PACE: PERSISTENT ADVERSARIAL COEFFICIENT EVOLUTION FOR RECURRENT VISUAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recurrent visual agents maintain internal states that integrate past observations, reducing the immediate impact of isolated perturbations while introducing a new attack surface: persistent adversarial influence over their evolving internal dynamics. Existing attacks typically optimize each frame independently, ignoring the temporal dependency introduced by recurrent policies and resulting in ineffective or unstable perturbations. We propose Persistent Adversarial Coefficient Evolution (PACE), a framework that formulates adversarial manipulation as the evolution of a compact latent attack state over time. PACE first constructs a low-dimensional adversarial basis by querying the frozen agent, generating performance-degrading perturbations, and extracting dominant perturbation directions through factorization. During interaction, the previous attack coefficient initializes the current optimization, while temporal smoothing regulates abrupt changes in the adversarial state. The resulting perturbations are projected onto the valid ℓ_∞ constraint and image domain before being applied to the victim agent. Through warm-started coefficient updates and temporal smoothing, PACE retains attack information across frames while adapting perturbations to the current observation. Experiments on DreamerV3 across DMC Walker Walk, Atari Pong, and Crafter show that PACE achieves the largest reported mean reward reduction among the compared methods, tying with UAP-RL on Crafter, while exhibiting substantially lower temporal variation than MAD, PA-AD, Illusory, and DAPGD. These results highlight the importance of modeling adversarial attacks as evolving processes when evaluating the robustness of recurrent visual decision-making systems.

1 INTRODUCTION

Visual decision-making requires integrating information across observations. Dreamer-style agents maintain a recurrent latent state that summarizes past observations and supports action selection through a learned world model (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; 2023). A perturbed observation can therefore influence both the current decision and the internal representation carried into subsequent interactions, making the temporal development of adversarial influence central to robustness evaluation.

Small observation perturbations can disrupt reinforcement learning policies (Huang et al., 2017; Kos & Song, 2017; Lin et al., 2017; Pattanaik et al., 2018). Subsequent work explores robust state-observation formulations (Zhang et al., 2020; 2021), real-time attacks (Tekgul et al., 2022), policy-aware adversaries (Sun et al., 2022), and detectability-aware attacks (Franzmeyer et al., 2024). However, an attack conditioned on the victim's recurrent state may still restart its optimization at every frame, discarding previous solutions. This distinction motivates maintaining an evolving attack state alongside the victim's recurrent representation.

We study white-box, causal, online attacks against a frozen DreamerV3 agent. The attacker modifies only the current image within an ℓ_∞ budget, using a fixed number of gradient updates before the agent acts. The challenge is to adapt to each observation while retaining useful attack information across steps. We address this through a shared perturbation basis and an evolving coefficient vector that determines how its directions are combined.

We propose **Persistent Adversarial Coefficient Evolution (PACE)**, which maintains this coefficient vector as a compact attack state. Offline, PACE collects critic-guided perturbations along clean

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.

trajectories and extracts their dominant directions through singular value decomposition. Motivated by shared adversarial structure and low-dimensional attack spaces (Moosavi-Dezfooli et al., 2017; Yan et al., 2019), this victim-aware basis provides fixed coordinates for subsequent attack updates.

Online, the previous coefficient state initializes the current critic-guided optimization. An exponential moving average moderates coefficient changes before the resulting perturbation is projected onto the ℓ_∞ constraint and image domain. The projected perturbation is then encoded back into coefficients for the next step. This combination of reuse, refinement, and smoothing creates a persistent optimization process without future observations or differentiation through future interactions.

We evaluate PACE on DreamerV3 across DMC Walker Walk (Tassa et al., 2018), Atari Pong, and Crafter. Among five attack baselines, PACE achieves the largest reported reward drop on all three tasks, tying with UAP-RL on Crafter. It also exhibits substantially lower temporal variation than MAD, PA-AD, Illusory, and DAPGD. Analyses of action-distribution shift and perceptual similarity further characterize its behavioral effects and visual trade-offs.

Our contributions are threefold:

- We distinguish conditioning attacks on recurrent context from preserving attack optimization state across frames, motivating persistent updates for causal interaction.
- We propose PACE, combining a critic-guided basis with warm-started coefficient optimization, temporal smoothing, and constraint projection.
- We evaluate PACE on three tasks against five baselines, assessing return reduction, temporal variation, action-distribution shift, and perceptual similarity.

2 Related Work

World-model reinforcement learning. World models learn environment dynamics for prediction and decision-making (Ha & Schmidhuber, 2018). Dreamer combines recurrent latent dynamics with behavior learning from imagined trajectories across continuous control, Atari, and open-ended domains (Hafner et al., 2019; 2020; 2023). Related directions include learned-model planning (Schrittwieser et al., 2020), transformer-based (Micheli et al., 2023; Robine et al., 2023; Zhang et al., 2023) and diffusion-based world models (Alonso et al., 2024), generative environments (Bruce et al., 2024), scalable model-based control (Hansen et al., 2024), and robotic deployment (Wu et al., 2022); see Ding et al. (2024) for a survey. We focus on DreamerV3, whose recurrent state motivates attacks that retain and adapt information across observations.

Adversarial attacks on deep policies. Adversarial vulnerability in deep networks (Szegedy et al., 2014; Goodfellow et al., 2015; Papernot et al., 2016; Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017; Madry et al., 2018) extends to reinforcement learning, where observation perturbations alter actions and reduce return (Huang et al., 2017; Kos & Song, 2017; Lin et al., 2017; Pattanaik et al., 2018). Subsequent work explores robust state-observation formulations (Zhang et al., 2020; 2021), real-time attacks (Tekgul et al., 2022), policy-aware attackers (Sun et al., 2022), detectability constraints (Franzmeyer et al., 2024), and policy-distribution objectives (Duan et al., 2025). Adversarial agents and broader robustness questions are also studied (Gleave et al., 2020; Korkmaz, 2023; Bai et al., 2025). These methods differ in their objectives and temporal structure. PACE focuses on preserving the attack optimization state, beyond conditioning the current objective on recurrent context.

Robust RL and defenses. Defenses include adversarial training with learned attackers (Pinto et al., 2017; Zhang et al., 2021), regularization against worst-case observations (Oikarinen et al., 2021), worst-case-aware training without explicit adversarial rollouts (Liang et al., 2022), and game-theoretic treatments of temporally coupled disturbances (Liang et al., 2024). PACE addresses the complementary problem of evaluating frozen recurrent agents through causal observation attacks, without modifying victim training.

Structured and temporally coherent attacks. Universal perturbations (Moosavi-Dezfooli et al., 2017), low-frequency attacks (Guo et al., 2019), and subspace attacks (Yan et al., 2019) exploit shared or restricted attack directions. Video attacks explore temporal structure and perceptual

constraints (Wei et al., 2019; Li et al., 2019), while universal attacks in reinforcement learning reuse perturbations across observations (Tekgul et al., 2022). PACE combines a fixed basis extracted from critic-guided perturbations with adaptive coefficients carried across frames. Warm-started refinement and smoothing couple this victim-aware representation with persistent online optimization.

3 Threat Model

3.1 Victim Model

We consider observation attacks against recurrent visual world-model agents, using a pretrained **DreamerV3** policy (Hafner et al., 2023) as the victim. At each step, the agent receives an RGB image $o_t \in [0, 255]^{H \times W \times C}$, where $H = W = 64$ and $C = 3$. The agent incorporates the observation into its recurrent latent state, which provides the context for action selection and value prediction. The actor produces an action distribution, while the critic predicts a categorical value distribution.

We evaluate this victim architecture on three tasks spanning different decision regimes: DMC Walker Walk for continuous control, Atari Pong for discrete control, and Crafter for open-ended interaction. The victim's parameters remain fixed throughout attack construction and evaluation. The attacker neither retrains nor fine-tunes the victim, and it cannot modify the environment's transition or reward mechanisms.

3.2 Attacker's Capability

The attacker can modify only the image observation supplied to the victim. It cannot directly modify the victim's actions, rewards, environment state, or recurrent latent state. Its influence on subsequent behavior and internal states occurs through the perturbed observations.

