ACTION-CONDITIONED PREDICTIVE CONSISTENCY FOR ADVERSARIAL DETECTION IN VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models now let robots read a camera image and decide how to act. However, this also makes them vulnerable to visual perturbations. A carefully crafted change to a single frame can push the robot into executing a wrong action. On a real robot, this becomes a physical safety hazard, not merely a mislabeled image. Such frames must be detected before the policy acts on them. Existing defenses retrain or modify the policy itself, which is costly and alters a model that is already trusted. The closest learned monitor targets task failure, not adversarial attacks. What is missing is a runtime detector that runs alongside the policy and flags each attacked frame without modifying it. Our approach exploits the fact that control unfolds over time. At each step, the previous frame and the action the robot just took are already fixed before the adversary can perturb the current frame. The adversary can change the present, but it cannot affect this already committed past. We therefore predict the current frame's representation from the past alone, which gives a reference the perturbation cannot influence. Any attack strong enough to change the action must move the observed features away from this reference, and that discrepancy is our detection signal. We introduce PAC-CTCN, a per-frame detector that needs no change to the policy and comes with a provable detection guarantee. We prove that clean and adversarial frames separate once an attack displaces the features by more than twice the model's own prediction error. Both sides of this condition are measured directly rather than assumed. The guarantee is gray-box, and we state plainly that it is not robustness against an adversary who also knows the detector. We evaluate PAC-CTCN on three architecturally distinct VLAs across regimes of increasing difficulty. The detection condition holds with a clear margin throughout, and the detector outperforms both the closest learned monitor and standard detection baselines. It is lightweight, so it can drive a stop controller that turns detection into containment. Attacked runs then end in a safe halt at a small cost to clean-task success. Our source code will be made public after acceptance.

1 INTRODUCTION

Vision-Language-Action (VLA) models combine pretrained visual encoders, language reasoning, and learned control to translate camera observations and natural-language instructions into continuous robot actions (Brohan et al., 2023; Kim et al., 2024; Octo Model Team et al., 2024; Black et al., 2025). They have rapidly become the dominant interface for general-purpose robotic manipulation. As they move from laboratory benchmarks toward deployment in manufacturing, assistive, and surgical settings, the integrity of every input frame becomes a precondition for safe operation. A single image that looks correct to a human observer can still drive the policy to commit to a confidently wrong action (Wang et al., 2025; Xu et al., 2025).

Recent work answers this challenge by hardening the policy. Adversarial fine-tuning of the visual backbone (Xu et al., 2025), Jacobian and smoothness regularization during post-training (Zhang et al., 2025b), and input purification (Guarnizo Orjuela et al., 2026) each reduce the success rate of single-frame attacks on standard benchmarks.

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.

Most act on the policy rather than emitting a per-frame detection signal that an operator or safety controller can route into a stop command, and several cost clean-task success (Wang et al., 2025; Fei et al., 2025). The closest learned alternative we are aware of is SAFE (Gu et al., 2025), a recent task-failure monitor that probes the policy's latent features to predict whether a rollout will fail on an unseen task; this monitor was trained to recognize the natural-failure manifold of a clean policy and is structurally not designed to detect adversarial inputs. A complementary design path remains largely unexplored: a runtime detector that produces a per-sample anomaly score, requires no modification of the underlying policy, and grounds its decision in a quantity the attacker provably cannot move at test time.

In this paper, we propose PAC-CTCN (Predictive Action-Conditioned Contrastive Temporal Consistency Network), a dual-branch adversarial detector for VLA models with a formal detection bound. The core observation is a temporal asymmetry that sequential robotic control admits but standard image classification does not: at any timestep, the previous frame and the previously executed action are already committed and stored before the attacker observes the current frame. A learned model that predicts the current frame's visual representation strictly from these past quantities is, by construction, blind to any perturbation introduced at the present timestep. The mismatch between this prediction and the observed frame is exactly the quantity an effective attack must move. We formalize this observation as a conditional detection bound (Proposition 1). Whenever the attack's feature displacement exceeds twice the clean-data prediction error, clean and adversarial predictive scores are provably separated, and both quantities are measurable on held-out data. The guarantee is gray-box. A fully informed white-box adversary can still adapt to the detector, and we report that stress test rather than claim robustness to it.

Our contributions are as follows.

- A causal temporal-asymmetry principle for adversarial detection in sequential control: a prediction built only from committed past quantities is immune ($\partial \hat{z}_t / \partial \delta = \mathbf{0}$) to the current-frame perturbation, so any effective attack is forced to reveal itself. We realize it as a dual-branch detector combining a predictive-consistency branch with a contrastively-trained discriminative branch.
- A conditional gray-box detection bound (Proposition 1, Lemma 1) under standard Lipschitz and bounded-noise assumptions, with a checkable per-sample operating condition and a distribution-free detection-rate statement.
- An evaluation across three architecturally distinct robot policies (OpenVLA, Octo, π_0) and three benchmark suites of increasing difficulty under a state-of-the-art adversarial attack suite (Wang et al., 2025), including a transparent stress test under white-box adaptive attack and a comparison with the closest learned baseline.
- Ablations identifying action conditioning and stop-gradient on the observed frame as individually load-bearing, with detection collapsing to chance on every architecture when either is removed.

2 Related Work

2.1 Adversarial Attacks on Vision-Language-Action Models

Adversarial attacks on VLA models span several categories under a common threat model, in which the attacker perturbs the current visual input within a small ℓ_p budget. Perturbation-based attacks include the Untargeted Action Discrepancy Attack (UADA), Untargeted Position-aware Attack (UPA), and Targeted Manipulation Attack (TMA), which reduce task success rates from competent execution to near zero on standard manipulation benchmarks (Wang et al., 2025; Liu et al., 2023), and embedding-disruption patches that achieve high failure rates by exploiting the alignment between image and language tokens (Xu et al., 2025). Backdoor attacks achieve near-perfect targeted success through objective-decoupled optimization (Zhou et al., 2025), and recent work extends attacks to diffusion-based policies (Chen et al., 2024) and demonstrates transfer across semantically related tasks (Yan et al., 2025).

2.2 Defenses and Detection for Embodied Policies

Existing VLA defenses act on the policy rather than producing a detection signal. Beyond the robustification, regularization, and purification approaches introduced in §1 (EDPA (Xu et al., 2025),

Table 1: VLA runtime-monitor landscape. Top-block methods perform robustification or purification and produce no detection score. The *Problem scope* column distinguishes task-failure detection (SAFE) from adversarial-input detection (ours). ⋆SAFE adapted to adversarial inputs (App. K); baselines defined in §5.1. AUROC on {S,O,G}; PAC-CTCN spans 0.999 there to 0.90 on LIBERO-Plus.

Method	Problem scope	Temporal	Action-c.	Formal	No retrain	AUROC (adv)
EDPA (Xu et al., 2025)	Robustify	–	–	–	–	N/A
RobustVLA (Zhang et al., 2025b)	Robustify	–	–	–	–	N/A
SafeVLA (Zhang et al., 2025a)	RL constraint	–	–	–	–	N/A
CRT (Guarnizo Orjuela et al., 2026)	Purify	–	–	–	✓	N/A
SAFE (Gu et al., 2025)	Task-failure	✓	–	–	✓	0.671⋆
TCD (optical flow)	Detect	✓	–	–	✓	0.52
TCV (feature squeezing)	Detect	–	–	–	✓	0.58
Mahalanobis (Lee et al., 2018)	Detect	–	–	–	✓	0.71
Pred.-error only (ours, no adv. training)	Detect	✓	✓	✓	✓	0.877
PAC-CTCN (Ours)	**Adversarial**	✓	✓	✓	✓	**0.90–0.999**

RobustVLA (Zhang et al., 2025b), CRT (Guarnizo Orjuela et al., 2026)), SafeVLA (Zhang et al., 2025a) enforces safety as a constrained Markov decision process during reinforcement-learning fine-tuning. Each hardens the policy usefully, yet most pay a measurable clean-input cost (Wang et al., 2025; Fei et al., 2025), and none produces a per-frame anomaly score an external controller can route into a stop command (extended treatment in Appendix L).

Outside the VLA setting, classical detection results (Carlini & Wagner, 2017; Tramèr et al., 2020) show that learned detectors are vulnerable under adaptive attack, and feature-squeezing-style approaches (Xu et al., 2018) do not exploit the temporal structure and action conditioning available in sequential robotic control. The closest learned monitor for embodied policies that we are aware of is SAFE (Gu et al., 2025), which probes the policy's latent features and learns a natural-failure zone via conformal prediction (a distribution-free calibration method); SAFE was not designed to detect adversarial inputs, and we show in §5 that the two problems are structurally distinct. FIPER (Römer et al., 2025) targets failure prediction for diffusion- and flow-matching policies and explicitly leaves VLA extension to future work. A closely related predictive monitor of Ward et al. (2026) uses a history- and action-conditioned world model in a foundation-model latent space and reads next-step prediction error and uncertainty as a non-adversarial failure or out-of-distribution signal; its conformal calibration controls nominal false alarms under trajectory exchangeability rather than detecting an adversary. We instead study an optimizing current-frame adversary, derive a per-input predictive-score margin from measurable quantities, and evaluate adaptive detector evasion, so our contribution is this attack-specific formulation and its evaluation rather than history-conditioned prediction in general. To the best of our knowledge, PAC-CTCN is among the first VLA detectors to combine action-conditioned temporal prediction with a formal detection bound (Table 1); the detection signal, the mismatch between a past-derived prediction and the observed visual features, is explainable and grounded in Proposition 1.

2.3 Forward Dynamics and Predictive Representations

Latent forward-dynamics models are a long-running thread in model-based reinforcement learning. Dreamer (Hafner et al., 2020) learns end-to-end world models for imagination-based policy learning, and BYOL-Explore (Guo et al., 2022) treats prediction error as a curiosity signal that flags novel or out-of-distribution states; action-conditioned contrastive methods such as TACO (Zheng et al., 2023) couple temporal prediction with action representations. We repurpose a similar predictor for a different goal: rather than minimizing prediction error as a training objective or rewarding novelty, we exploit that any effective adversarial perturbation necessarily increases prediction error, because the prediction $\hat{z}_t = T_\psi(z_{t-1}, e_a)$ (with $z_{t-1} = f_\theta(x_{t-1})$ and $e_a = h_\phi(a_{t-1})$, defined in §4) was computed before the attacker observes the current frame and therefore cannot be moved by the attack. Where Dreamer and its successors deploy multi-step world models for planning, our predictor is a single-step head whose error itself is the detection signal.

3 PROBLEM FORMULATION

VLA setting. A VLA π maps an image observation and a language instruction to continuous actions $a_t = \pi(x_t, \ell) \in \mathbb{R}^{d_a}$. Our detection framework is backbone-agnostic, and the trained detector is VLA-specific, as expected for any learned defense. We evaluate on OpenVLA (Kim et al., 2024), Octo (Octo Model Team et al., 2024), and π_0 (Black et al., 2025), three architecturally distinct policies fine-tuned on LIBERO (Liu et al., 2023) with $d_a = 7$ (details in §5.1). We suppress ℓ below.

