653	81.5	L27	0.2019	91.0
L13	1.2415	34.0	L28	0.1749	90.5
L14	1.9948	17.0	L29	2.1620	8.0

D Cross-architecture diagnostics

D.1 Reduced-Step Clean-Performance Controls

LingBot/Cosmos use reduced-step solvers: one video step (endpoint $s = 0.6$) plus ten action steps, and one joint step, respectively. FastWAM retains its released configuration (Appendix A.1). Table 28 reports clean performance under these settings.

Table 28: **Reduced-step clean controls.** Each condition has 200 episodes; bracketed values are two-sided Wilson 95% intervals. Drops use the high-step reference.

System	Solver	Use	Success	Clean SR [95% CI]	Drop
LingBot-VA	v20/a50	High ref.	199/200	99.5% [97.22, 99.91]	—
LingBot-VA	v3/a10	Mid.	191/200	95.5% [91.67, 97.61]	−4.0 pp
LingBot-VA	v1/a10	Unified	187/200	93.5% [89.20, 96.16]	−6.0 pp
Cosmos	j5	High ref.	198/200	99.0% [96.43, 99.73]	—
Cosmos	j3	Mid.	191/200	95.5% [91.67, 97.61]	−3.5 pp
Cosmos	j1	Unified	194/200	97.0% [93.61, 98.62]	−2.0 pp

The selected settings retain 93.5%/97.0% clean SR, 6.0/2.0 points below their high-step references.

D.2 Historical target-scope stress tests

Table 29: **Historical target-scope stress tests.** Complete within-system singleton sweeps broaden the K/V objective while keeping RGB patch placement fixed; episode-pooled Policy SR (%).

LingBot-VA: 20k updates, 6,000 episodes/row

K/V objective	Video steps	Clean	Attack SR
Current	1	93.50	93.07
Current + future3	1	93.50	93.47
Current	3	95.50	96.13

Cosmos: 5k updates, 5,600 episodes/row

Objective	Clean	Attack SR
C-KV: conditioning	97.00	89.98
A-KV: action-facing	97.00	90.71
W-KV: future	97.00	90.20
J-KV: joint objective	97.00	90.09

LingBot: 30 layers × 200 episodes, ten action steps; "future3" extends the K/V loss, not patch placement. Cosmos: one joint step, 28 layers × 200 episodes; distinct from the matched-10k component ablation. Native clean: 200 episodes. Full protocols: Appendix D.2.

Table 29 contains complete singleton sweeps. LingBot uses 20k updates, a 29×29 current-frame patch, video endpoint $s = 0.6$, and ten action steps: one-step current/current-plus-future3 give 5,584/6,000 and 5,608/6,000 successes; three-step current-only gives 5,768/6,000. Cosmos uses one joint step, a current-primary 50×50 patch, and 5k updates. C/A/W/J target conditioning, action-facing, future, and joint slots; each has 28 layers and 5,600 episodes, with 5,039/5,080/5,051/5,045 successes.

D.3 FastWAM cross-replan deployment diagnostic

E1 uses the primary Top-3 patch (L29/L02/L03) on LIBERO-10. Trial 0 supplies matched initial states across clean, one-shot, and persistent conditions (environment/policy seed 42). One-shot patches R0; persistent deployment patches each current observation. The diagnostic has 30 trajectories and 1,325 replans.

Local versus closed-loop action discrepancy. Let $U_{tr}^{c} \in \mathbb{R}^{32\times 7}$ be the full normalized action chunk on condition c's trajectory for task t, replan r, and let $\widetilde{U}_{tr}^{c}$ be the unpatched paired forward at that same current state. Let U_{tr}^{0} be the actual action chunk from the separately executed clean branch at the same replan index. We report

$$d_{tr,c}^{\text{local}} = \frac{\|U_{tr}^{c} - \widetilde{U}_{tr}^{c}\|_F}{\sqrt{32 \cdot 7}}, \qquad d_{tr,c}^{\text{loop}} = \frac{\|U_{tr}^{c} - U_{tr}^{0}\|_F}{\sqrt{32 \cdot 7}}. \tag{15}$$

Figure 3(b) uses full-chunk normalized-action RMS, including the unexecuted suffix; the first 10×7 denormalized entries are executed.