We adopt a *white-box, causal, online* setting. The attacker has access to the frozen victim's parameters and differentiable computations, including its observation encoder, recurrent model, actor, and critic. At step t, it can evaluate attack objectives using the current observation and the victim's incoming recurrent context. This context is held fixed during the current optimization; the attacker differentiates with respect to the current image without backpropagating through preceding environment interactions. It must construct the perturbation before the victim selects the action used for the next environment step, without accessing future test observations.

As specified in Section 5, deployment-time causality applies to all evaluated methods. Methods requiring offline preprocessing, such as universal perturbation fitting or adversarial basis construction, may use clean rollouts of the same frozen victim before evaluation. The resulting shared parameters, including the basis V_r used by PACE, remain fixed during deployment. Online attack variables, such as PACE's coefficient state, may evolve causally using information available up to the current step.

Each perturbed observation must satisfy

$$\tilde{o}_t = Q_8(o_t + \delta_t), \qquad \|\delta_t\|_\infty \le \varepsilon_{\mathrm{px}}, \qquad o_t + \delta_t \in [0, 255]^{H \times W \times C}. \tag{1}$$

Here, δ_t denotes the floating-point perturbation, and Q_8 denotes elementwise conversion to `uint8`, which truncates fractional pixel values on $[0, 255]$. For integer-valued clean images and the integer pixel budget used here, this conversion preserves the perturbation budget. We use $\varepsilon_{\mathrm{px}} = 24$, equivalent to $\varepsilon = 24/255$ when images are normalized to $[0, 1]$. All compared methods use the same perturbation budget. Online optimization is additionally limited to a small number of gradient updates per observation, making effective use of the available updates an important design objective.

3.3 Attacker's Goal

Let T denote the episode horizon and t_{start} the attack onset. The attacker seeks to reduce the victim's expected return after the attack begins. Writing $\mathcal{A}$ for a causal attack procedure, the objective is

$$\begin{aligned} \min_{\mathcal{A}} \quad & \mathbb{E}_{\pi, \mathcal{P}, \mathcal{A}} \left[\sum_{t=t_{\mathrm{start}}}^{T-1} r_t \right] \\ \text{s.t.} \quad & \|\delta_t\|_\infty \le \varepsilon_{\mathrm{px}}, \\ & o_t + \delta_t \in [0, 255]^{H \times W \times C}, \quad t_{\mathrm{start}} \le t \le T - 1, \end{aligned} \tag{2}$$

where π is the frozen victim policy and $\mathcal{P}$ denotes the environment dynamics. The perturbation δ_t is generated using only information available at step t, and $\delta_t = 0$ before the attack onset.

We do not differentiate through the environment or optimize using true environment reward gradients. Instead, attacks use differentiable surrogate objectives exposed by the frozen victim. PACE uses a critic-guided objective, while baselines retain their method-specific objectives within the same access constraints. These surrogates guide online perturbation updates; they do not constitute direct optimization of the full episodic return.

Beyond return reduction, we evaluate temporal variation and perceptual similarity to characterize perturbation consistency and visual distortion. These are additional evaluation criteria rather than guarantees implied by the ℓ_∞ constraint.

3.4 Challenges

The above setting raises three coupled challenges that motivate PACE.

(C1) Maintaining attack continuity across recurrent decisions. The victim's decisions depend on a latent state that evolves throughout interaction. An attack conditioned on the current recurrent context can still discard its own previous solution if optimization restarts at every frame. The challenge is to retain useful attack information while adapting to changing observations and recurrent contexts.

(C2) Effective optimization with limited updates. Each image contains $D = H \times W \times C$ pixel coordinates, while the attacker has only a few gradient updates per observation. Searching this space from scratch provides no explicit mechanism for reusing directions discovered in previous attack updates. A compact, victim-aware parameterization can restrict optimization to a shared set of attack directions and support reuse across frames, although fewer optimization variables alone do not guarantee lower runtime.

(C3) Balancing adaptation, effectiveness, and temporal consistency. The attack must adapt to the current observation while remaining within the perturbation budget. Large changes between successive perturbations may increase temporal variation, whereas excessive smoothing may limit adaptation and weaken the attack. The challenge is therefore to preserve useful continuity without preventing observation-dependent refinement.

Together, these challenges motivate representing the attack as an evolving coefficient state in a fixed, victim-aware basis. PACE combines this representation with warm-started online optimization and temporal smoothing, as detailed in Section 4.

4 Method

Motivated by Section 3.4, we propose **Persistent Adversarial Coefficient Evolution (PACE)**. PACE represents the attack state using a compact coefficient vector that persists across observations. As shown in Figure 1, it comprises two stages: **Stage A** constructs a fixed adversarial basis from critic-guided perturbations collected offline; **Stage B** evolves its coefficients through warm-started online optimization, temporal smoothing, and constraint projection. The basis defines shared attack directions, while the coefficients adapt their combination throughout interaction.

4.1 Offline Adversarial Basis Construction

Critic-guided perturbation collection. We collect perturbations along a clean rollout of the frozen victim, starting after a 200-step warm-up. Let c_t denote the incoming recurrent context, held fixed during optimization at step t. For categorical critic logits $\ell_k(o; c_t)$, we use the expected bin index as a differentiable surrogate:

$$v(o; c_t) = \sum_{k=0}^{K-1} k \cdot \mathrm{softmax}(\ell(o; c_t))_k. \tag{3}$$

The bins are ordered by value (Hafner et al., 2023); minimizing this index-based surrogate favors lower-index predictions. We do not interpret it as a calibrated estimate of environment return.

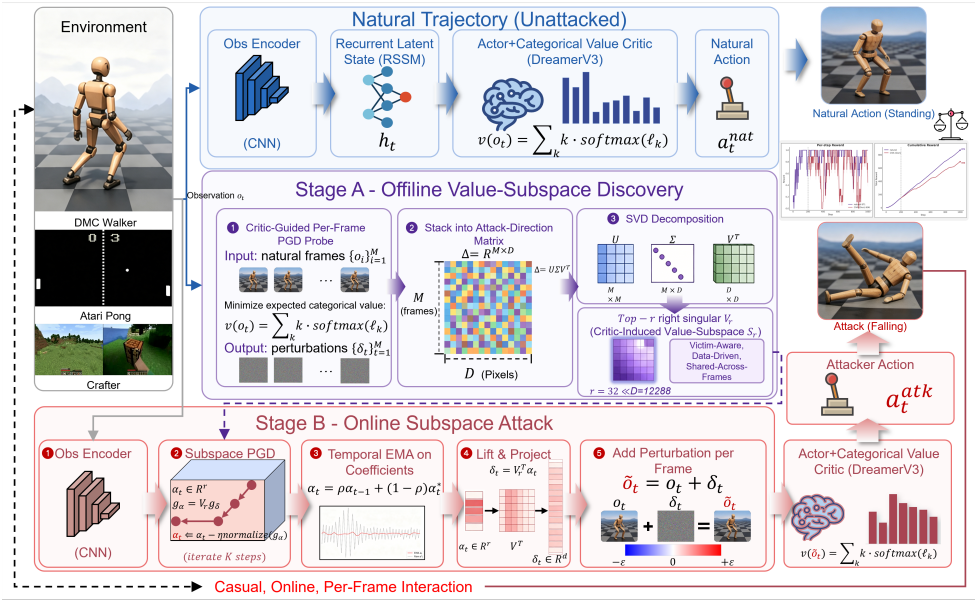

Figure 1: **PACE framework.** Stage A extracts a fixed adversarial basis from offline critic-guided perturbations. Stage B carries coefficients across frames, refines them online, and applies temporal smoothing before projecting the resulting perturbation.In the diagram, V_r and K correspond to $V_r^\top$ and N in the main text, respectively.