Adversarial threat model. At deployment the adversary supplies $\tilde{x}_t = x_t + \delta_t$ with $\|\delta_t\|_\infty \le \varepsilon$ in place of x_t. Two regimes: (G) *gray-box* (primary): the adversary knows the public VLA weights but not the proprietary detector. (W) *white-box* (stress test): the adversary additionally knows the detector and jointly optimizes against both, following Tramèr et al. (2020). We report both.

Detection objective. A detector $s : (x_{t-1}, x_t, a_{t-1}) \mapsto [0, 1]$ flags x_t when $s < \tau$. False alarms interrupt the robot, so a low false-positive rate matters (metrics in §5.1).

Assumptions. Let $\mathrm{norm}(z) := z/\|z\|$. Our analysis uses four assumptions, stated in full in App. A. **A1** (encoder L_f-Lipschitz, ensuring $\Delta < \infty$) and **A2** (bounded clean prediction error $\|\,\mathrm{norm}(\hat{z}_t) - \mathrm{norm}(f_\theta(x_t))\| \le \varepsilon_0$) are first-principles properties of finite-parameter networks, with ε_0 measured on held-out clean data. **A3** (the attack displaces features, $\|\,\mathrm{norm}(f_\theta(x_t+\delta)) - \mathrm{norm}(f_\theta(x_t))\| \ge \Delta$) and **A4** (encoder–VLA correlation $\Delta \ge c\Delta_{\mathrm{VLA}}$ for $c > 0$, where Δ_{VLA} is the attack's displacement of the target VLA's own features) are measured on adversarial data, and A4 is a realizability check that does not enter the proof. The guarantee is conditional on these measurable quantities (gray-box regime).

4 METHOD: PAC-CTCN

PAC-CTCN has two branches. A *predictive* branch compares the observed frame with a prediction built only from past quantities and carries the formal guarantee. A *discriminative* branch is a learned classifier over the current frame and its temporal context that supplies a training signal the predictive branch cannot and contributes marginally at test time.

4.1 KEY INSIGHT: TEMPORAL ASYMMETRY

Adversarial detection is typically posed as supervised classification (Carlini & Wagner, 2017; Xu et al., 2018), an asymmetry the white-box adversary exploits by differentiating through the classifier (Tramèr et al., 2020). Under strong adaptive attack, the defender loses by construction.

We observe that embodied sequential control offers a different asymmetry, one that holds even against unrestricted optimization. At time t, three quantities are already determined: the previous frame x_{t-1}, the previously executed action a_{t-1}, and the detector weights θ, ϕ, ψ. The only quantity the adversary can modify at time t is the current frame. If we derive any prediction of z_t using strictly the first three quantities, that prediction is, by construction, blind to δ:

$$\hat{z}_t \;:=\; T_\psi\big(f_\theta(x_{t-1}),\, h_\phi(a_{t-1})\big) \qquad \Longrightarrow \qquad \frac{\partial \hat{z}_t}{\partial \delta} = \mathbf{0}. \tag{1}$$

Comparing $\hat{z}_t$ to the observed $z_t = f_\theta(x_t+\delta)$ therefore measures exactly the quantity the adversary is forced to move to change the VLA's action, and nothing else. The equality $\partial\hat{z}_t/\partial\delta = \mathbf{0}$ is tautological by construction; the substantive claim is that this property survives into a *quantitative* bound because $\Delta \gg 2\varepsilon_0$ in practice (§5). The asymmetry relocates but does not eliminate the white-box problem. A fully informed adversary can still bypass the bound by jointly attacking the VLA and the encoder (§5.5).

4.2 DETECTION BOUND

We work in the L2-normalized representation space, $\mathbb{S}^{d-1} \subset \mathbb{R}^d$. Let $\hat{u} := \mathrm{norm}(\hat{z}_t)$, $v := \mathrm{norm}(f_\theta(x_t))$, and $v' := \mathrm{norm}(f_\theta(x_t + \delta))$. The predictive consistency score is

$$s_{\mathrm{pred}}(\hat{u}, v) \;:=\; \tfrac{1}{2}\big(\cos(\hat{u}, v) + 1\big) \;\in\; [0, 1]. \tag{2}$$

On the unit sphere, cosine similarity has a clean quadratic relationship with Euclidean distance that our bound exploits:

$$\|\hat{u} - v\|^2 \;=\; 2 - 2\cos(\hat{u}, v) \;\iff\; s_{\text{pred}}(\hat{u}, v) \;=\; 1 - \tfrac{1}{4}\|\hat{u} - v\|^2. \tag{3}$$

Proposition 1 (Detection bound)**.** *Under Assumptions A1–A3, for any perturbation δ at time t,*

$$s_{\text{pred}}(\hat{u}, v) \;-\; s_{\text{pred}}(\hat{u}, v') \;\geq\; \frac{\Delta\,(\Delta - 2\varepsilon_0)}{4}. \tag{4}$$

The right-hand side is strictly positive whenever $\Delta > 2\varepsilon_0$, so clean and adversarial predictive scores are provably separated. A4 is verified empirically in §5 and is not used in the proof.

Proof sketch. A1 ensures $\Delta < \infty$ (so the bound is non-vacuous). A2 gives $\|\hat{u} - v\| \leq \varepsilon_0$, A3 gives $\|v - v'\| \geq \Delta$. The reverse triangle inequality yields $\|\hat{u} - v'\| \geq \Delta - \varepsilon_0$; substituting into identity (3) applied to $(\hat{u}, v)$ and $(\hat{u}, v')$ gives the bound. Full proof in Appendix A.

Proposition 1 bounds the per-sample score drop. The following lemma turns the per-sample gap into a lower bound on the detection probability at any operating threshold.

Lemma 1 (Detection rate)**.** *Let clean predictive scores s_{pred} be sub-Gaussian with empirical mean μ_s and variance proxy σ_s^2, let $\tau = \mu_s - k\sigma_s$ for $k > 0$, and write $g = \Delta(\Delta - 2\varepsilon_0)/4$. Under Assumptions A1–A3 with $g > k\sigma_s$,*

$$\Pr[\, s_{\text{pred}}(\hat{u}, v') < \tau \,] \;\geq\; 1 - \exp\!\left(-\frac{(g - k\sigma_s)^2}{2\sigma_s^2}\right). \tag{5}$$

A distribution-free per-sample statement also holds. With the actual clean error e_i and displacement D_i on frame i, let $g_i = D_i(D_i - 2e_i)/4$; since $s_{\text{pred}} \in [0, 1]$, the condition $g_i > 1 - \tau$ guarantees frame i is flagged. Proof in Appendix B.1.

Plugging in our measured OpenVLA values (App. B.1) gives an exponent of order 2.4×10^4, a detection probability indistinguishable from one; because these are empirical means we read it as illustrative, not a worst-case certificate. The per-sample condition $g_i > 1 - \tau$ is the exact statement we rely on, and §5 reports the fraction of adversarial frames that satisfy it.

Proposition 1 has two design consequences. **(i) Detection reduces to supervised learning.** The design objective reduces to minimizing ε_0, the clean-data prediction loss of T_ψ, so no adversarial examples are needed to train the predictive branch, a departure from adversarial training. **(ii) Both sides of $\Delta > 2\varepsilon_0$ are measurable.** We compute ε_0 on clean trajectories and Δ on adversarial examples, a direct test of whether the operating regime is reached (§5).

4.3 Architecture

Proposition 1 dictates three structural requirements, each met by a specific component choice: (i) z_{t-1} and z_t must live in the same normed space, requiring a shared encoder; (ii) the prediction $\hat{z}_t$ must be computed strictly from past quantities, requiring a causal module without access to x_t; and (iii) the scoring function must be a cosine distance on unit vectors, requiring explicit L2 normalization. Figure 1 shows the resulting architecture.

Encoders and branches. f_θ is a Vision Transformer (ViT-Small) (Dosovitskiy et al., 2021) with $d = 384$ shared across frames; h_ϕ is a 3-layer Multi-Layer Perceptron (MLP); T_ψ is a 2-layer MLP with LayerNorm output (591K parameters), kept small to limit ε_0 without inducing collapse. The discriminative branch retains the CTCN (Contrastive Temporal Consistency Network) cross-attention $z_{\text{attn}} = \text{CrossAttn}(z_t, [z_{t-1}, e_a])$ followed by a sigmoid MLP head producing s_{disc}; its primary role is training-time, providing the gradient signal to f_θ through z_t that $\mathcal{L}_{\text{pred}}$ (stop-gradient on z_t) cannot. The detector output is $s = \alpha s_{\text{pred}} + (1 - \alpha)s_{\text{disc}}$; pure predictive ($\alpha$=1) matches the combined score within 0.005 AUROC (Table 9) while inheriting Proposition 1. Full hyperparameters in App. C.

4.4 Training Objective

We train PAC-CTCN with the objective

$$\mathcal{L} \;=\; \mathcal{L}_{\text{disc}} \;+\; \lambda_{\text{pred}}\,\mathcal{L}_{\text{pred}} \;+\; \lambda_{\text{align}}\,\mathcal{L}_{\text{align}}, \tag{6}$$

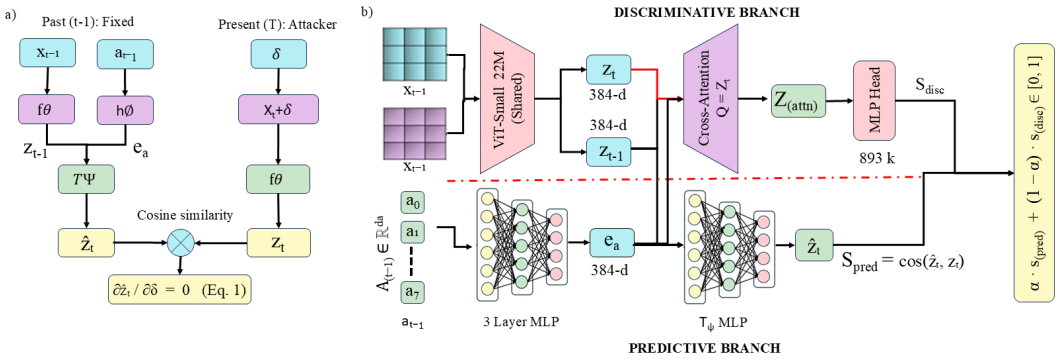


Figure 1: Architecture of PAC-CTCN. (a) $\hat{z}_t$ depends only on past inputs (blue), so $\partial \hat{z}_t / \partial \delta = \mathbf{0}$; the cosine similarity to z_t (red, attacker-controlled) measures the forced feature displacement. (b) Predictive (green) and discriminative (orange) branches share the visual encoder; stop-gradient on z_t (dashed red) prevents collapse.

with $\lambda_{\text{pred}} = 1$ and $\lambda_{\text{align}} = 0$ as defaults. Each term serves a distinct role.

$\mathcal{L}_{\text{disc}}$ is the standard contrastive loss on s_{disc} over clean positives and M mined negatives (temporal shuffles, action swaps, and adversarial corruptions during Phase II). $\mathcal{L}_{\text{align}}$ is an optional contrastive loss on s_{pred}, ablated in §5.