Table 30: Corrected FastWAM E1 outcomes. One fixed initial state per LIBERO-10 task and ten trajectories per condition.

Deployment	Successes	Policy SR (%)	Failed tasks
Clean	10/10	100	None
One-shot	8/10	80	T4, T9
Persistent	0/10	0	All ten

Table 31: FastWAM E1 full-chunk normalized-action RMS discrepancy. Each entry is a macro mean over the same ten tasks. Local and closed-loop comparisons follow Eq. 15.

Comparison	Condition	R0	R1	R2	R3	R4
Local	Clean	0.001883	0.001962	0.001799	0.003379	0.001816
	One-shot	0.231217	0.001864	0.001829	0.001685	0.002762
	Persistent	0.231149	0.233908	0.231584	0.224538	0.235500
Closed-loop	Clean	0	0	0	0	0
	One-shot	0.232862	0.156301	0.234456	0.286428	0.278348
	Persistent	0.232465	0.261577	0.332843	0.435893	0.523264

Each R0–R4 point averages ten tasks; Table 31 distinguishes one-shot local recovery from persistent disruption and trajectory divergence.

D.4 LingBot fixed-input history and cache-aware attack

For fixed Tail-AV Top-3 inputs, increasing history length from R1 to R15 decreases mean relative K/V-L2 (0.3658/0.3519/0.3380/0.3276/0.3198 at R1/R2/R4/R8/R15), whereas normalized Action-L2 is non-monotonic (0.2649/1.0401/0.9335/0.5250/0.6796; Figure 3(c)). The action response peaks at R2, so the video-side cache displacement and action response do not move in lockstep. Relative K/V-L2 is normalized by the clean representation and does not isolate changes in the perturbation numerator or token span.

A separate cache-aware L05 patch backpropagates through R1–R4 history for 10k exposures. In this condition, clean and attacked SR are 94% and 82%; Task 4 accounts for 18 additional failures. This is a task-concentrated attack result, distinct from the fixed-input history-length diagnostic.

D.5 COSMOS FIXED-PATCH INPUT SENSITIVITY

Figure 3(d) uses the fixed 10k Top-5 patch (L14/L10/L01/L26/L19). Nine conditions and one input per task give 90 shared-noise pairs.

For normalized action chunks $a^p, a^0 \in \mathbb{R}^{16\times 7}$, the metric is

$$D_{\text{act}} = \frac{1}{\sqrt{16\cdot 7}} \sum_{t=1}^{16} \left\| (a_t^p - a_t^0)/\sigma_{\text{action}} \right\|_2, \qquad \sigma_{\text{action}} = \frac{2\,\text{std}(a)}{a_{\max} - a_{\min}}. \tag{16}$$

All seven dimensions use dataset statistics and equal task averaging. This sigma-normalized cumulative reduction differs from FastWAM RMS.

Table 32: Fixed Cosmos Tail-AV Top-5 input sensitivity. Ten-task mean sigma-normalized cumulative Action-L2 (Eq. 16); ten paired inputs per condition, 90 total.

View	Patched frames R	Joint steps S	Action-L2
Primary	1	1	0.131475
Primary	2	1	0.163918
Primary	4	1	0.325626
Wrist	4	1	0.414439
Both	4	1	0.649395
Primary	4	3	0.348381
Primary	4	5	0.349069
Both	4	3	0.824799
Both	4	5	0.826553

With one joint step, patching one versus four primary-view frames raises D_{act} from 0.131 to 0.326; at four frames, wrist-only and both-view patching give 0.414 and 0.649 (Table 32). Increasing joint steps from one to three or five changes the primary-only four-frame response only to 0.348/0.349, whereas the both-view response reaches 0.825/0.827. Within these fixed inputs, patched view/frame coverage matters more than extra joint steps for the primary-only condition. These are action-response diagnostics, not additional closed-loop SR measurements.

Here R counts patched current-slot frames (not history length), S counts Hybrid-EDM joint steps, and diagnostic noise uses base seed 42 with per-input seed $4200000 + \text{sample_index}$.