For raw pixels in $[0, 255]$, define the feasible projection

$$\mathcal{P}_t(d) = \mathrm{clip}_{[0,255]}\left(o_t + \mathrm{clip}_{[-\varepsilon_{\mathrm{px}},\varepsilon_{\mathrm{px}}]}(d)\right) - o_t, \tag{4}$$

where $\varepsilon_{\mathrm{px}} = 24$ corresponds to a normalized budget of $24/255$. Starting from $\delta^{(0)} = 0$, we perform L projected gradient updates (Madry et al., 2018):

$$\delta^{(\ell+1)} = \mathcal{P}_t\left(\delta^{(\ell)} - \eta_{\mathrm{off}}\ \mathrm{normalize}\left(\nabla_o v(o_t + \delta^{(\ell)}; c_t)\right)\right), \tag{5}$$

where $\mathrm{normalize}(g) = g/(\sqrt{\mathrm{mean}(g^2)}+10^{-6})$. The probes do not alter the clean rollout. Flattening the resulting perturbations from M frames gives

$$\Delta = [\delta_1, \ldots, \delta_M]^\top \in \mathbb{R}^{M\times D}, \qquad D = HWC. \tag{6}$$

Thus, the basis construction uses optimized perturbations rather than raw observations or raw gradients.

Extracting the shared basis. We apply SVD (Eckart & Young, 1936):

$$\Delta = U\Sigma V^\top, \qquad V_r = [v_1, \ldots, v_r] \in \mathbb{R}^{D\times r}. \tag{7}$$

The columns of V_r are the top-r right singular vectors, with $V_r^\top V_r = I$. They capture the dominant directions in the collected perturbations and define the attack subspace $\mathcal{S}_r = \mathrm{col}(V_r)$. We use $r = 32$ for $D = 12288$ and freeze the basis during deployment. This provides a shared coordinate system for persistent attack updates; it does not imply that every vector in $\mathcal{S}_r$ is effective at every observation.

4.2 Online Coefficient Evolution

Persistent attack state. PACE maintains a coefficient state $\alpha_t \in \mathbb{R}^r$, initialized to zero before the attack begins. Given the current observation and incoming recurrent context, the online objective uses the same critic surrogate:

$$\mathcal{L}_{\mathrm{val}}(\delta; o_t, c_t) = v(o_t + \delta; c_t). \tag{8}$$

The coefficient vector parameterizes a candidate perturbation as

$$d = V_r \alpha. \tag{9}$$

At each frame, the previous state initializes the search. Following the implementation, we first project its reconstructed perturbation for the current image and map it back:

$$\alpha_t^{(0)} = V_r^\top \mathcal{P}_t(V_r \alpha_{t-1}). \tag{10}$$

This preserves information from preceding updates while accounting for the current pixel constraints.

Observation-dependent refinement. For each online update, we evaluate the image gradient at $x_t^{(\ell)} = \mathrm{clip}_{[0,255]}(o_t + V_r \alpha_t^{(\ell)})$ and project it into coefficient space:

$$\begin{aligned} g_\alpha^{(\ell)} &= V_r^\top \, \nabla_x v(x; c_t)|_{x=x_t^{(\ell)}}, \\ \beta_t^{(\ell+1)} &= \alpha_t^{(\ell)} - \eta_{\mathrm{on}} \, \mathrm{normalize}(g_\alpha^{(\ell)}), \\ \alpha_t^{(\ell+1)} &= V_r^\top \mathcal{P}_t(V_r \beta_t^{(\ell+1)}). \end{aligned} \tag{11}$$

After N updates, a final projection and encoding produce $\alpha_t^\star = V_r^\top \mathcal{P}_t(V_r \alpha_t^{(N)})$. Here $\alpha_t^\star$ denotes the finite-step optimization output, not a guaranteed optimum. The recurrent context remains fixed throughout these updates; no gradients are propagated through future interactions.

Temporal smoothing and state carryover. We blend the refined coefficients with the preceding attack state:

$$\bar{\alpha}_t = \rho \alpha_{t-1} + (1 - \rho)\alpha_t^\star, \qquad \rho \in [0, 1]. \tag{12}$$

Larger ρ places more weight on the preceding state, whereas $\rho = 0$ removes explicit EMA smoothing but retains warm-started optimization. We then construct the feasible perturbation and store its coefficient representation:

$$\delta_t = \mathcal{P}_t(V_r \bar{\alpha}_t), \quad \alpha_t = V_r^\top \delta_t, \quad \tilde{o}_t = Q_8(o_t + \delta_t). \tag{13}$$

Before victim inference, the clipped image is converted to `uint8` by Q_8. Coefficient re-encoding uses the floating-point perturbation δ_t before this conversion. The victim processes $\tilde{o}_t$ to update its recurrent state and select the next action.

Pixel-space clipping can move δ_t outside $\mathcal{S}_r$; the projected floating-point perturbation satisfies the pixel constraints, while α_t retains its subspace representation for the next frame. EMA moderates coefficient changes without guaranteeing improved attack strength or a fixed bound on temporal variation. Together, warm starts, current-observation refinement, and smoothing implement persistent coefficient evolution under causal interaction.

5 Experiment

5.1 Experimental Setup

Environments and victim. We evaluate on three visual decision-making tasks: DMC Walker Walk (Tassa et al., 2018) (continuous control), Atari Pong (Bellemare et al., 2013; Mnih et al., 2015) (discrete control), and Crafter (Hafner, 2021) (open-ended survival). All tasks use $64 \times 64 \times 3$ RGB observations ($D = 12{,}288$) and an evaluation horizon of at most $1{,}000$ steps. The victim is DreamerV3 (Hafner et al., 2023), trained per task using the official configuration. Its checkpoint remains frozen throughout attack construction and evaluation.

Baselines and attack budget. We compare PACE against five observation-space attackers: MAD (Zhang et al., 2020), UAP-RL (Tekgul et al., 2022), PA-AD (Sun et al., 2022), Illusory (Franzmeyer et al., 2024), and DAPGD (Duan et al., 2025). All methods share the same white-box access and normalized ℓ_∞ budget $\varepsilon = 24/255$, equivalent to $\varepsilon_{\mathrm{px}} = 24$ on raw pixels. Attacks begin at $t = 200$ and follow the causal deployment protocol in Section 3.2.

Table 1: **Attack results on DMC walker walk.** "Natural" is the unattacked baseline; arrows show the favorable direction for the attacker. **Bold** = best, underline = second best.

Method	Venue / Year	Episode Reward ↓	Drop% ↑	Action-KL ↑	TempAbs ↓	SSIM ↑
Natural (no attack)	—	952.25	—	—	—	1.000
MAD (Zhang et al., 2020)	NeurIPS 2020	934.71	1.84%	0.753	9.126	**0.748**
UAP-RL (Tekgul et al., 2022)	ESORICS 2022	730.86	23.25%	1.000	**0.00**	0.656
PA-AD (Sun et al., 2022)	ICLR 2022	786.75	17.38%	**1.35**	19.97	0.35
Illusory (Franzmeyer et al., 2024)	ICLR 2024	795.13	16.50%	1.018	10.97	0.445
DAPGD (Duan et al., 2025)	ICASSP 2025	773.33	18.79%	0.775	12.31	0.501
PACE (Ours)	**—**	**703.97**	**26.07%**	1.130	0.646	0.710

Table 2: **Attack results on Atari Pong.**

Method	Venue / Year	Episode Reward ↓	Drop% ↑	Action-KL ↑	TempAbs ↓	SSIM ↑
Natural (no attack)	—	21.0	—	—	—	1.000
MAD (Zhang et al., 2020)	NeurIPS 2020	7.0	66.67%	0.155	4.357	**0.864**
UAP-RL (Tekgul et al., 2022)	ESORICS 2022	18.0	14.29%	0.098	**0.000**	0.344
PA-AD (Sun et al., 2022)	ICLR 2022	17.0	19.05%	0.069	4.029	0.697
Illusory (Franzmeyer et al., 2024)	ICLR 2024	16.0	23.81%	0.138	9.570	0.599
DAPGD (Duan et al., 2025)	ICASSP 2025	19.0	9.52%	0.080	17.194	0.719
PACE (Ours)	**—**	**3.0**	**85.71%**	**0.184**	0.394	0.566

PACE hyperparameters. Stage A collects $M = 600$ critic-guided perturbations after a 200-step clean warm-up, using $L = 8$ gradient updates per frame. SVD extracts a rank-32 basis from $\Delta \in \mathbb{R}^{600\times 12288}$. Stage B performs $N = 8$ coefficient updates per frame, warm-started from the preceding attack state, followed by EMA with $\rho = 0.75$. Both stages use RMS-normalized gradients with numerical step size $\eta_{\text{off}} = \eta_{\text{on}} = \varepsilon_{\text{px}}/4 = 6$ in the implementation's raw-pixel parameterization. The basis remains fixed during evaluation.