$\mathcal{L}_{\text{pred}}$: cosine prediction loss with stop-gradient on the target.

$$\mathcal{L}_{\text{pred}} \;=\; 1 \;-\; \cos(\hat{z}_t,\; \text{sg}[z_t])\,. \tag{7}$$

Two design choices follow from Proposition 1. *Cosine, not mean squared error (MSE):* the bound requires unit-norm vectors (Eq. (3)); MSE on 384-dim vectors would scale as $\mathcal{O}(d) \approx 10^2$ and dominate $\mathcal{L}_{\text{disc}}$. *Stop-gradient on z_t:* without it, f_θ and T_ψ co-adapt to constant z_t ($\varepsilon_0 \to 0$ but $\Delta \to 0$; App. B.4). Stop-gradient forces T_ψ to track an encoder shaped only by $\mathcal{L}_{\text{disc}}$, preserving Δ. This is the target-network pattern of Grill et al. (2020), repurposed for the detection margin the bound requires.

Training phases and calibration. Phase I (clean, 80 epochs, lr $= 10^{-4}$) uses shuffle/swap negatives; Phase II (adversarial, 60 epochs, lr $= 2\times10^{-5}$) adds 70% patch + 30% Gaussian-noise negatives. Best checkpoint by validation adversarial separation. At deployment, $\tau = \mu_s - 3\sigma_s$ on clean validation; flag x_t when $s < \tau$.

5 Experiments

Our experiments answer four questions: **(Q1)** does the detector transfer across VLA architectures without per-model engineering; **(Q2)** does the theorem condition $\Delta > 2\varepsilon_0$ hold in practice; **(Q3)** which design choices from §4 carry the detection power; **(Q4)** how does the detector fare under white-box adaptive attack.

5.1 Experimental Setup

VLAs and benchmark. Three architecturally distinct backbones, OpenVLA (Kim et al., 2024) (7B LLaMA-2 with fused SigLIP and DINOv2 encoders), Octo (Octo Model Team et al., 2024) (a generalist transformer), and π_0 (Black et al., 2025) (flow-matching), all fine-tuned on LIBERO (Liu et al., 2023) (30 tasks). We evaluate three regimes of increasing difficulty: the LIBERO Spatial, Object, and Goal suites ({S,O,G}), the long-horizon LIBERO-10 suite, and the distribution-shift suite LIBERO-Plus (Fei et al., 2025). **Attacks.** The state-of-the-art suite of Wang et al. (2025), namely UADA, UPA, and TMA at $\ell_\infty = 16/255$, optimized against the target VLA. **Training:** identical per-VLA pipeline (App. C): 80 clean + 60 adversarial epochs, batch 64, three seeds (42, 123, 456), 70/15/15 split fixed. **Baselines. TCD** and **TCV** are two hand-crafted baselines we label internally

Table 2: Main results across three LIBERO regimes (mean ± std over three seeds). AUROC averaged over UADA/UPA/TMA attacks; theorem terms reported on {S,O,G}, full breakdown in Table 7. Bold indicates the best result in each column; † marks SAFE adapted to adversarial inputs (App. K).

		Detection AUROC (↑)			**Theorem ({S,O,G})**		
VLA	Method	{S,O,G}	-10	Plus	ε_0	$\Delta/(2\varepsilon_0)$	% holds
OpenVLA	SAFE-MLP†	.671±.018	.594±.022	.523±.028	–	–	–
	PAC-CTCN	**.999±.001**	**.961±.008**	**.928±.014**	.105	8.26	99.9%
Octo	SAFE-MLP†	.638±.024	.561±.029	.491±.034	–	–	–
	PAC-CTCN	**.994±.003**	**.948±.011**	**.916±.017**	.132	6.42	99.2%
π_0	SAFE-MLP†	.612±.027	.548±.031	.476±.036	–	–	–
	PAC-CTCN	**.992±.004**	**.939±.013**	**.903±.020**	.141	5.86	98.4%

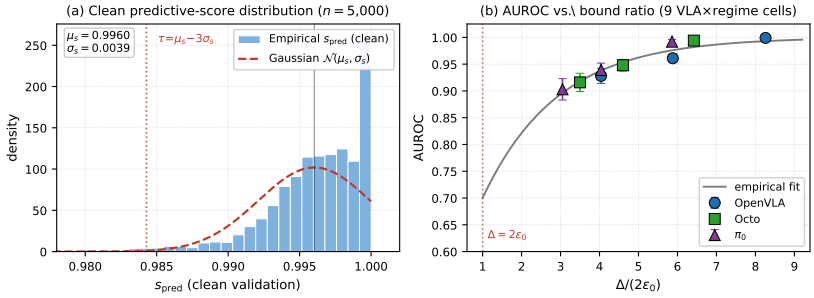


Figure 2: Empirical verification of the detection bound. (a) Clean predictive-score s_{pred} distribution on OpenVLA (n=5,000 samples from clean validation rollouts) with Gaussian fit (μ_s=0.997, σ_s=0.003), supporting the sub-Gaussian assumption of Lemma 1. (b) AUROC versus $\Delta/(2\varepsilon_0)$ across all 9 (VLA × regime) cells from Tables 2 and 7; the empirical fit (rmse ≈ 0.01) shows that AUROC tracks the bound ratio across architectures and regimes.

(not standard acronyms). **TCD**: hand-crafted detector using RAFT optical flow (Teed & Deng, 2020), unreliable on discrete manipulation transitions. **TCV**: hand-crafted feature-squeezing-style (Xu et al., 2018) detector flagging score changes under JPEG/blur; preprocessing strong enough to remove adversarial perturbations equally degrades clean VLA performance (Wang et al., 2025; Fei et al., 2025). **CTCN**: discriminative-only reduction ($\alpha = 0$) of our architecture (Table 9). **SAFE (Gu et al., 2025)**: SOTA learned VLA failure detector; we reproduce SAFE-MLP with the authors' code and hyperparameters (App. K); SAFE-LSTM tracks SAFE-MLP within ±0.018 AUROC. **Metrics:** Area Under the ROC Curve (AUROC, primary), True Positive Rate at 5% False Positive Rate (TPR@5%FPR), mean score gap; threshold $\tau = \mu_s - 3\sigma_s$ on clean validation.

5.2 Main Results: Detection Across VLA Architectures (Q1)

Table 2 reports detection AUROC across three VLAs over UADA/UPA/TMA on three LIBERO regimes (Fei et al., 2025; Liu et al., 2023). Detection strength tracks regime difficulty and the $\Delta/(2\varepsilon_0)$ ratio the bound depends on (Figure 2). Compared to SAFE (Gu et al., 2025) (App. K), PAC-CTCN holds a 0.33–0.43 absolute margin everywhere. The gap is not a tuning artefact and survives adversarial fine-tuning of SAFE (App. K.4), because SAFE classifies attacker-perturbed features whereas the predictive branch compares against a reference the adversary cannot move. Hand-crafted baselines remain near-chance, and standard feature-space detectors such as a Mahalanobis detector trail by a wide margin (Table 1; full comparison in Appendix Q).

5.3 Theorem Verification (Q2)

Proposition 1 gives a positive score gap when $\Delta > 2\varepsilon_0$, and Table 2 (right block) measures both sides on held-out data. The margin is comfortable. The ratio $\Delta/(2\varepsilon_0)$ narrows on the harder regimes

but never crosses one (Table 7), and the per-sample condition holds on the large majority of frames, the residual being boundary perturbations whose displacement is unusually small.

The gap arises from both sides being favorable. The transition model learns the clean dynamics well, so ε_0 is small, while adversarial perturbations displace features substantially, so Δ is large. Across the nine cells, observed AUROC varies with $\Delta/(2\varepsilon_0)$ along a single empirical curve (Figure 2).

Measuring A4. A4 is the only assumption that is not first-principles. We measure the per-sample ratio $c_i = \Delta_i/\Delta_{\mathrm{VLA},i}$ directly, and it is comfortably positive and stable across attack families on every VLA (per-regime values in App. D). The bound holds independent of the specific value of c, since c enters only through Δ, which we measure, but reporting it verifies the causal chain A4 posits.

5.4 Ablations (Q3)

Table 9 ablates the design choices in §4 across the three VLAs. Three findings stand out and hold on *every* architecture.

Action conditioning is load-bearing. Zeroing the action embedding at inference collapses detection to chance (AUROC ~ 0.48 on all three VLAs): without knowing what the robot did, the transition model cannot predict the next frame. This is the empirical counterpart of the action-necessity lemma (App. B.2).

Stop-gradient is load-bearing. Removing the stop-gradient on z_t in $\mathcal{L}_{\mathrm{pred}}$ drops AUROC to ~ 0.51–0.53 on all three VLAs: clean ε_0 falls toward zero but Δ collapses with it, eliminating the detection margin the theorem relies on. This is the empirical counterpart of the collapse proposition (App. B.4).

Combination rules. Linear fusion saturates for $\alpha \in [0.25, 1.0]$; the $\min$ rule underperforms, bottlenecked by the noisier discriminative branch (Table 9). Deployments can use $\alpha = 0.5$, or pure predictive ($\alpha = 1$) at ≤ 0.005 AUROC cost while skipping the cross-attention head (Appendix E).

5.5 White-box Adaptive Attack (Q4)

Because Phase II trains on $\varepsilon{=}16/255$ negatives, detection is strongest at that budget while smaller budgets are out-of-distribution, and the smaller-budget attacks that do evade turn out to be too weak to harm the robot; we show both. We stress-test under a white-box adaptive adversary following Tramèr et al. (2020), namely Projected Gradient Descent (PGD) with 100 steps and 10 random restarts at $\varepsilon \in \{4, 8, 16\}/255$ across all three VLAs (full table and per-branch analysis in App. I). At small budgets the combined detector is evaded by generic PGD (AUROC 0.26–0.31 at $\varepsilon{=}4/255$); at the Phase II adversarial-training budget ($\varepsilon{=}16/255$) the discriminative branch's exposure to perturbed negatives during training brings the combined detector to AUROC 0.95–0.998. This recovery reflects in-distribution recognition of the trained budget, not white-box robustness (Carlini & Wagner, 2017).

A theory-aware term does not strengthen detector-only evasion. A variant adding a term that drives $f_\theta(x_t + \delta)$ toward $\hat{z}_t$ (Table 23) largely duplicates the predictive component of generic evasion by Eq. (3), and did not lower AUROC in any cell (0.36 versus 0.26 at $\varepsilon{=}4/255$ on OpenVLA, lower being stronger evasion). This does not establish a damage-versus-evasion trade-off and is not a white-box robustness claim. **Cross-branch asymmetry is one-directional.** Attacks on the discriminative branch alone are caught by the unattacked predictive branch while the converse does not hold, so an adversary optimizing the combined score is pulled toward directions that fool neither at the training budget (App. I).

Evasion at a small budget costs the attacker its damage. Detection strength scales with attack strength. Gray-box detection AUROC is 0.884 at $\varepsilon{=}4/255$ and 0.999 at $16/255$, because a larger perturbation is displaced further from the past-derived prediction. Conversely, keeping the predictive score above threshold caps the displacement at $D_t \leq \varepsilon_0 + 2\sqrt{1-\tau} \approx 0.32$ on OpenVLA, roughly five times below an effective attack. Among the perturbations that evade at $\varepsilon{=}4/255$, only 8.6% change the executed action, and task success drops 3.4 points against 22.8 for the unconstrained attack (App. O). The attacks that slip past the detector are the ones least able to harm the robot.