E PASA-KV GUARD

E.1 LAYER SELECTION AND CLEAN CALIBRATION

Thirty singleton Probe1000 patches are scored on ten paired task inputs with shared clean/attack noise. For normalized chunks $a_{\ell t}^p, a_t^0 \in \mathbb{R}^{H\times D}$,

$$s_\ell^{\text{Guard}} = \frac{1}{10} \sum_{t=0}^{9} \frac{\left\| (a_{\ell t}^p - a_t^0)/\sigma_{\text{action}} \right\|_2}{\sqrt{HD}}. \tag{17}$$

Here $H = 32$, $D = 7$, and σ_{action} is the dataset's per-dimension global standard deviation mapped through the min/max normalizer. Descending scores select L02/L01/L06/L00/L12 independently of Tail-AV; All30 and Top5 thus test broad and targeted mitigation. One clean trajectory per task sets thresholds: the 99.5th percentile is computed within task and then averaged, giving $C_{\ell h}$ for visual-attention dominance and $T_{\ell h}$ for local contribution.

Table 33: Frozen high-response candidates selected from all 30 FastWAM layers by normalized paired Action-L2. Each layer has ten task-balanced paired records from an equal-budget Probe1000.

Rank	Layer	Normalized paired Action-L2	95% bootstrap CI
1	L02	0.47284	$[0.36650, 0.58085]$
2	L01	0.25876	$[0.19552, 0.33376]$
3	L06	0.12833	$[0.09933, 0.15727]$
4	L00	0.09233	$[0.07501, 0.11057]$
5	L12	0.08447	$[0.06727, 0.10072]$

E.2 GUARD EQUATIONS

At each active layer/head, Guard applies the same token-wise rule; Targeted uses five selected layers and All30 uses all layers. With $D_j = \max_i A_{ij}$,

$$g_j^K = \sigma((D_j - C_{\ell h})/\tau)\,, \qquad K'_j = K_j(1 - \lambda_K g_j^K). \tag{18}$$

It recomputes attention with K'_j, compresses the visual tail, and normalizes each query row:

$$\widehat{A}_{ij} = C_{\ell h}\tanh(A'_{ij}/C_{\ell h}), \qquad \overline{A}_{ij} = \widehat{A}_{ij}/\sum_k \widehat{A}_{ik}. \tag{19}$$

The bound $\widehat{A}_{ij} \le C_{\ell h}$ holds before, not after, row normalization; Guard compresses tails rather than imposes a hard final cap. Finally, with $I'_{ij} = \overline{A}_{ij}\|V_j\|_2$,

$$g_{ij}^V = \sigma\big((I'_{ij} - T_{\ell h})/\tau\big)\,, \qquad y_i^{\text{guard}} = \sum_j \overline{A}_{ij}(1 - \lambda_V g_{ij}^V)V_j. \tag{20}$$

We use $\tau = 0.02$ and $\lambda_K = \lambda_V = 0.50$ in `kv_joint` mode. The differentiable rule uses no patch coordinates, position prior, or attack label, and is disabled unless activated.

E.3 FIXED TAIL-AV PATCH RESULTS

Table 34 reports pooled results and Table 35 the three 200-episode fixed patches: L29/L02/L03, then L01, then L24. Targeted Guard protects all 24 heads at L02/L01/L06/L00/L12 under fixed calibration.

Table 34: **Pooled Guard results on fixed Tail-AV attacks.** SR and recovery are percentages. Attacked SR pools three patches (600 episodes); clean SR uses 200 episodes per deployment. Recovery is the fraction of the attack-induced SR loss restored, relative to the 93.5% undefended clean reference. Per-patch results: Appendix Table 35.

Deployment	Clean SR	Attacked SR	Recovery
No defense	93.50	1.33	–
Guard (30 layers)	53.00	49.67	52.44
Guard targeted (5)	93.50	87.50	93.49

The recovery fraction $(\text{SR}_{\text{guard}} - \text{SR}_{\text{attack}})/(93.5 - \text{SR}_{\text{attack}})$ is 52.44% for All30 and 93.49% for Targeted5. Targeted deployment improves both clean and attacked SR.

Separate Eval200 clean controls give All30 53.0% (106/200; Wilson 95% CI 46.09–59.80%) and Targeted5 93.5% (187/200; 89.20–96.16%). Clean controls are counted once across the patch family; calibration trajectories are excluded.

Table 35: **Per-patch Guard results.** Policy SR (%) and success counts, with 200 episodes per patch/deployment. No-defense results use the fixed patch family evaluated for Guard.