Evaluation protocol. Each method is evaluated over five independent seeds, with one episode per seed. We report across-seed means of episode reward, reward drop relative to the natural policy (**Drop%**) (Pattanaik et al., 2018; Zhang et al., 2020), KL divergence between clean and attacked action distributions (**Action-KL**) (Kullback & Leibler, 1951; Sun et al., 2022; Duan et al., 2025), mean absolute change between successive perturbations (**TempAbs**, in raw-pixel units) (Wei et al., 2019; Li et al., 2019), and clean–perturbed image SSIM (Wang et al., 2004). All experiments run on a single NVIDIA RTX PRO 6000 Blackwell GPU with 96 GB memory.

5.2 Main Results

Attack effectiveness. PACE achieves the lowest episode reward among the compared methods on DMC Walker Walk and Atari Pong (Tables 1 and 2). On Walker Walk, it reduces reward from 952.25 to 703.97, a 26.07% drop, exceeding UAP-RL's 23.25% by 2.82 percentage points. On Pong, PACE reduces reward from 21 to 3, achieving an 85.71% drop compared with MAD's 66.67%. The remaining baselines achieve less than 25% drop. Figure 2 shows the corresponding cumulative reward trajectories, with PACE finishing lowest on both tasks. On Crafter, PACE ties UAP-RL for the largest reward drop (Appendix A.2).

Behavioural deviation. PACE attains Action-KL of 1.130 on DMC and 0.184 on Pong, ranking second and first, respectively. On DMC, PA-AD produces a larger Action-KL (1.35) but a smaller reward drop (17.38%). Thus, larger distributional deviation does not necessarily translate into stronger performance degradation. PACE achieves the greatest return reduction among the compared methods without maximizing Action-KL on every task.

Temporal consistency and visual similarity. PACE achieves TempAbs of 0.646 on DMC and 0.394 on Pong, second only to UAP-RL's reported zero variation. Compared with MAD, PA-AD,

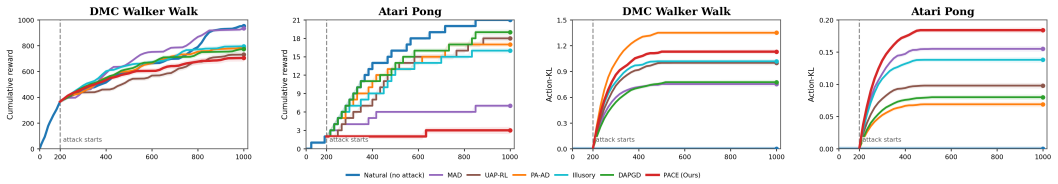

Figure 2: **Attack dynamics over time.** Cumulative reward (left pair) and Action-KL (right pair) on DMC Walker Walk and Atari Pong. Dashed lines mark attack onset at $t = 200$.

Illusory, and DAPGD, its temporal variation is approximately 14–31 times lower on DMC and 10–44 times lower on Pong. UAP-RL achieves lower temporal variation but weaker reward suppression, particularly on Pong. Visual similarity presents a separate trade-off: PACE's SSIM is second highest among attacks on DMC (0.710), but falls below several baselines on Pong (0.566). The results therefore support strong attack effectiveness with low temporal variation, rather than uniformly superior perceptual similarity.

5.3 Analysis of Policy Decision Shifts

Figure 3 compares action logits and selected actions under the natural policy, Illusory, and PACE at steps 250, 300, and 350 on Atari Pong. PACE selects actions 3, 4, and 14, differing from the natural policy's 16, 17, and 11 in all three examples. Illusory changes the selected action at steps 250 and 350 but retains action 17 at step 300. These snapshots illustrate decision changes accompanying the observed distributional shifts; aggregate attack effectiveness is assessed through episodic return.

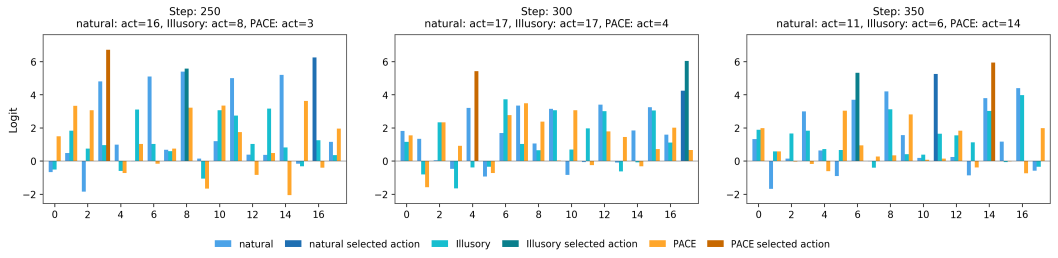

Figure 3: **Policy-logit shifts on Atari Pong.** Dark bars mark selected actions under the natural policy, Illusory, and PACE at three attack-time steps.

5.4 Ablation Study

Tables 3 and 4 compare PACE (V0) with four variants on DMC Walker Walk. V1 replaces the critic-guided basis with a random orthonormal basis of the same rank. V2 uses full-pixel critic-guided PGD without EMA. V3 removes EMA while retaining coefficient warm starts. V4 uses a reward-head signal and a corresponding reward-guided SVD basis.

Critic-guided basis. Replacing the learned basis with a random basis (V1) reduces reward drop from 26.07% to 6.90% and Action-KL from 1.130 to 0.058. This supports the importance of the extracted directions beyond dimensionality reduction alone. Full-pixel PGD without EMA (V2) achieves a 12.61% drop and TempAbs of 3.235, compared with PACE's 26.07% and 0.646. However, V2 also removes smoothing, so this comparison does not isolate the subspace constraint. Its higher SSIM (0.908) further illustrates that visual similarity alone does not indicate attack effectiveness.

Temporal smoothing. Removing EMA (V3) raises reward drop from 26.07% to 26.84%, a gain of 0.77 percentage points. This comes with approximately 2.7 times higher TempAbs (1.7496 versus 0.646) and lower SSIM (0.675 versus 0.710). EMA therefore improves temporal consistency and

Table 3: **Ablation: reward drop on DMC walker walk.** The **Online** and **EMA** columns indicate whether subspace-restricted online PGD and temporal EMA are enabled.

ID	Variant	(A) Signal	(B) Subspace	Online	EMA	EpReward ↓	Drop% ↑
V0	**PACE (full)**	critic	critic-SVD	✓	✓	703.97	26.07%
V1	w/o critic-induced subspace	critic	random	✓	✓	886.54	6.90%
V2	w/o subspace constraint (full-pixel)	critic	—	✗	✗	832.21	12.61%
V3	w/o temporal EMA	critic	critic-SVD	✓	✗	**696.66**	**26.84%**
V4	w/o critic (reward-grad subspace)	reward	reward-SVD	✓	✓	906.13	4.84%

Table 4: **Ablation: behavioural deviation, temporal variation, and visual similarity (DMC Walker Walk).** Companion to Table 3.

ID	Variant	(A) Signal	(B) Subspace	Online	EMA	Action-KL ↑	TempAbs ↓	SSIM ↑
V0	**PACE (full)**	critic	critic-SVD	✓	✓	**1.130**	0.646	0.710
V1	w/o critic-induced subspace	critic	random	✓	✓	0.058	0.8439	0.612
V2	w/o subspace constraint (full-pixel)	critic	—	✗	✗	0.730	3.2350	**0.908**
V3	w/o temporal EMA	critic	critic-SVD	✓	✗	0.798	1.7496	0.675
V4	w/o critic (reward-grad subspace)	reward	reward-SVD	✓	✓	1.054	**0.542**	0.833

visual similarity at a small cost in reward suppression. Since V3 retains warm-started coefficient updates, it tests smoothing rather than the removal of all cross-frame persistence.

Critic-guided versus reward-guided attacks. Using the reward-head signal and reward-derived basis (V4) reduces reward drop to 4.84%, despite an Action-KL of 1.054, close to PACE's 1.130. V4 achieves lower TempAbs (0.542) and higher SSIM (0.833), but substantially weaker return suppression. This comparison favors the critic-guided configuration for attack effectiveness, without separating the effects of the optimization signal and basis construction. Overall, PACE combines near-best reward suppression among the ablations with low temporal variation.