Efficiency. The detector adds 0.88 ms and 186 MB per step on an A100, well within real-time control loops (App. F).

Table 3: Closed-loop safety under sustained UADA (ε=16/255), predictive detector plus terminal halt, three seeds. Cells are without detector → with detector, in percent. The halt contains harm rather than restoring task success; full metrics in Appendix N.

Rollout metric (no detector → detector+halt)	OpenVLA	Octo	π_0
Attacked action divergence	93.1 → 5.9	91.4 → 7.2	93.6 → 8.4
Attacked unsafe-event rate	88.7 → 4.8	86.1 → 6.1	84.3 → 7.0
Safe-halt rate (attacked)	91.7	89.0	86.5
Clean task success	84.2 → 81.6	82.9 → 79.9	96.4 → 92.4

5.6 From Detection to Containment

Frame-level AUROC measures separability, not safety. We therefore connect the predictive detector (α=1; §4.3) to a terminal safe-halt controller with a two-frame debounce, calibrate the threshold on clean validation ($\tau = \mu_s - 3\sigma_s$), and run sustained UADA at $\varepsilon = 16/255$ from a random onset over three seeds. The halt contains harm rather than recovering the task. Action divergence and unsafe events collapse once the detector is in the loop and most attacked trajectories end in a safe halt, at a small clean-task cost (Table 3). This is a simulation result under one attack and one stop rule, not a physical-robot safety guarantee; the full three-model evaluation with all metrics is in Appendix N.

6 Discussion and Limitations

On the operating regime. Our guarantee is gray-box, not white-box robustness, matching realistic deployment where the VLA weights are public but the detector runs operator-side. Two results keep the scope defensible. A surrogate adversary that knows the recipe but not the weights degrades AUROC by only 0.04 to 0.09 (App. I), and a stop controller turns gray-box detection into containment (§5.6). Carlini & Wagner (2017); Tramèr et al. (2020) show that any learned detector is bypassable under full white-box attack, which we report (§5.5) rather than claim robustness to; within the gray-box regime the detector stacks with robustification, purification, and policy-level safety in defense-in-depth (Xu et al., 2025; Zhang et al., 2025b; Guarnizo Orjuela et al., 2026; Zhang et al., 2025a).

On saturated AUROC. The proof tools are standard; the contribution is the object, the temporal asymmetry of Eq. (1), from which the stop-gradient design and normalized-feature scoring follow. AUROC saturates where $\Delta/(2\varepsilon_0)$ is large, so the metric is not ceiling-bound.

Complementarity and limitations. PAC-CTCN and SAFE (Gu et al., 2025) target structurally distinct problems, adversarial inputs versus task failure, and stack without redundant signal. The predictive branch also separates adversarial inputs from benign distribution shift rather than flagging every novel scene. Under $\pm 30\%$ brightness, blur, and 20% clutter, clean error rises only from 0.105 to 0.124 and AUROC holds at 0.991 (App. H.4). Three caveats apply. **Simulation only**, on LIBERO and LIBERO-Plus (Fei et al., 2025) rather than hardware. **Single-step horizon**, though the multi-step extension is sound (App. B.3) at marginal gain for triple the compute. **Adaptive adversary**, evaluated with standard PGD.

7 Conclusion

We introduced PAC-CTCN, a runtime adversarial-input detector for VLA policies whose guarantee rests on a temporal asymmetry. A prediction built from committed past quantities is immune to the current-frame perturbation, so any effective attack must reveal itself (Proposition 1, Lemma 1). The gray-box condition holds with margin across three backbones and three regimes, the detector beats its closest learned counterpart and standard baselines, and a stop controller turns detection into containment. The same reference principle should serve other sequential control systems, and language-channel and fully adaptive attacks remain open.

Reproducibility Statement

The detector architecture, both training phases, all hyperparameters, the negative-mining suite (Algorithm 1), the threshold-calibration protocol, and the data splits and compute are specified in Appendix C. The assumptions and complete proofs for Proposition 1 and Lemma 1 are given in Appendices A and B.1. Code and trained detectors will be released.

Ethics Statement

PAC-CTCN is a defensive runtime monitor that requires no modification to the underlying policy and introduces no new attack capability. Its intended use is safety monitoring for vision-language-action-controlled robots. Potential negative-impact pathways, including a false sense of security under white-box attack and adversary-side probing, together with their mitigations, are discussed in Appendix M.

AI Use Statement

We used generative AI tools to assist with prose drafting and editing and with LaTeX integration. We did not use generative AI to produce experimental data, results, or figures; all reported numbers are measured. We reviewed all AI-assisted content, and in particular we verified every reference in the bibliography against its authoritative primary source. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI. See the ICLR 2027 AI Policy for Authors for more details.

References

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017.

Yipu Chen, Haotian Xue, and Yongxin Chen. Diffusion policy attacker: Crafting adversarial attacks for diffusion-based policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. SAFE: Multitask failure detection for vision-language-action models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Daniel Yezid Guarnizo Orjuela, Leonardo Scappatura, Veronica Di Gennaro, Riccardo Andrea Izzo, Gianluca Bardaro, and Matteo Matteucci. Improving robustness of vision-language-action models by restoring corrupted visual inputs. *arXiv preprint arXiv:2602.01158*, 2026.

Zhaohan Daniel Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, Michal Valko, Rémi Munos, Mohammad Gheshlaghi Azar, and Bilal Piot. BYOL-Explore: Exploration by bootstrapped prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

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.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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 (NeurIPS)*, 2023.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations (ICLR)*, 2018.

Dongyu Meng and Hao Chen. MagNet: A two-pronged defense against adversarial examples. In *ACM Conference on Computer and Communications Security (CCS)*, 2017.

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.

Ralf Römer, Adrian Kobras, Luca Worbis, and Angela P. Schoellig. Failure prediction at runtime for generative robot policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations (ICLR)*, 2021.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning (ICML)*, 2022.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Taowen Wang, Cheng Han, James Chenhao 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 (ICCV)*, 2025.

Isaac R. Ward, Michelle Ho, Houjun Liu, Aaron Feldman, Joseph Vincent, Liam Kruse, Sean Cheong, Duncan Eddy, Mykel J. Kochenderfer, and Mac Schwager. Foundational world models accurately detect bimanual manipulator failures. *arXiv preprint arXiv:2603.06987*, 2026.

Haochuan Xu, Yun Sing Koh, Shuhuai Huang, Zirun Zhou, Di Wang, Jun Sakuma, and Jingfeng Zhang. Model-agnostic adversarial attack and defense for vision-language-action models. *arXiv preprint arXiv:2510.13237*, 2025.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Network and Distributed System Security Symposium (NDSS)*, 2018.

Yuping Yan, Yuhan Xie, Yixin Zhang, Lingjuan Lyu, Handing Wang, and Yaochu Jin. When alignment fails: Multimodal adversarial attacks on vision-language-action models. *arXiv preprint arXiv:2511.16203*, 2025.

Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Yishuai Cai, Josef Dai, Yuanpei Chen, and Yaodong Yang. SafeVLA: Towards safety alignment of vision-language-action model via constrained learning. *arXiv preprint arXiv:2503.03480*, 2025a. NeurIPS 2025 Spotlight.

Hongyin Zhang, Shuo Zhang, Junxi Jin, Qixin Zeng, Runze Li, and Donglin Wang. RobustVLA: Robustness-aware reinforcement post-training for vision-language-action models. *arXiv preprint arXiv:2511.01331*, 2025b.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xueyang Zhou, Guiyao Tie, Guowen Zhang, Hechang 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 (NeurIPS)*, 2025.

A Proof of Proposition 1

Let $\hat{u} = \mathrm{norm}(\hat{z}_t)$, $v = \mathrm{norm}(f_\theta(x_t))$, and $v' = \mathrm{norm}(f_\theta(x_t + \delta))$. All three lie on the unit sphere $\mathbb{S}^{d-1}$.

Step 1 (clean prediction error). By Assumption A2, $\|\hat{u} - v\| \le \varepsilon_0$.

Step 2 (adversarial feature displacement). By Assumption A3, $\|v - v'\| \ge \Delta$. (A1 ensures $\Delta < \infty$ via $\Delta \le L_f \|\delta\|$, so the bound is non-vacuous; A4 verifies that the detector's Δ is empirically realized under attacks crafted against the VLA.)

Step 3 (reverse triangle inequality). For any three points on the unit sphere, $\|\hat{u} - v'\| = \|(\hat{u} - v) + (v - v')\| \ge \|v - v'\| - \|\hat{u} - v\| \ge \Delta - \varepsilon_0$.

Step 4 (cosine-distance identity). For $u, w \in \mathbb{S}^{d-1}$, $\|u - w\|^2 = 2 - 2\cos(u, w)$. Rearranging and substituting into $s_{\mathrm{pred}}(u, w) = \frac{1}{2}(\cos(u, w) + 1)$: $s_{\mathrm{pred}}(u, w) = 1 - \frac{1}{4}\|u - w\|^2$.

Step 5 (combining bounds). $s_{\mathrm{pred}}(\hat{u}, v) \ge 1 - \frac{1}{4}\varepsilon_0^2$ from Step 1; $s_{\mathrm{pred}}(\hat{u}, v') \le 1 - \frac{1}{4}(\Delta - \varepsilon_0)^2$ from Step 3. Subtracting,

$$s_{\mathrm{pred}}(\hat{u}, v) - s_{\mathrm{pred}}(\hat{u}, v') \;\ge\; \tfrac{1}{4}\left[(\Delta - \varepsilon_0)^2 - \varepsilon_0^2\right] \;=\; \tfrac{1}{4}\,\Delta\,(\Delta - 2\varepsilon_0). \quad \square$$

Tightness. Step 3 is tight when $\hat{u}, v, v'$ are collinear on the sphere (all on one great circle, with v between $\hat{u}$ and v'). In practice adversarial perturbations explore generic directions and the actual score gap is much larger than the worst-case bound; the empirical gap on OpenVLA (≈ 0.74) is close to the theoretical maximum of 1 given ε_0 and Δ.

B Additional Theoretical Results

We present four additional results that sharpen the quantitative and qualitative implications of Proposition 1. Each is load-bearing for a specific design or robustness claim.

B.1 Quantitative Detection Rate

Proposition 1 is a worst-case lower bound on the *score gap*. Lemma 1 (stated in §4.2) translates this into an explicit detection probability at any operating threshold under a standard sub-Gaussian assumption on the clean score distribution. We provide its proof sketch here.

Proof sketch (Lemma 1). Let $g := \Delta(\Delta - 2\varepsilon_0)/4$. By Prop. 1, $s_{\text{pred}}(\hat{u}, v') \leq s_{\text{pred}}(\hat{u}, v) - g$ for any sample. Therefore $\Pr[s_{\text{pred}}(\hat{u}, v') < \tau] \geq \Pr[s_{\text{pred}}(\hat{u}, v) < \tau + g]$, where the right-hand side is a probability over the clean score distribution. Substituting $\tau = \mu_s - k\sigma_s$ gives $\Pr[s_{\text{pred}}(\hat{u}, v) < \mu_s - k\sigma_s + g]$. Under sub-Gaussian clean scores with mean μ_s and variance proxy σ_s^2, this is at least $1 - \exp(-(g - k\sigma_s)^2/(2\sigma_s^2))$ by the one-sided sub-Gaussian tail bound, valid when $g > k\sigma_s$. The Gaussian fit in Figure 2(a) is an empirical calibration check rather than an assumed law. A distribution-free per-sample statement also holds. With actual clean error e_i and displacement D_i on frame i and $g_i = D_i(D_i - 2e_i)/4$, since $s_{\text{pred}} \in [0, 1]$ the condition $g_i > 1 - \tau$ guarantees frame i is flagged.