	No defense		Guard (30)		Guard targeted (5)	
Input	SR	Count	SR	Count	SR	Count
Clean	93.50	187/200	53.00	106/200	93.50	187/200
Tail-AV Top-3	0.00	0/200	52.50	105/200	85.50	171/200
Tail-AV Top-4	0.00	0/200	40.50	81/200	89.00	178/200
Tail-AV Top-5	4.00	8/200	56.00	112/200	88.00	176/200
Pooled attacks	1.33	8/600	49.67	298/600	87.50	525/600

F REAL-ROBOT DEPLOYMENT AND RESULTS

Platform and patch protocol. We tested update-10,500 FastWAM on a 6-DoF AgileX Piper with head and wrist RGB cameras. Its four-task suite covers pen-in-cup, carrot-in-bowl, second-drawer placement and closing, and bowl stacking. Figure 13 pairs each initial scene with an annotated representative frame from its provided ground-truth rollout; these panels illustrate approximate task progress rather than complete trajectories. The 224×224 views were concatenated side by side; at each replan, the digital patch was overlaid on the current head view while the wrist view remained unchanged.

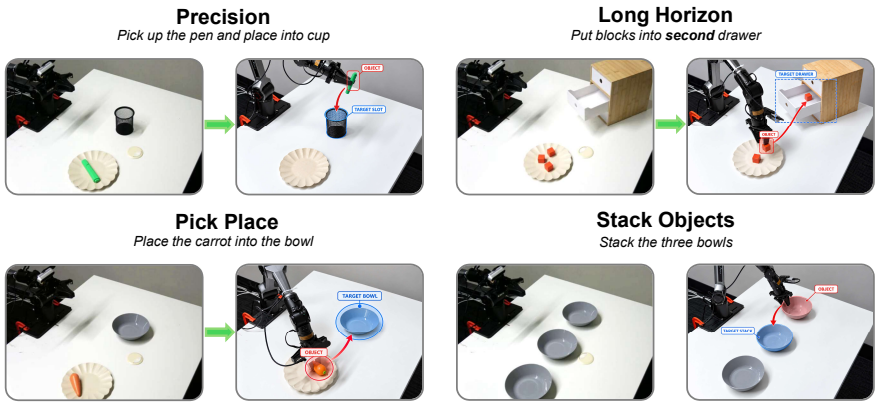


Figure 13: **Piper FastWAM task scenes and reference rollouts.** Each pair shows the initial scene (left) and a representative annotated frame from the provided ground-truth rollout (right). These frames give an approximate illustration of task progress, not complete trajectories. The manipulated object is marked in red and its target slot or placement site in blue. Tasks are pen-in-cup, second-drawer placement, carrot-in-bowl, and bowl stacking; closed-loop attack results appear in Table 3.

PASA-KV construction. Probe100 scored all 30 video layers using task-balanced samples and one held-out input per task. Top-3 selected L04/L27/L00; Top-5 added L02/L01. For each Top-k, one fresh patch was trained with four-task-balanced samples and shared across tasks at deployment. The 50×50 RGB patches used 10,000 BS1 AdamW ascent updates (learning rate 5×10^{-4}, seed 42), without transformation sampling. Each patch was placed at $(5, 170)$ in the head view; only RGB pixels were optimized. Top-4 was trained but not physically evaluated; selected layer IDs are not inference inputs.

Evaluation coverage. Each physical condition used 10 in-distribution trials. Table 3 reports clean, Top-3, and Top-5 on three tasks; clean FastWAM scored 0/10 on the second-drawer task, with no attack comparison. Table 36 gives the shared stacking-task results. Baselines used 10,000 updates without EOT: DP-Attacker used 45×45 at $(5, 175)$, DRIFT 32×32 at $(5, 188)$, and the other four 50×50 at $(5, 170)$. Results remain task-specific because method coverage differs.

Table 36: **Piper results on Stack Three Bowls.** Ten in-distribution trials per method. Geometries are method-specific; this is a shared-task system-level comparison, not a matched-geometry ranking.

Method	Successes / trials
Clean FastWAM	10/10
DP-Attacker	0/10
DRIFT	2/10
UPA-RFAS-noPSM	7/10
UPA	3/10
TMA	1/10
UADA	0/10
PASA-KV Top-3	0/10
PASA-KV Top-5	0/10