6 Conclusion

This paper proposes PACE, Persistent Adversarial Coefficient Evolution, for attacking recurrent visual agents under online and ℓ_∞ constraints. PACE constructs a compact critic-guided perturbation basis from a frozen victim and maintains an evolving coefficient state through warm-started online optimization and temporal smoothing. Experiments on DreamerV3 across three visual decision-making tasks show that PACE achieves the largest reported mean reward reduction among the compared methods, tying with UAP-RL on Crafter, while maintaining low temporal variation. The ablations highlight the contribution of the critic-guided basis and the trade-off between attack strength and temporal smoothness. These findings motivate evaluating attacks on recurrent visual agents through both episodic return and temporal perturbation characteristics.

Limitations. PACE requires white-box access to the victim's differentiable computations, including its critic, and clean rollouts for offline basis construction. Our main evaluation focuses on DreamerV3, with supplementary basis-transfer experiments on STORM and IRIS on Atari Pong. Further discussion appears in Appendix A.8.

AI Use Statement

We used AI tools to assist with organizing references, refining the writing based on our ideas and research direction, and checking clarity and consistency. AI assistance also supported the review of mathematical notation and the correspondence between the algorithm description and implementation. This assistance focused on presenting the method clearly and identifying inconsistencies across the text, equations, and implementation.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Fengshuo Bai, Runze Liu, Yali Du, Ying Wen, and Yaodong Yang. Rat: Adversarial attacks on deep reinforcement agents for targeted behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15453–15461, 2025.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2017.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, Fengli Xu, and Yong Li. Understanding world or predicting future? a comprehensive survey of world models. *arXiv preprint arXiv:2411.14499*, 2024.

Tianyang Duan, Zongyuan Zhang, Zheng Lin, Yue Gao, Ling Xiong, Yong Cui, Hongbin Liang, Xianhao Chen, Heming Cui, and Dong Huang. Rethinking adversarial attacks in reinforcement learning from policy distribution perspective. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Tim Franzmeyer, Stephen McAleer, Joao F Henriques, Jakob Foerster, Philip Torr, Adel Bibi, and Christian Schroeder de Witt. Illusory attacks: Information-theoretic detectability matters in adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2024.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

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

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

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

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. In *International Conference on Learning Representations (ICLR) Workshop*, 2017.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V. Krishnamurthy, Amit K. Roy-Chowdhury, and Ananthram Swami. Adversarial perturbations against real-time video classification systems. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, and Furong Huang. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, Xiangyu Liu, Benjamin Eysenbach, Tuomas Sandholm, Furong Huang, and Stephen McAleer. Game-theoretic robust reinforcement learning handles temporally-coupled perturbations. In *International Conference on Learning Representations (ICLR)*, 2024.

Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

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 (ICLR)*, 2018.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations (ICLR)*, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Tuomas Oikarinen, Wang Zhang, Alexandre Megretski, Luca Daniel, and Tsui-Wei Weng. Robust deep reinforcement learning through adversarial loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2018.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. *arXiv preprint arXiv:2303.07109*, 2023.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. Who is the strongest enemy? towards optimal and efficient evasion attacks in deep RL. In *International Conference on Learning Representations (ICLR)*, 2022.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Buse GA Tekgul, Shelly Wang, Samuel Marchal, and N Asokan. Real-time adversarial perturbations against deep reinforcement learning policies: attacks and defenses. In *European Symposium on Research in Computer Security*, pp. 384–404. Springer, 2022.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. DayDreamer: World models for physical robot learning. In *Conference on Robot Learning (CoRL)*, 2022.

Ziang Yan, Yiwen Guo, and Changshui Zhang. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in neural information processing systems*, 33:21024–21037, 2020.

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*, 2021.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. STORM: Efficient stochastic transformer based world models for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A Appendix

A.1 Detailed Experimental Setup

This appendix supplements Section 5.1 with victim-training configurations, the attack pipeline, and metric definitions. Additional experiments examine basis transfer across victim models, online computational efficiency, and hyperparameter sensitivity.

Software and hardware. Experiments run on a single NVIDIA RTX PRO 6000 Blackwell GPU with 96 GB memory. The primary DreamerV3 experiments use JAX for the victim and differentiable attack computations. NumPy and scikit-image support basis construction and metric computation. The supplementary efficiency comparison uses PyTorch implementations, as specified in Section A.4. Although PACE maintains only $r = 32$ attack coefficients, gradient computation still requires backpropagation through the frozen victim.

Victim training. For the main three-task evaluation, we use the official DreamerV3 implementation without changing its model architecture or training objectives. A separate checkpoint is trained for each task using the following configurations:

- **DMC Walker Walk:** 1.1×10^6 training steps, `train_ratio=256`, and vision-only $64 \times 64 \times 3$ observations with proprioception disabled.
- **Atari Pong:** 5.1×10^7 training steps, `train_ratio=32`, and the official task configuration for image preprocessing and environment interaction.
- **Crafter:** 1.1×10^6 training steps, `train_ratio=512`, `envs=1`, and $64 \times 64 \times 3$ observations.

The resulting checkpoints remain frozen during attack construction and evaluation. Gradients update attack variables only. Section A.3 additionally evaluates basis reuse on another DreamerV3 checkpoint and two transformer-based world-model agents, STORM and IRIS.

Attack pipeline. PACE operates in two stages:

1. **Offline basis construction.** After a 200-step clean warm-up, we collect $M = 600$ critic-guided perturbations using $L = 8$ updates per frame. These probes do not alter the clean rollout. SVD of the flattened perturbation matrix $\Delta \in \mathbb{R}^{600\times 12288}$ yields a fixed orthonormal basis $V_r \in \mathbb{R}^{12288\times 32}$.
2. **Online coefficient evolution.** Evaluation starts in a fresh episode, with attacks beginning at $t_0 = 200$. The coefficient state is initialized to zero and carried across attacked frames. Each step uses $N = 8$ coefficient updates, EMA with $\rho = 0.75$, pixel-space projection, and coefficient re-encoding for the next step.

Both stages use RMS-normalized gradients with numerical step size $\eta_{\text{off}} = \eta_{\text{on}} = 6$ in the raw-pixel parameterization. The raw-pixel budget is $\varepsilon_{\text{px}} = 24$, equivalent to $24/255$ on normalized images.

Within each main-task comparison, methods share the victim checkpoint, perturbation budget, and deployment-time access constraints. The main evaluation uses five independent evaluation seeds per task, with one episode per seed. The transfer experiment changes the victim while fixing the source basis, and the sensitivity experiment varies the specified hyperparameters. Their protocols are described in the corresponding subsections.

Metric definitions. Index interaction steps by $t = 0, \ldots, T-1$ and let $\mathcal{I} = \{t_0, \ldots, T-1\}$ denote the attacked window. Natural and attacked episodes may follow different trajectories and terminate at different times. Their returns are $R^{\text{nat}} = \sum_t r_t^{\text{nat}}$ and $R^{\text{atk}} = \sum_t r_t^{\text{atk}}$, using the actual length of each episode. Metrics are computed per seed and then averaged.

- **Episode Reward.** The cumulative environment reward R^{atk} over the attacked evaluation episode, including steps before attack onset.

Table 5: **Attack results on Crafter.**

Method	Venue / Year	Episode Reward ↓	Drop% ↑	Action-KL ↑	TempAbs ↓	SSIM ↑
Natural (no attack)	—	10.10	—	—	—	1.000
MAD	NeurIPS 2020	8.10	19.80%	1.837	6.25	0.874
UAP-RL	ESORICS 2022	**7.10**	**29.70%**	0.310	**0.00**	0.527
PA-AD	ICLR 2022	8.10	19.80%	0.913	8.12	0.478
Illusory	ICLR 2024	8.10	19.80%	1.562	19.99	0.495
DAPGD	ICASSP 2025	9.10	9.90%	**1.960**	33.64	0.529
PACE (Ours)	**—**	**7.10**	**29.70%**	0.326	0.27	**0.970**

- **Drop%.** The relative return reduction:

$$\text{Drop}\% = \frac{R^{\text{nat}} - R^{\text{atk}}}{R^{\text{nat}}} \times 100\%. \tag{14}$$

  Natural reference returns are positive in the reported tasks.