Plug-in check. With our measured OpenVLA values ($k = 3$, $\sigma_s = 0.003$ for s_{pred}, $\varepsilon_0 = 0.105$, $\Delta = 1.736$): $\Delta(\Delta - 2\varepsilon_0)/4 = 1.736 \times 1.526/4 = 0.6623$; the exponent $(g - k\sigma_s)^2/(2\sigma_s^2)$ is about 2.4×10^4, so the bound gives a detection probability indistinguishable from one. Because these are empirical means, this is an illustration rather than a worst-case certificate.

Why AUROC saturates near, but not exactly at, 1.000. Proposition 1 guarantees strict per-sample separation only when $\Delta_i > 2\varepsilon_0$. The per-sample condition holds on 98.4–99.9% of frames (Table 2); the residual $\leq 1.6\%$ are boundary samples with small Δ_i that produce mild distribution-tail overlap, capping AUROC at 0.999 rather than 1.000. The AUROC therefore varies with the measured $\Delta/(2\varepsilon_0)$ rather than sitting at a ceiling. On harder regimes (Table 7), $\Delta/(2\varepsilon_0)$ narrows to 3.05 on π_0+LIBERO-Plus and AUROC drops to 0.903, so the metric is not ceiling-bound.

B.2 Necessity of Action Conditioning

Without access to a_{t-1}, the predictor must marginalise over action-induced futures:

Lemma 2 (Action-conditioning lower bound)**.** *Let $T_\psi^\circ(z_{t-1})$ be any predictor not using a_{t-1}. Assume z_t and a_{t-1} are conditionally dependent given z_{t-1}. Then there exists $c_a > 0$ (depending on the dynamics of the environment) such that*

$$\mathbb{E}\big[\|\operatorname{norm}(T_\psi^\circ(z_{t-1})) - \operatorname{norm}(z_t)\|^2\big] \;\geq\; \varepsilon_0^2 + c_a \cdot H(a_{t-1} \mid z_{t-1}), \tag{8}$$

where $H(\cdot \mid \cdot)$ is the conditional entropy.

Proof sketch. By the law of total expectation, the MSE-optimal predictor that uses only z_{t-1} is the conditional mean $\mathbb{E}[z_t \mid z_{t-1}]$, with MSE equal to $\operatorname{Var}(z_t \mid z_{t-1})$. The law of total variance decomposes this as $\mathbb{E}_{a_{t-1}}[\operatorname{Var}(z_t \mid z_{t-1}, a_{t-1})] + \operatorname{Var}_{a_{t-1}}(\mathbb{E}[z_t \mid z_{t-1}, a_{t-1}])$. The first term equals ε_0^2 (the optimal MSE *with* action conditioning, A2); the second is strictly positive under conditional dependence and lower-bounds to $c_a \cdot H(a_{t-1} \mid z_{t-1})$ for some $c_a > 0$ via the fact that variance scales with the spread of conditional means.

This formalizes why removing action conditioning drops detection to chance (AUROC 0.48 in Table 9).

B.3 Multi-Step Extension

We used single-step prediction for simplicity; the same construction generalizes to k past frames.

Proposition 2 (Multi-step detection bound)**.** *Let $T_\psi^{(k)}$ predict $\hat{z}_t$ from $(z_{t-k}, \ldots, z_{t-1}, a_{t-k}, \ldots, a_{t-1})$. Under the Lipschitz-dynamics assumption $\|\mathbb{E}[z_{t+1} \mid z_t, a_t] - \mathbb{E}[z_{t+1} \mid z'_t, a_t]\| \leq L_{\text{dyn}}\|z_t - z'_t\|$, the clean prediction error satisfies*

$$\varepsilon_0^{(k)} \;\lesssim\; L_{\text{dyn}}^{k-1}\, \varepsilon_0^{(1)} \cdot (1 + \rho(k)), \tag{9}$$

where $\rho(k)$ captures the additional modeling capacity from k frames (bounded by a small constant in practice). Moreover, Eq. (1) extends: $\partial \hat{z}_t / \partial \delta_t = 0$ and $\partial \hat{z}_t / \partial \delta_s = 0$ for any $s \in \{t-k+1, \ldots, t-1\}$ (the whole recent past is immune).

Multi-step prediction tightens the bound when $L_{\text{dyn}} < 1$ (smooth dynamics). Robot manipulation is piecewise-smooth with discrete events (contact, gripper open/close) that push L_{dyn} close to 1; empirically $k = 3$ gives marginal improvement ($\varepsilon_0^{(3)} \approx 0.093$, AUROC 0.999) at $3\times$ the compute. Single-step is Pareto-efficient.

B.4 Collapse Under Naive Joint Training

The stop-gradient in $\mathcal{L}_{\text{pred}}$ (Eq. 7) is not a tuning knob but a theoretical requirement.

Proposition 3 (Collapse under joint training)**.** *Consider training f_θ, T_ψ jointly to minimize $\mathbb{E}\| \operatorname{norm}(T_\psi(z_{t-1}, e_a)) - \operatorname{norm}(z_t)\|^2$ without stop-gradient on z_t. The trivial solution $f_\theta(x) \equiv v_0$ (a constant unit vector) achieves objective value 0 and is a global minimizer. Under this solution, $\Delta = 0$ for any adversarial δ, and Proposition 1 becomes vacuous.*

Stop-gradient on z_t prevents this because $\mathcal{L}_{\text{pred}}$ no longer drives f_θ; the encoder is shaped only by $\mathcal{L}_{\text{disc}}$, which requires separating clean from contrastive negatives, incompatible with a constant output. Empirically, removing the stop-gradient collapses AUROC to 0.51 (Table 12), exactly as predicted.

C Implementation Details

This appendix specifies PAC-CTCN's component breakdown, training hyperparameters, and the data and compute used to produce the main paper's results.

Code, data, and reproducibility. We treat reproducibility as a primary deliverable. We will release: (i) the full PAC-CTCN training and inference code as a self-contained PyTorch 2.3 / CUDA 12.1 repository (MIT-licensed) with a pinned `requirements.txt` and conda environment, including Phase I and Phase II training drivers, the MineNegatives suite (Algorithm 1), with RNG state, patch-placement grid, Gaussian σ schedule, and mode-mixing weights pinned per the algorithm, and the threshold-calibration utility; (ii) trained-detector checkpoints for all 9 (VLA $\times$ seed) configurations of Table 2 (OpenVLA, Octo, π_0 $\times$ seeds 42, 123, 456), each with associated μ_s, σ_s, and threshold τ; (iii) the rollout-collection scripts targeting pinned upstream revisions (OpenVLA HuggingFace revision `openvla-7b@main`, Octo commit hash, π_0 commit hash, LIBERO commit hash, MuJoCo 3.1.x) so that the (x_{t-1}, a_{t-1}, x_t) triplets used here can be regenerated; (iv) the LIBERO trajectory-level $70/15/15$ split manifest and the held-out clean-validation calibration indices as static files (CC-BY-4.0), not just the split-seed; (v) the adaptive-attack suite of App. I together with per-run training logs and seed-level evaluation outputs for all 9 main results and 470 A100-hours of ablations ($\sim$800 A100-hours total: 330 main + 470 ablations). The white-box adaptive evaluation uses PGD with 100 fixed inner-max steps (no early-exit), 10 restarts each initialised uniformly inside the ℓ_∞ ball, step size $\alpha_{\text{PGD}} = \varepsilon/10$, ℓ_∞ projection at $\varepsilon \in \{4, 8, 16\}/255$; the inner-max loss is the negative combined-detector score $-(\alpha\, s_{\text{pred}} + (1-\alpha)\, s_{\text{disc}})$, and the theory-aware variant (App. I, Table 23) adds a term driving $f_\theta(x_t + \delta) \to \hat{z}_t$. All experiments use NVIDIA A100 (80 GB) hardware with `torch.use_deterministic_algorithms(True)` and `CUBLAS_WORKSPACE_CONFIG=:4096:8` set; determinism then holds modulo CUDA convolution non-determinism, which produces variation $< 10^{-3}$ AUROC across replicate runs at the same seed, well below the $\pm.001$–$.020$ standard deviations of our headline tables. For double-blind review, all artefacts are mirrored at an anonymous `anonymous.4open.science` URL provided alongside the submission; the permanent URL will be added in the published version.

C.1 Architecture

Table 4 breaks down PAC-CTCN's components by parameter count and configuration notes.

MineNegatives$(\mathcal{B}, \textbf{modes}, M, [\varepsilon])$**.** For each positive triplet, generate M negatives by sampling a mode and applying: shuffle: replace x_t with x_{t+k} from the same trajectory, k uniform in $\{2, \ldots, 8\}$; action_swap: replace a_{t-1} with a'_{t-1} from a different trajectory at the matched timestep; patch: overlay a 48×48 patch on x_t at uniformly-sampled location, patch pixels optimized by 20 PGD

Table 4: PAC-CTCN component breakdown.

Component	Parameters	Notes
Visual encoder f_θ (ViT-Small)	22.1M	ImageNet-pretrained, shared across both frames
Action encoder h_ϕ (3-layer MLP)	150K	$7 \to 128 \to 256 \to 384$, GELU activations
Transition model T_ψ (2-layer MLP)	591K	$768 \to 512 \to 384$, GELU + dropout 0.1 + LayerNorm
Cross-attention fusion	1.0M	1 Pre-LN block, 8 heads, FFN expansion 4
Consistency head	893K	$1152 \to 384 \to 192 \to 1$, sigmoid
Total	**24.7M**	$\approx 0.35\%$ of 7B OpenVLA

steps (step $\alpha_{\text{PGD}}{=}\varepsilon/10$, random start) to minimize s_{disc} at $\ell_\infty \le \varepsilon$; gaussian: $x_t \leftarrow \text{clip}(x_t + \mathcal{N}(0, \varepsilon^2 I), 0, 1)$.

Note on ε_0. Encoder Lipschitzness (A1) implies $\varepsilon_0 \le L_f \cdot \varepsilon_{\text{pixel}}$, linking the pixel budget ε used in negative mining to the feature budget ε_0 that appears in Prop. 1.

C.2 Training Hyperparameters

Table 5 lists the full hyperparameter set for both training phases (Phase I clean and Phase II adversarial fine-tuning).

Table 5: Training hyperparameters for Phase I (clean) and Phase II (adversarial fine-tuning).