- **Action-KL.** Let c_t denote the incoming policy context on the attacked trajectory, which is the recurrent context for DreamerV3. We compare the victim's distributions for the unmodified and perturbed current image under that same context:

$$\text{Action-KL} = \frac{1}{|\mathcal{I}|} \sum_{t \in \mathcal{I}} \text{KL}(\pi(\cdot \mid o_t, c_t) \,\|\, \pi(\cdot \mid \tilde{o}_t, c_t)) \,. \tag{15}$$

  We use diagonal-Gaussian KL for continuous actions and categorical KL for discrete actions. This measures the current observation's effect conditional on the available history, rather than divergence between two independently evolving trajectories.

- **TempAbs.** For projected floating-point perturbations δ_t, we compute

$$\text{TempAbs} = \frac{1}{|\mathcal{I}|D} \sum_{t \in \mathcal{I}} \|\delta_t - \delta_{t-1}\|_1, \qquad \delta_{t_0-1} = 0. \tag{16}$$

  This includes the transition at attack onset and measures variation in raw-pixel units. Lower values indicate smaller changes between successive perturbations.

- **SSIM.** We average structural similarity over attacked frames:

$$\text{SSIM} = \frac{1}{|\mathcal{I}|} \sum_{t \in \mathcal{I}} \text{SSIM}(o_t, \tilde{o}_t). \tag{17}$$

  The metric uses the actual image inputs, with scikit-image's channel-aware implementation and `data_range=255`.

A.2 Additional Results on Crafter

Crafter evaluates open-ended interaction involving survival, resource collection, and achievements. We report episodic environment reward under the same attack budget and evaluation protocol as the main experiments. This quantity is distinct from the benchmark's aggregate achievement score; reward drop should not be interpreted directly as the percentage of achievements prevented.

Attack effectiveness. PACE reduces mean episode reward from 10.10 to 7.10, achieving a 29.70% drop. This ties UAP-RL for the strongest reward suppression, compared with 19.80% for MAD, PA-AD, and Illusory, and 9.90% for DAPGD. Thus, PACE matches the best baseline on Crafter while outperforming the remaining four in the reported mean return.

Behavioural deviation. PACE achieves Action-KL of 0.326, substantially below DAPGD (1.960), MAD (1.837), and Illusory (1.562), despite matching the largest reward drop. UAP-RL similarly combines low Action-KL (0.310) with strong return reduction. These results reinforce that distributional deviation alone does not determine attack effectiveness; they do not establish which latent directions cause the observed return reduction.

Table 6: **Basis transfer on Atari Pong.** All rows use PACE with the same frozen source basis. Values are mean ± standard deviation over five evaluation seeds; Drop% is in percent. The source-reference row is highlighted without ranking different victim models.

Target Victim	Setting	Natural $R\uparrow$	Attacked $R\downarrow$	Drop% $\uparrow$	Action-KL $\uparrow$	TempAbs $\downarrow$	SSIM $\uparrow$
DreamerV3-Seed 1	Source reference	21.00 ± 0.00	3.20 ± 0.45	84.76 ± 2.13	0.191 ± 0.012	0.401 ± 0.053	0.550 ± 0.011
DreamerV3-Seed 2	Cross-checkpoint	20.80 ± 0.45	4.80 ± 1.10	76.95 ± 5.06	0.176 ± 0.009	0.410 ± 0.052	0.576 ± 0.014
STORM	Cross-architecture	19.80 ± 0.84	5.60 ± 1.95	71.85 ± 9.02	0.165 ± 0.012	0.468 ± 0.057	0.578 ± 0.021
IRIS	Cross-architecture	20.00 ± 0.71	5.40 ± 1.67	72.98 ± 8.38	0.155 ± 0.019	0.424 ± 0.024	0.572 ± 0.023

Temporal consistency and visual similarity. PACE has the second-lowest reported TempAbs (0.27), behind UAP-RL (0.00), and the highest SSIM (0.970). Its temporal variation is approximately 23–125 times lower than MAD, PA-AD, Illusory, and DAPGD. PACE exceeds the next-highest SSIM, MAD's 0.874, by 0.096, and exceeds UAP-RL's 0.527 by 0.443. On this task, it therefore matches UAP-RL's return reduction while preserving substantially higher image similarity and maintaining low temporal variation.

A.3 Basis Transfer Across Victim Models

Protocol. We evaluate whether the offline adversarial basis can be reused across victim checkpoints and architectures on Atari Pong. We construct V_r once using a source DreamerV3 checkpoint, denoted DreamerV3-Seed 1, freeze it, and reuse the same basis on a second DreamerV3 checkpoint, STORM, and IRIS. The target checkpoints do not contribute perturbations to offline basis construction.

Each target is evaluated over five evaluation seeds. The checkpoint identifiers in Table 6 are distinct from these evaluation seeds. The source-reference results belong to this transfer experiment and are reported separately from the main attack comparison.

Target adaptation. Only the online attack coefficients are optimized on the target, using its current observation and available policy context. The models use a compatible observation protocol so that the source basis can be applied in the same pixel coordinates. For targets whose value heads differ from DreamerV3's ordered categorical bins, the online attack objective uses the target's native differentiable scalar value estimate.

This experiment therefore evaluates basis reuse with target-specific online gradients. It does not evaluate a black-box attack or transfer of a fixed perturbation without target adaptation.

Results. The source-basis attack produces a 76.95% reward drop on the second DreamerV3 checkpoint and 71.85% and 72.98% drops on STORM and IRIS, respectively, compared with 84.76% on the source checkpoint. The corresponding TempAbs values remain between 0.401 and 0.468 across the four evaluated victims.

These results support reuse of the learned basis across the tested Pong victims without target-specific basis reconstruction. The lower reward drops on the target victims indicate that transfer does not preserve the full source-victim effect.

Scope of generalization. The experiment extends the evaluation beyond a single DreamerV3 checkpoint and architecture. Its scope remains basis transfer under a shared task and compatible observation protocol. It does not establish that the transferred basis is optimal for each target or that it generalizes across tasks, observation transformations, or arbitrary environmental distribution shifts. Online access to the target's value gradients remains required.

A.4 Online Computational Efficiency

Measurement scope. Table 7 reports per-frame attack latency, its 95th percentile (p95), and floating-point operation counts on DMC Walker Walk using a single RTX PRO 6000 GPU. These timing measurements use PyTorch implementations, with `torch.cuda.synchronize()` used for per-frame synchronization. They are distinct from timing the primary JAX implementation; the reported speedup applies to the PyTorch comparison.

Table 7: **Online attack efficiency on DMC Walker Walk.** Speedup is relative to DAPGD. Bold and underlining indicate the best and second-best attack-effectiveness or timing/cost results among the listed methods. Offline preprocessing is excluded.

Method	Drop% ↑	Opt. Dim.	Dim. Reduction	Latency (ms/frame) ↓	p95 (ms) ↓	GFLOPs/frame ↓	Speedup ↑
PA-AD	17.38	—	—	52.46 ± 1.87	56.31	148.65	$0.73\times$
Illusory	16.50	12,288	$1\times$	45.73 ± 1.54	49.28	144.96	$0.84\times$
DAPGD	<u>18.79</u>	12,288	$1\times$	<u>38.20 ± 1.60</u>	<u>41.10</u>	<u>141.20</u>	<u>$1.00\times$</u>
PACE (Ours)	**26.07**	**32**	**$384\times$**	**22.86 ± 0.94**	**25.37**	**138.60**	**$1.67\times$**

The timed attack includes victim forward/backward computations used to generate the perturbation and attack-variable updates. Rendering and environment stepping are excluded. The comparison concerns online attack generation and excludes offline basis construction and other method-specific preprocessing. It therefore does not measure end-to-end episode runtime or the total cost of preparing and deploying an attacker.

Measured efficiency. PACE reduces mean attack latency from DAPGD's 38.20 to 22.86 ms per frame, corresponding to a $1.67\times$ speedup in the reported comparison. Its p95 latency is 25.37 ms, compared with 41.10 ms for DAPGD. PACE also has lower reported mean latency than PA-AD (52.46 ms) and Illusory (45.73 ms).

Optimization dimension versus runtime. The reduction from 12,288 pixel variables to 32 coefficients is $384\times$. This is a reduction in optimization dimension rather than a runtime speedup. The reported operation count changes only from 141.20 to 138.60 GFLOPs per frame relative to DAPGD, consistent with the continued need to differentiate through the victim.

These measurements characterize the tested implementation. Low-dimensional parameterization alone does not guarantee the same speedup for other victims, hardware, or implementations, and the online measurements do not account for the cost of constructing the offline basis.

A.5 Hyperparameter Sensitivity

Experimental scope. We examine the effects of the subspace rank, EMA coefficient, number of offline probe frames, and PGD step count on attack effectiveness, temporal variation, and image similarity. All sweeps are conducted on DMC Walker Walk and report means over five evaluation seeds.

For the rank, EMA, and probe-frame sweeps, we vary one parameter at a time while keeping the remaining settings at their defaults: $r = 32$, $\rho = 0.75$, $M = 600$, $L = 8$, and $N = 8$. The PGD step-count results are reported using k as a neutral step-budget notation, with eight steps as the reference setting.

Subspace rank. Increasing r from 8 to 16 improves reward drop from 22.73% to 25.16%. The default $r = 32$ achieves 26.07%, while increasing the rank to 64 gives 25.84%. Across the tested ranks 16–64, reward drop therefore lies between 25.16% and 26.07%.

The larger tested ranks are accompanied by higher TempAbs and lower SSIM. These results support $r = 32$ as an effective setting for the evaluated task, while showing that increasing the number of attack directions does not necessarily improve return suppression.

EMA coefficient. Setting $\rho = 0$ removes explicit EMA smoothing while retaining warm-started coefficient optimization. It yields a 26.84% reward drop with TempAbs of 1.750 and SSIM of 0.675. The default $\rho = 0.75$ gives a 26.07% reward drop, TempAbs of 0.646, and SSIM of 0.710.

Increasing ρ to 0.90 further reduces TempAbs to 0.341 and raises SSIM to 0.738, while reward drop decreases to 25.49%. The sweep exposes a strength–smoothness trade-off: stronger smoothing reduces temporal variation but can weaken reward suppression. It extends the no-EMA ablation in the main paper rather than testing removal of all cross-frame persistence.

Table 8: **Hyperparameter sensitivity on DMC Walker Walk.** Results are means over five evaluation seeds; Drop% is in percent. Highlighted rows and bold parameter values indicate the reference configuration rather than the best result. The PGD step-count sweep is denoted by k.

Parameter	Value	Episode Reward ↓	Drop% ↑	Action-KL ↑	TempAbs ↓	SSIM ↑
Rank r	8	735.82	22.73	0.968	0.534	0.754
	16	712.64	25.16	1.083	0.598	0.727
	32 (default)	703.97	26.07	1.130	0.646	0.710
	64	706.18	25.84	1.119	0.711	0.698
EMA ρ	0.00	696.66	26.84	0.798	1.750	0.675
	0.50	701.82	26.30	1.142	0.891	0.697
	0.75 (default)	703.97	26.07	1.130	0.646	0.710
	0.90	709.53	25.49	1.021	0.341	0.738
Probe frames M	100	731.42	23.19	0.997	0.583	0.731
	300	711.86	25.24	1.086	0.617	0.719
	600 (default)	703.97	26.07	1.130	0.646	0.710
	1200	705.14	25.95	1.122	0.663	0.704
PGD steps k	1	745.67	21.69	0.846	0.429	0.757
	2	721.35	24.25	0.992	0.506	0.741
	4	707.28	25.73	1.105	0.579	0.724
	8 (reference)	703.97	26.07	1.130	0.646	0.710
	16	702.88	26.19	1.141	0.751	0.695

Offline probe count. Increasing M from 100 to 300 improves reward drop from 23.19% to 25.24%. The default $M = 600$ reaches 26.07%, while doubling the probe count to 1200 gives 25.95%. Thus, the tested range $M = 300$–1200 yields reward drops between 25.24% and 26.07%.

These results indicate limited additional benefit from increasing the probe count beyond the default in this setting. They do not establish that the same sample requirement applies to different victims or observation distributions.

PGD step count. The reported step-count sweep increases reward drop from 21.69% at one step to 24.25% at two steps and 25.73% at four steps. Across the tested settings of four to sixteen steps, reward drop lies between 25.73% and 26.19%.

Relative to the eight-step reference, sixteen steps improve reward drop by only 0.12 percentage points, while increasing TempAbs from 0.646 to 0.751 and decreasing SSIM from 0.710 to 0.695. The results show diminishing improvements in return suppression over the tested step-count range.

Interpretation. The sweeps characterize how the evaluated configuration trades attack effectiveness against temporal variation and image similarity. Moderate changes around the default rank and probe count produce relatively small changes in reward drop, whereas the EMA coefficient directly controls a visible strength–smoothness trade-off. These observations are specific to the tested DMC setting and do not establish universal hyperparameter insensitivity.

A.6 Perturbation Visualization

Figure 4 shows perturbation intensity maps for six consecutive Crafter frames under PACE, MAD, PA-AD, and Illusory. PACE produces spatially distributed patterns whose intensities vary across the displayed frames, including a low-intensity map at $t + 4$.

These examples illustrate differences in perturbation structure, but do not identify task-relevant regions or establish temporal smoothness by themselves. We assess temporal variation and image similarity using TempAbs and SSIM in Table 5.

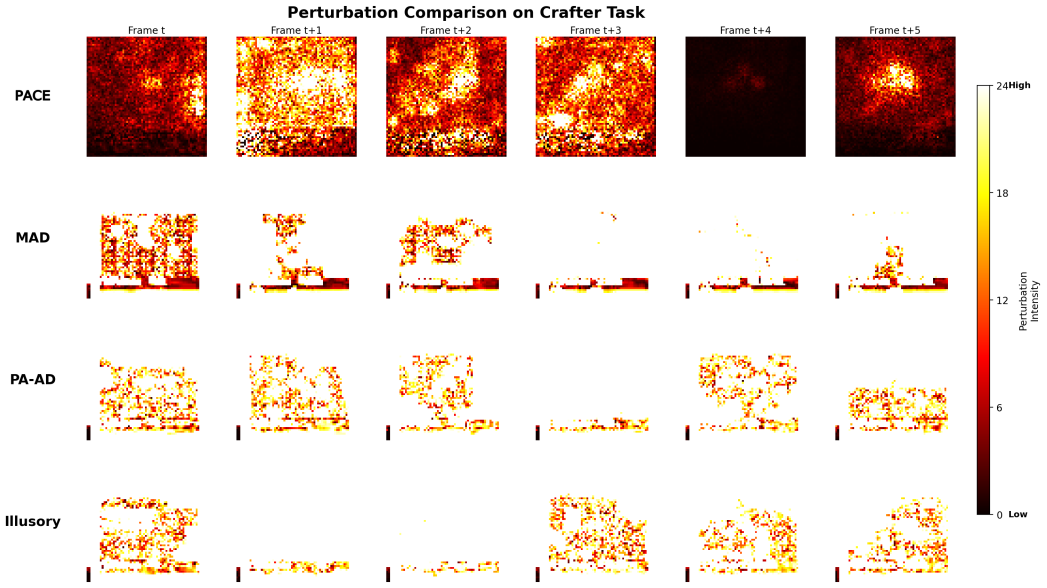

Figure 4: **Perturbation intensity maps on Crafter.** Rows show PACE and three baselines; columns show six consecutive frames. The color bar indicates perturbation intensity on a scale from 0 to 24.

A.7 Attack Formulations and Analysis

We summarize the baseline design principles and clarify the mathematical properties of PACE. The analysis concerns its parameterization and update rules; it does not provide a guarantee of episodic return reduction.

Common notation. Let $\pi_t(o) = \pi_\theta(\cdot \mid o, c_t)$ denote the frozen victim's action distribution under incoming recurrent context c_t. All image perturbations satisfy

$$\mathcal{C}_t = \{\delta : \|\delta\|_\infty \le \varepsilon_{\mathrm{px}},\ 0 \le o_t + \delta \le 255\}. \tag{18}$$

The context is held fixed during each current-frame optimization. The projection $\mathcal{P}_t$ in Eq. (4) maps pixel perturbations into $\mathcal{C}_t$.