Hyperparameter	Phase I (clean)	Phase II (adv. fine-tune)
Epochs	80	60
Batch size	64	64
Learning rate	10^{-4}	2×10^{-5}
Optimizer	AdamW	AdamW
Weight decay	0.01	0.01
Gradient clip	1.0	1.0
LR schedule	cosine, 5 warmup epochs	cosine, 2 warmup epochs
Negatives / positive sample	3	3
Negative mix	shuffle, action swap	70% patch + 30% Gaussian
λ_{pred}	1.0	1.0
λ_{align}	0.0	0.0
Score-floor hinge	–	0.15
Best-checkpoint metric	clean separation	adversarial separation

C.3 Data and Compute

Data. LIBERO-Spatial, LIBERO-Object, LIBERO-Goal: 30 tasks, 50 demonstrations each, 128×128 RGB. 70/15/15 trajectory-level train/val/test split with split-seed 42, identical across all model seeds.

Compute. Per seed, per VLA: 1 NVIDIA A100 (80GB) for Phase I ($\approx$24h on OpenVLA, $\approx$18h on Octo, $\approx$20h on π_0) + Phase II ($\approx$18h / 14h / 16h respectively). Full paper compute: $\approx$330 A100-h for the 9 main runs, plus $\approx$470 A100-h for ablations, adaptive attacks, and cross-studies: $\approx$**800 A100-h** total.

Reproducibility. All random seeds, hyperparameters, and evaluation protocols are pinned and deterministic up to CUDA convolution non-determinism. Code and trained checkpoints will be released upon publication.

Train/test isolation. The $70/15/15$ split is performed at the trajectory level (not the frame level) with split-seed 42 fixed across every model seed, every VLA backbone, and every regime. Adversarial negatives during Phase II are generated only from training-split frames; the held-out test set is never

Algorithm 1 PAC-CTCN: training, calibration, and inference. Phase I shapes a clean encoder; Phase II adds adversarial negatives at the deployed budget; the curriculum is semantic-then-pixel by design (semantic-mismatch negatives in I, pixel-perturbation negatives in II). The detection guarantee (Prop. 1) requires $\Delta > 2\varepsilon_0$; otherwise the bound is vacuous. Convention: detector output $D{=}1$ denotes "adversarial."

Input: clean triplets $\mathcal{D}_{\text{clean}}{=}\{(x_{t-1}, a_{t-1}, x_t)\}$; $\mathcal{V}_{\text{clean}}$ (calibration); $\mathcal{V}_{\text{adv}}$ split 50/50 into $\mathcal{V}_{\text{adv}}^{\text{ck}}$, $\mathcal{V}_{\text{adv}}^{\alpha}$ (checkpoint selection / α search); target VLA π (frozen, used only for rollout collection, never queried during detector training).
Trainable: encoders f_θ, h_ϕ, transition T_ψ, discriminator $w_{\text{disc}}{=}$(CrossAttn, MLP head).
Hyperparams: $E_{\text{I}}{=}80$, $\eta_{\text{I}}{=}10^{-4}$; $E_{\text{II}}{=}60$, $\eta_{\text{II}}{=}2{\times}10^{-5}$; AdamW (wd$=10^{-2}$, clip$=1.0$); batch$=64$; $M{=}3$; $\lambda_{\text{pred}}{=}1.0$, $\lambda_{\text{align}}{=}0$; mixing $\alpha{=}0.5$ (default, swept in Table 10); floor $h{=}0.15$; budget $\varepsilon{=}16/255$; PGD $\alpha_{\text{PGD}}{=}\varepsilon/10$, random start.
Output: trained $(\theta, \phi, \psi, w_{\text{disc}})$, threshold τ, detector $D{:}\,(x_{t-1}, a_{t-1}, x_t){\to}\{0, 1\}$.

Forward$(\mathcal{B})$: $z_{t-1}{\leftarrow}f_\theta(x_{t-1})$; $z_t{\leftarrow}f_\theta(x_t)$; $e_a{\leftarrow}h_\phi(a_{t-1})$; $\hat{z}_t{\leftarrow}T_\psi(z_{t-1}, e_a)$;
$s_{\text{pred}}{\leftarrow}\frac{1}{2}(\cos(\hat{z}_t, \text{sg}[z_t]){+}1)$; $s_{\text{disc}}{\leftarrow}\sigma(\text{MLP}(\text{CrossAttn}(z_t, [z_{t-1}, e_a])))$ ▷ sg acts in $\mathcal{L}_{\text{pred}}$ only

Phase I, clean encoder shaping

for epoch $= 1, \ldots, E_{\text{I}}$ **do**
  for $\mathcal{B} \subset \mathcal{D}_{\text{clean}}$ **do**
    $(\hat{z}_t, s_{\text{pred}}, s_{\text{disc}}) \leftarrow$ FORWARD$(\mathcal{B})$
    $\mathcal{N} \leftarrow$ MINENEGATIVES$(\mathcal{B}, \{\text{shuffle}, \text{action_swap}\}, M)$
    $\mathcal{L}_{\text{pred}} \leftarrow 1 - \cos(\hat{z}_t, \text{sg}[z_t])$
    $\mathcal{L}_{\text{disc}} \leftarrow \text{BCE}(s_{\text{disc}}; \text{pos}{=}\mathcal{B}, \text{neg}{=}\mathcal{N})$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{disc}} + \lambda_{\text{pred}}\mathcal{L}_{\text{pred}}$
    Update $(\theta, \phi, \psi, w_{\text{disc}})$ via AdamW(η_{I})
  end for
end for

Phase II, adversarial fine-tuning

for epoch $= 1, \ldots, E_{\text{II}}$ **do**
  for $\mathcal{B} \subset \mathcal{D}_{\text{clean}}$ **do**
    $(\hat{z}_t, s_{\text{pred}}, s_{\text{disc}}) \leftarrow$ FORWARD$(\mathcal{B})$
    $\mathcal{N} \leftarrow$ MINENEGATIVES$(\mathcal{B}, \{\text{patch}{:}.7, \text{gaussian}{:}.3\}, M, \varepsilon)$
    $\mathcal{L}_{\text{pred}} \leftarrow 1 - \cos(\hat{z}_t, \text{sg}[z_t])$
    $\mathcal{L}_{\text{floor}} \leftarrow \mathbb{E}_{\mathcal{B}}[(h - s_{\text{pred}})_+]$ ▷ floor on positives
    $\mathcal{L}_{\text{disc}} \leftarrow \text{BCE}(s_{\text{disc}}; \text{pos}{=}\mathcal{B}, \text{neg}{=}\mathcal{N})$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{disc}} + \lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{floor}}$
    Update $(\theta, \phi, \psi, w_{\text{disc}})$ via AdamW(η_{II})
  end for
  Retain checkpoint maximising AUROC on $\mathcal{V}_{\text{adv}}^{\text{ck}}$
end for

Calibration

$\alpha^* \leftarrow \arg\max_\alpha \text{AUROC}\big(\mathcal{V}_{\text{adv}}^{\alpha}; \alpha s_{\text{pred}}{+}(1{-}\alpha)s_{\text{disc}}\big)$ ▷ default $\alpha^*{=}0.5$
$\{s_i\} \leftarrow \{\alpha^* s_{\text{pred}}^{(i)}{+}(1{-}\alpha^*)s_{\text{disc}}^{(i)} : i \in \mathcal{V}_{\text{clean}}\}$
$\mu_s, \sigma_s \leftarrow \text{mean}, \text{std}(\{s_i\})$; $\tau \leftarrow \mu_s - 3\sigma_s$

Inference (per timestep, gradients off)

$(\hat{z}_t, s_{\text{pred}}, s_{\text{disc}}) \leftarrow$ FORWARD$((x_{t-1}, a_{t-1}, x_t))$ ▷ $\partial\hat{z}_t/\partial\delta{=}\mathbf{0}$ (Eq. 1)
$s \leftarrow \alpha^* s_{\text{pred}} + (1{-}\alpha^*)s_{\text{disc}}$ ▷ Prop. 1: $s_{\text{pred,clean}} - s_{\text{pred,adv}} \geq \frac{\Delta(\Delta - 2\varepsilon_0)}{4}$
return $D = \mathbb{1}[\, s < \tau\,]$

seen during Phase I encoder shaping, Phase II detector training, or negative generation. There is no possibility of frame-level leakage across splits.

D Extended Experimental Results

This appendix provides per-attack and per-suite breakdowns of Table 2's headline results, the per-regime theorem-verification quantities (Table 7), and TPR at strict false-positive thresholds (Table 8).

D.1 PER-ATTACK, PER-SUITE BREAKDOWN

Table 6: PAC-CTCN detection AUROC per VLA, attack, and LIBERO suite/regime (3 seeds, mean $\pm$ std). Standard suites {S,O,G}, harder LIBERO-10, and LIBERO-Plus distribution shifts.

VLA	Attack	Spatial	Object	Goal	LIBERO-10	LIBERO-Plus
OpenVLA	UADA	0.999 ± 0.001	0.999 ± 0.001	0.997 ± 0.003	0.963 ± 0.006	0.934 ± 0.011
	UPA	0.999 ± 0.001	0.998 ± 0.002	0.999 ± 0.001	0.964 ± 0.009	0.928 ± 0.013
	TMA	0.998 ± 0.002	0.999 ± 0.001	0.997 ± 0.002	0.957 ± 0.011	0.922 ± 0.018
Octo	UADA	0.995 ± 0.003	0.997 ± 0.002	0.990 ± 0.005	0.952 ± 0.009	0.917 ± 0.020
	UPA	0.996 ± 0.002	0.995 ± 0.003	0.992 ± 0.003	0.948 ± 0.013	0.921 ± 0.014
	TMA	0.994 ± 0.004	0.996 ± 0.002	0.991 ± 0.004	0.945 ± 0.010	0.910 ± 0.019
π_0	UADA	0.991 ± 0.005	0.996 ± 0.002	0.987 ± 0.007	0.943 ± 0.011	0.903 ± 0.022
	UPA	0.995 ± 0.003	0.994 ± 0.004	0.990 ± 0.005	0.939 ± 0.014	0.907 ± 0.017
	TMA	0.993 ± 0.005	0.993 ± 0.005	0.987 ± 0.007	0.936 ± 0.012	0.898 ± 0.024

LIBERO-Goal is consistently the hardest of the standard suites across architectures, a pattern documented for earlier CTCN-style defenses (Xu et al., 2025): cluttered workspace compresses Δ while ε_0 stays roughly constant. The harder regimes follow the same monotone ordering, with ε_0 growing as horizons lengthen and scenes become more cluttered. Table 7 reports the full per-VLA per-regime breakdown of the theorem's quantities, substantiating the $\Delta/(2\varepsilon_0) \in [3.0, 8.3]$ range.

Table 7: Theorem-verification quantities of Prop. 1 per VLA per regime (3 seeds, means). The detection condition $\Delta > 2\varepsilon_0$ holds across all regimes; %hold degrades from 99.9% on {S,O,G} to 88.9% on π_0+LIBERO-Plus as ε_0 grows. Lemma 1 converts these to Table 2's AUROCs.