A.7.1 Baseline Design Principles

MAD. MAD uses action-distribution deviation as an attack objective, represented by

$$\mathcal{L}_{\mathrm{MAD}}(\delta; o_t, c_t) = \mathrm{KL}(\pi_t(o_t) \,\|\, \pi_t(o_t + \delta)). \tag{19}$$

For diagonal-Gaussian distributions, the KL is

$$\mathrm{KL}(p\|q) = \frac{1}{2}\sum_j \left[\log \frac{\sigma_{q,j}^2}{\sigma_{p,j}^2} + \frac{\sigma_{p,j}^2 + (\mu_{p,j} - \mu_{q,j})^2}{\sigma_{q,j}^2} - 1\right]. \tag{20}$$

UAP-RL. UAP-RL constructs an input-independent perturbation for reuse across observations. Its shared attack parameter is fixed at deployment. However, a fixed parameter does not necessarily imply a constant applied perturbation after image-dependent clipping: $\mathcal{P}_t(\delta^{\mathrm{u}})$ can vary with o_t. Temporal metrics must therefore distinguish the shared parameter from the perturbation actually applied.

PA-AD. PA-AD separates attack generation into a learned director that proposes policy-perturbation directions and an actor that translates these directions into observation perturbations. This differs from directly learning a policy over the full pixel perturbation vector.

Illusory. Illusory attacks explicitly consider information-theoretic detectability in sequential interaction. Their detectability criterion is distinct from perceptual similarity measured by SSIM or a generic image-distance penalty.

DAPGD. DAPGD uses the Bhattacharyya distance to measure policy-distribution differences. For categorical distributions, this distance is

$$\mathcal{L}_{\mathrm{DAPGD}}(\delta; o_t, c_t) = -\log \sum_a \sqrt{\pi_t(o_t)_a \, \pi_t(o_t+\delta)_a}. \tag{21}$$

Both KL and Bhattacharyya distance compare full distributions; they differ in their geometry and gradients.

These methods employ different forms of optimization, reuse, and temporal structure. PACE's defining combination is a critic-guided fixed basis and an observation-dependent coefficient state carried across interaction steps.

A.7.2 PACE: Persistent Adversarial Coefficient Evolution

Categorical-bin surrogate. For the DreamerV3 experiments, PACE minimizes $v(o; c_t) = \sum_{k=0}^{K-1} k\, p(k \mid o, c_t)$, the expected index of the ordered critic bins. If one distribution first-order stochastically dominates another, expectations of increasing functions of the bin index respect that ordering.

The converse does not hold: decreasing the expected index alone does not establish stochastic dominance or guarantee a decrease in decoded value or actual return. We therefore treat this quantity as a differentiable attack surrogate whose effectiveness is evaluated empirically. The transfer experiment uses a target's native differentiable scalar value estimate when its value-head parameterization differs, as described in Section A.3.

What the SVD captures. The rows of Δ are optimized perturbations, not raw critic gradients. For $P_r = V_r V_r^\top$, truncated SVD gives

$$\|\Delta - \Delta P_r\|_F^2 = \sum_{i>r} \sigma_i^2(\Delta). \tag{22}$$

This characterizes reconstruction error on the collected perturbations. It does not guarantee that unseen optimal attacks lie near the same subspace. Likewise, a bound on the encoder Jacobian's rank at one observation does not establish a shared low-rank span across all observations, because the Jacobian changes with its input.

Limits of a subspace approximation. Let $f_t(\delta) = v(o_t + \delta; c_t)$. If f_t is L_v-smooth along the segment between feasible perturbations δ and $\hat{\delta}$, then

$$\begin{aligned} f_t(\hat{\delta}) - f_t(\delta) \le \nabla f_t(\delta)^\top (\hat{\delta} - \delta) \\ + \frac{L_v}{2} \|\hat{\delta} - \delta\|_2^2. \end{aligned} \tag{23}$$

For a constrained optimum, the first-order term need not vanish. Moreover, $P_r\delta$ need not satisfy the pixel constraints. Consequently, the SVD reconstruction identity alone does not imply a quadratic bound on attack suboptimality or a guarantee about episodic return.

Projection and coefficient carryover. PACE alternates coefficient updates with pixel-space clipping and re-encoding:

$$d = V_r \alpha, \qquad \delta = \mathcal{P}_t(d), \qquad \alpha' = V_r^\top \delta. \tag{24}$$

The projected floating-point perturbation δ is feasible, but it can leave $\mathcal{S}_r$. Conversely, reconstructing $V_r\alpha'$ restores subspace membership but can violate pixel constraints, which is why a final pixel projection is required.

This procedure is not an exact projection onto $\mathcal{S}_r \cap \mathcal{C}_t$ and does not implement Dykstra's correction steps. The state passed to the next frame is the coefficient representation of the final projected perturbation.

Temporal smoothing. Before pixel projection, PACE computes

$$\bar{\alpha}_t = \rho\alpha_{t-1} + (1-\rho)\alpha_t^\star. \tag{25}$$

It follows directly that

$$\|\bar{\alpha}_t - \alpha_{t-1}\|_2 = (1-\rho)\|\alpha_t^\star - \alpha_{t-1}\|_2. \tag{26}$$

Since V_r is orthonormal, the same equality holds for the corresponding reconstructed pixel vectors. This moderates the current update relative to the retained state, but does not by itself bound temporal variation after image-dependent clipping.

For an idealized EMA without projection or re-encoding, treating $\alpha_t^\star$ as an input sequence yields

$$H(z) = \frac{1-\rho}{1-\rho z^{-1}}. \tag{27}$$

This is a low-pass filter. The complete PACE procedure also includes warm-started optimization and nonlinear projection, so it is not fully described by this linear transfer function. For a fixed basis, coefficient EMA is equivalent to pixel-space EMA of the corresponding unprojected vectors; its benefit is not a guarantee of preserving attack strength. Section A.5 empirically examines the resulting strength–smoothness trade-off.

Rank selection. A spectral diagnostic for basis compression is

$$E(r) = \frac{\sum_{i=1}^{r}\sigma_i^2(\Delta)}{\sum_{i=1}^{\min(M,D)}\sigma_i^2(\Delta)}. \tag{28}$$

The main and transfer experiments use $r = 32$. Section A.5 additionally evaluates $r \in \{8, 16, 32, 64\}$ through attack-performance measurements.

Spectral energy measures reconstruction of the collected perturbations, rather than attack effectiveness on unseen frames. Selecting rank by an energy threshold is a possible alternative, but requires separate empirical evaluation.

A.8 Limitations and Future Directions

PACE assumes white-box access to the frozen victim, including its critic and observation gradients. It also requires clean rollouts for offline basis construction. The transfer experiment demonstrates reuse of a source basis on the tested Pong victims, but online optimization still requires target-specific value gradients. It therefore does not establish black-box applicability.

The main evaluation covers DreamerV3 on three tasks with five evaluation seeds per task. The supplementary transfer study additionally evaluates another DreamerV3 checkpoint, STORM, and IRIS on Pong. This extends the tested architecture coverage, while leaving generalization across tasks, incompatible observation representations, and environmental distribution shifts unresolved.

The hyperparameter study is limited to DMC Walker Walk. The current ablations examine basis construction, smoothing, and attack signals, but do not separately isolate the effect of warm-starting the coefficient state. Further experiments are needed to determine how these findings change across victim architectures and task distributions.

The efficiency results characterize online attack generation for the tested PyTorch implementations. They exclude offline preparation and do not establish an end-to-end acceleration guarantee or directly measure the runtime of the primary JAX implementation. Likewise, low TempAbs and high SSIM do not establish resistance to automated or human detection.

Future work may investigate query-efficient and black-box attacks, transfer across tasks and observation distributions, adaptation of the basis under distribution shift, and defenses against persistent observation perturbations. Broader evaluation should include additional victim–task combinations, perturbation budgets, uncertainty estimates, and dedicated detectability measurements.

Broader impact. PACE provides a tool for evaluating the vulnerability of recurrent visual agents to temporally linked observation attacks. Such methods may support robustness research but could also be misused against deployed policies. Releasing implementations and evaluation protocols can help researchers reproduce the findings and assess defenses under clearly specified access and perturbation constraints.