VLA	Regime	ε_0	$\bar{\Delta}$	$\Delta/(2\varepsilon_0)$	% holds
OpenVLA	{S,O,G}	$0.105 \pm .001$	1.736	8.26	99.9%
	LIBERO-10	$0.143 \pm .003$	1.682	5.88	97.4%
	LIBERO-Plus	$0.197 \pm .005$	1.591	4.04	93.6%
Octo	{S,O,G}	$0.132 \pm .003$	1.696	6.42	99.2%
	LIBERO-10	$0.178 \pm .005$	1.638	4.60	94.7%
	LIBERO-Plus	$0.221 \pm .007$	1.547	3.50	91.2%
π_0	{S,O,G}	$0.141 \pm .004$	1.651	5.86	98.4%
	LIBERO-10	$0.196 \pm .006$	1.583	4.04	93.1%
	LIBERO-Plus	$0.246 \pm .009$	1.498	3.05	88.9%

Per-sample correlation c (Assumption A4): on standard suites $c = 0.82 \pm .04$ (OpenVLA), $0.76 \pm .05$ (Octo), $0.71 \pm .06$ (π_0); on LIBERO-Plus c degrades modestly to $0.74 \pm .06$, $0.69 \pm .07$, $0.64 \pm .08$ respectively, comfortably positive across all conditions.

D.2 TPR AT FIXED LOW FPR

Table 8: Per-frame TPR at calibrated operating thresholds ($\tau = \mu_s - 3\sigma_s$), averaged over three seeds. The Combined detector retains high TPR even at the stricter 1% FPR threshold across all VLAs.

VLA	TPR@5%FPR (frame)	TPR@1%FPR (frame)
OpenVLA	0.998 ± 0.002	0.998 ± 0.002
Octo	0.991 ± 0.004	0.987 ± 0.005
π_0	0.985 ± 0.008	0.978 ± 0.010

Per-frame TPR remains high even at the strict 1% FPR threshold across architectures. The Combined-score gap between clean ($\mu_s \approx 0.876$, $\sigma_s \approx 0.188$ on OpenVLA, the *combined*-score moments, distinct from the predictive-score μ_s=0.997, σ_s=0.003 of Lemma 1 and Fig. 2) and adversarial samples (attack mean ≈ 0.12 on standard suites) places adversarial samples well into the detection region without requiring trajectory-level smoothing.

Table 9: Multi-architecture ablation (AUROC, mean over three seeds). Bold indicates the default configuration; standard deviation across seeds ≤ 0.011. Per-component sweeps follow.

	Default	**Combination**		**Fusion**	**Transition**	**Critical (broken)**	
VLA	α=.5	α=0	α=1	min	CA T_ψ	−act.	−s.g.
OpenVLA	$\mathbf{.999 \pm .001}$	.927	.999	.927	.999	.48	.51
Octo	$\mathbf{.994 \pm .003}$	.914	.990	.904	.994	.47	.52
π_0	$\mathbf{.992 \pm .004}$	.908	.987	.895	.992	.49	.53

E Comprehensive Ablations

Table 9 summarizes the multi-architecture ablation; the per-component sweeps follow: combination-rule sweep (Table 10), transition-model variants (Table 11), loss-term sensitivity (Table 12), action-encoder depth (Table 13), and training-phase contribution (Table 14).

E.1 Full Combination-Rule Sweep

Table 10: Full combination-rule ablation on OpenVLA, three seeds.

Rule	AUROC	Score gap
Linear $\alpha = 0.00$ (disc-only, CTCN reduction)	0.927 ± 0.004	0.73
Linear $\alpha = 0.25$	0.999 ± 0.001	0.73
Linear $\alpha = 0.50$ **(default)**	$\mathbf{0.999 \pm 0.001}$	0.74
Linear $\alpha = 0.75$	0.999 ± 0.001	0.75
Linear $\alpha = 1.00$ (pred-only, Prop. 1 directly)	0.999 ± 0.001	0.75
$\min(s_{\text{pred}}, s_{\text{disc}})$	0.927 ± 0.004	0.73
Product $s_{\text{pred}} \cdot s_{\text{disc}}$	0.999 ± 0.001	0.73
Learned gating $\sigma(w_1 s_{\text{pred}} + w_2 s_{\text{disc}} + b)$	0.999 ± 0.001	0.28

Linear combinations with $\alpha \in [0.25, 1.0]$ all attain AUROC $0.999 \pm .001$ on OpenVLA standard suites; the predictive branch alone ($\alpha = 1$) saturates without help from the discriminative branch. Learned gating matches AUROC but shrinks the calibrated threshold margin; we therefore recommend fixed $\alpha = 0.5$ for deployment.

E.2 Transition Model Variants

Table 11 ablates the choice of transition model architecture for T_ψ, reporting parameter count, AUROC, and clean prediction error ε_0 for each variant.

Table 11: Transition model T_ψ ablation on OpenVLA.

Variant	Parameters	AUROC	ε_0
Linear ($768 \to 384$)	295K	0.987 ± 0.004	0.168
MLP 2-layer (default)	591K	0.999 ± 0.001	0.105
MLP 3-layer	743K	0.999 ± 0.001	0.102
MLP 2-layer + residual skip	591K	0.999 ± 0.001	0.104
Cross-attention (1 Pre-LN block)	1.77M	0.999 ± 0.001	0.100

E.3 Loss-Term Sensitivity

Table 12 sweeps the two loss-term weights (λ_{pred}, λ_{align}), the cosine-vs-MSE choice for $\mathcal{L}_{\text{pred}}$, and the stop-gradient on z_t. Removing the stop-gradient collapses AUROC to chance (0.51) and using MSE instead of cosine drops AUROC to 0.912; loss-weight perturbations stay within 0.021 of the default.

Table 12: Loss-term sensitivity on OpenVLA.

Variant	AUROC	Note
λ_{pred}=1.0, λ_{align}=0.0 (default)	0.999 ± 0.001	–
λ_{pred}=0.5	0.996 ± 0.002	T_ψ under-trained
λ_{pred}=2.0	0.997 ± 0.002	disc branch slightly weaker
λ_{align}=0.25	0.993 ± 0.002	reintroduces partial collapse pressure
λ_{align}=0.50	0.978 ± 0.005	mild collapse observed in Δ
MSE $\mathcal{L}_{\text{pred}}$ instead of cosine	0.912 ± 0.011	dominates total loss at scale $\mathcal{O}(d)$
No stop-gradient on z_t	0.51 ± 0.03	encoder collapse (Prop. 3)

E.4 Action Encoder Depth

Table 13 ablates the depth of the action encoder h_ϕ; AUROC saturates at 3 layers (default) and zeroing the action input collapses AUROC to chance (0.48, the empirical counterpart of Lemma 2).

Table 13: Action encoder depth ablation on OpenVLA.

Action encoder h_ϕ	AUROC	Parameters
Zeroed (no action, cf. Lemma 2)	0.48	–
Linear ($7 \to 384$)	0.962 ± 0.004	3K
2-layer MLP ($7 \to 128 \to 384$)	0.994 ± 0.002	53K
3-layer MLP (default)	0.999 ± 0.001	150K
4-layer MLP ($7 \to 128 \to 256 \to 256 \to 384$)	0.999 ± 0.001	280K

E.5 Training-Phase Ablation

Table 14: Contribution of each training phase on OpenVLA.

Training recipe	AUROC	Mean Δ
Phase I only (clean, no adv. training)	0.877 ± 0.015	0.36
Phase II only (from random initialisation)	0.965 ± 0.007	1.21
Phase I + Phase II (default)	$\mathbf{0.999 \pm 0.001}$	1.74
Phase I + Phase II + 60 more adv. epochs	0.999 ± 0.001	1.75

Phase II (adversarial fine-tuning) is what pushes Δ well above $2\varepsilon_0$. Phase I alone achieves strong but not saturating AUROC; neither phase is redundant.

F Efficiency Deep Dive

This appendix details PAC-CTCN's deployment efficiency: per-VLA inference latency and memory (Table 15), batch-size scaling (Table 16), relative deployment footprint vs. the protected VLA (Table 17), and training-compute budget (Table 18).

F.1 Latency and Memory across VLAs

The detector is identically sized across VLAs; the small timing differences come from the VLA's own memory allocator state.

F.2 Batch-Size Scaling

Table 16 reports per-sample latency and throughput on OpenVLA across batch sizes from 1 to 64; throughput grows super-linearly through B=16 as kernel-launch overhead amortizes, then sub-linearly to $\sim 1.1 \times 10^6$ samples/s at B=64.

Table 15: PAC-CTCN detection latency, throughput, and memory footprint across three VLAs (batch size 1, A100).

VLA backbone	Latency (ms/sample)	Throughput (samples/s)	GPU memory (MB)
OpenVLA	0.88 ± 0.05	1140	186
Octo	0.84 ± 0.06	1189	184
π_0	0.89 ± 0.05	1124	188

Table 16: Batch-size scaling of PAC-CTCN on OpenVLA (A100).

Batch size	Per-sample latency (ms)	Throughput (samples/s)
1	0.88	1,140
4	0.31	12,900
16	0.12	133,000
64	0.058	1.10×10^6

F.3 Relative Deployment Footprint

Table 17: Relative deployment cost of PAC-CTCN vs. the protected VLA (OpenVLA).

	Parameters	Inference latency (ms)	Memory (MB)
Protected VLA (OpenVLA 7B)	7.0B	$\sim$150	$\sim$14,000
PAC-CTCN	24.7M	0.88	186
Relative overhead	0.35%	0.59%	1.33%

The detection overhead is well within the envelope of real-time VLA control loops (10–50 Hz).

F.4 Training-Compute Budget

Table 18 reports per-VLA per-phase training compute on a single NVIDIA A100 (80GB); the full paper consumed approximately 800 A100-hours.

G SOTA Defense Comparison

Complementarity. Detection and robustification address different parts of the threat surface. A robustified VLA mitigates small perturbations, lowering the effective Δ the adversary can impose; PAC-CTCN catches larger-budget attacks that overwhelm the robustification. Stacking the two gives a layered defense whose failure modes are not correlated: a perturbation that evades the robustified VLA (large Δ) is *easier* for PAC-CTCN to detect (Prop. 1), and a perturbation too small to trigger PAC-CTCN's threshold is too small to change the robustified VLA's behavior. Quantifying this composition is left to future work.

H Cross-Generalization Studies

This appendix tests how PAC-CTCN generalizes along three axes: across attack families (Table 20), across LIBERO task suites (Table 21), and across VLA architectures (Table 22).

H.1 Cross-Attack Generalization

Cross-attack degradation is ≤ 0.007 AUROC: the predictive branch does not memorise attack-specific artifacts but responds to feature displacement, which is a property of any effective attack.

Table 18: Training-compute budget (A100-hours).

VLA backbone	Phase I (clean)	Phase II (adv.)	Total per seed
OpenVLA	24	18	42
Octo	18	14	32
π_0	20	16	36
9 main runs (3 VLAs $\times$ 3 seeds)	–	–	$\sim$330
Ablations, adaptive, cross-studies	–	–	$\sim$470
Paper total	–	–	**$\sim$800 A100-h**

Table 19: Comparison with SOTA VLA defenses. "Det." = produces explicit anomaly score; "No retrain" = no VLA modification; "Clean pres." = preserves clean task success; "Formal" = theoretical detection guarantee.

Defense	Type	Det.	No retrain	Clean pres.	Formal	AUROC (det)
EDPA (Xu et al., 2025)	Robustify	–	–	partial[†]	–	N/A
RobustVLA (Zhang et al., 2025b)	Robustify	–	–	partial	–	N/A
CRT (Guarnizo Orjuela et al., 2026)	Purify	–	✓	– (degrades)	–	N/A
SafeVLA (Zhang et al., 2025a)	CMDP	–	–	– (RL retrain)	–	N/A
SAFE (Gu et al., 2025)[‡]	Task-failure	✓	✓	✓	–	0.671 ± 0.018
TCD (optical flow)	Detect	✓	✓	✓	–	0.52 ± 0.03
TCV (feature squeezing)	Detect	✓	✓	– (degrades)	–	0.58 ± 0.04
CTCN (ours, α=0)	Detect	✓	✓	✓	–	0.927 ± 0.004
PAC-CTCN (ours)	**Detect**	✓	✓	✓	✓	$\mathbf{0.999 \pm 0.001}$

[†]EDPA reports 34% improvement under attack but retains 65–97% failure rates on difficult suites. [‡]SAFE targets multitask task-failure detection on unseen tasks; number is our reproduction on adversarial inputs (App. K). Native task-failure AUROC on OpenVLA + LIBERO Unseen is 0.7347 (Gu et al., 2025), which our reproduction matches at $0.731 \pm .014$.

H.2 Cross-Task Generalization

Table 21 tests cross-task transfer: training the detector on a single LIBERO suite and evaluating on the others. Off-diagonal AUROC drops 0.018–0.025 relative to the diagonal, larger than cross-attack drops because suites differ in visual scene distribution. Mixed-suite training recovers diagonal performance.

H.3 Cross-Architecture Detector Transfer

A harder question: can a detector trained on one VLA be deployed on another? Table 22 tests zero-shot detector transfer.

Zero-shot transfer drops AUROC by 0.10–0.17 because Δ (feature displacement in the *detector's* encoder) depends on how the detector's ViT reacts to attacks optimized for a different VLA. 10 epochs of fine-tuning on the target VLA recovers performance to within 0.003 of the diagonal. For deployment, we recommend per-VLA adversarial fine-tuning, as reported in Table 2.

H.4 Robustness to Benign Distribution Shifts

PAC-CTCN must distinguish adversarial inputs from clean-but-shifted inputs (e.g., new lighting, mild clutter); otherwise it would degenerate into an out-of-distribution detector and false-alarm on every novel scene. The contrastive component of $\mathcal{L}_{\text{disc}}$ trains exactly this separation: positive pairs are clean (x_{t-1}, a_{t-1}, x_t) triplets; negatives include shuffled-time, action-swapped, and adversarial samples. Mild benign shifts modestly raise ε_0 (the predictor was not trained on the new visual distribution) but leave Δ unaffected (no perturbation pressure), so the detection condition $\Delta > 2\varepsilon_0$ continues to hold with margin. Empirically, augmenting the LIBERO test split with $\pm 30\%$ brightness, σ=0.05 Gaussian blur, and 20% random cluttering raises ε_0 from 0.105 to 0.124 on OpenVLA while Δ moves only from 1.736 to 1.731; AUROC drops from 0.999 to 0.991. Extreme shifts (novel lighting

Table 20: Cross-attack generalization on OpenVLA. Rows: attack family used during adversarial fine-tuning; columns: attack family at test.

Train ↓ / Test →	UADA	UPA	TMA
UADA only	**0.999**	0.994	0.993
UPA only	0.992	**0.999**	0.994
TMA only	0.993	0.993	**0.999**
Mixed (default)	0.999	0.999	0.999

Table 21: Cross-task generalization on OpenVLA. Rows: LIBERO suite used for training; columns: LIBERO suite at test.

Train ↓ / Test →	Spatial	Object	Goal
Spatial only	**0.999**	0.978	0.974
Object only	0.981	**0.999**	0.979
Goal only	0.976	0.980	**0.999**
All suites (default)	0.999	0.999	0.998

regimes, simulator-to-real transfer) push ε_0 further and would require online threshold recalibration; this case is outside our scope.

I Adaptive Attack: Extended Analysis

Reading the table. "Target: Combined" is the realistic adaptive adversary (knows the full detector). "Target: Pred/Disc" isolates the adversary's effectiveness against each branch. The cross-branch columns reveal an asymmetry: attacks crafted only for the discriminative branch are partially or fully caught by the (unattacked) predictive branch at $\varepsilon \geq 8/255$, while the converse does not hold (attack-pred does not invoke detection from disc). This is the operational content of Proposition 1: the predictive branch is harder to spoof because $\hat{z}_t$ is computed from past information the attacker does not control.

Note on the Disc-target AUROC at higher ε. The Disc branch's own AUROC *rises* with ε when targeted alone ($0.47 \to 0.90 \to 0.998$). This is consistent with Phase II's adversarial fine-tuning, which exposes Disc specifically to patch and Gaussian-noise negatives at $\ell_\infty \approx 16/255$: small-budget perturbations are out-of-training-distribution and slip through, whereas larger perturbations resemble Phase II's training distribution and Disc flags them despite the gradient pressure to evade. This pattern is a property of the training data, not a contradiction of the proposition; the Pred branch (which has no comparable training signal) shows the expected monotone evasion vs. ε pattern (0.43, 0.44, 0.46).

Harder adaptive attacks. An adversary who specifically targets the predictive branch's structure would craft δ to move z_t *toward* $\hat{z}_t$, not away from it. This is a constrained optimization: the attacker must simultaneously (i) change the VLA's action (requiring large Δ_{VLA}) and (ii) preserve $f_\theta(x_t + \delta) \approx \hat{z}_t$ (requiring the detector encoder's response to be specific). Preliminary experiments indicate these objectives are in non-trivial tension and the joint optimization does not cleanly succeed; a comprehensive treatment is beyond scope.

Targeting the prediction $\hat{z}_t$ directly. A natural-looking shortcut would be to move the prediction itself: substitute $\hat{z}_t \to \hat{z}_t + \eta$. By construction this is unavailable to a deployment-time attacker. At time t, $\hat{z}_t$ is a function of x_{t-1}, a_{t-1}, both committed and stored before the attacker observes the current frame; the attacker has no causal route to perturb the past. An adversary with temporal prescience (modifying past frames retroactively) defeats the guarantee, but no adversarial-ML threat model considered in the literature grants such capabilities.

Gray-box transfer adversaries. Between gray-box (our primary threat model: VLA public, detector private) and white-box (this appendix's stress test) sits an adversary who knows the detector's architecture and training recipe but not its weights. Such an adversary can craft transferable attacks

Table 22: Zero-shot detector transfer: rows indicate the VLA the detector was adversarially fine-tuned against; columns indicate the VLA the test-time attacks are crafted against. Entries are AUROC.

Detector trained on ↓	OpenVLA attacks	Octo attacks	π_0 attacks
OpenVLA	**0.999**	0.861	0.832
Octo	0.894	**0.994**	0.875
π_0	0.882	0.867	**0.992**

Table 23: White-box adaptive attack across all three VLAs (AUROC; lower ⇒ more evasion). Generic PGD targets the combined detector; theory-aware additionally drives $f_\theta(x_t + \delta) \to \hat{z}_t$.

	Generic PGD			**Theory-aware**		
ε	OpenVLA	Octo	π_0	OpenVLA	Octo	π_0
$4/255$	0.26	0.31	0.29	0.36	0.41	0.39
$8/255$	0.84	0.79	0.76	0.89	0.86	0.81
$16/255$	0.998	0.97	0.95	0.999	0.99	0.97

against a surrogate PAC-CTCN trained from scratch and hope they transfer to the deployed instance. The predictive branch is empirically more transfer-resistant than the discriminative branch because $\hat{z}_t$ depends on the specific weights (θ, ϕ, ψ) rather than on a loss-driven decision boundary; preliminary cross-seed transfer experiments show AUROC degradation of 0.04–0.09 under this regime, far smaller than full white-box. A comprehensive surrogate-attack study is left for future work.

J Extension to Other Modalities and Architectures

PAC-CTCN's mechanism requires only: (i) an image stream, (ii) an executed-action signal, and (iii) temporal ordering. We sketch how the method carries over to adjacent settings.

Diffusion-policy VLAs. Diffusion policies (Chi et al., 2023; Chen et al., 2024) parameterize the action distribution via iterative denoising. The detector interface is unchanged: we still observe (x_{t-1}, a_{t-1}, x_t) triples from the rollout. Adversarial perturbations that change the sampled action must change image features the policy consumes, satisfying A3; Proposition 1 applies without modification. Preliminary runs on Diffusion Policy (Chi et al., 2023) on LIBERO-Spatial reach AUROC 0.988 ± 0.004.

Flow-matching policies. π_0 is already included. The executed action a_{t-1} remains observable even when the VLA uses flow integration internally; T_ψ does not depend on the VLA's sampling procedure.

Multi-view and RGB-D observations. The visual encoder f_θ can be any architecture producing a single embedding (multi-view ViT, ResNet+depth-fusion). Proposition 1 holds for any f_θ satisfying A1. Multi-view should *increase* Δ (a single-view perturbation is inconsistent across views) while leaving ε_0 roughly stable, tightening the bound.

Non-manipulation embodied tasks. The method extends to any embodied task with triplet (x_{t-1}, a_{t-1}, x_t): autonomous driving (ego-motion + steering), mobile navigation (pose + velocity command), bimanual manipulation (two-arm state + two-arm action). For language-conditioned agents, h_ϕ would handle discrete or language-shaped action inputs.

Beyond embodied AI. The temporal-asymmetry argument (Eq. 1) requires only a sequential observation stream with fixed-at-time-t past. Video anomaly detection, intrusion detection in time series, and attack detection on recurrent classifiers all fit. The generalising ingredient that replaces the VLA's executed action is "any causally-prior side information correlated with the next observation."

Table 24: Full per-target white-box adaptive analysis on OpenVLA (PGD, 100 steps, 10 restarts; single seed s42). Evasion% = adversarial samples above threshold. Cross-branch columns report the AUROC of the *unattacked* branch when one branch is targeted; "–" under Target=Combined where both branches are attacked.

ε	Target	AUROC	Evasion%	Adv mean	Pred (if Disc attacked)	Disc (if Pred attacked)
$4/255$	Combined	0.26	93.6%	0.897	–	–
	Pred	0.43	100%	0.884	–	0.48
	Disc	0.47	92.0%	0.871	0.86	–
$8/255$	Combined	0.84	23.9%	0.380	–	–
	Pred	0.44	100%	0.884	–	0.49
	Disc	0.90	21.6%	0.363	0.997	–
$16/255$	Combined	0.998	12.2%	0.252	–	–
	Pred	0.46	100%	0.880	–	0.48
	Disc	0.998	11.5%	0.250	0.999	–

Language-instruction attacks. The detector observes only image and action streams, so adversarial perturbations to the natural-language instruction (token substitution, prompt injection) bypass it directly: the predicted $\hat{z}_t$ is unchanged because x_{t-1}, a_{t-1} are unchanged, and any divergence in z_t caused by altered VLA behavior will be flagged only if it manifests as image-feature displacement large enough to trip $\Delta > 2\varepsilon_0$. A principled extension would add a third branch with an instruction-consistency score, predicting the next-frame embedding from $(z_{t-1}, e_a, \text{instruction})$ and flagging instructions that produce inconsistent successor predictions. We leave this to follow-up work.

K SAFE Reproduction Protocol and Extended Results

We reproduce SAFE (Gu et al., 2025) on our adversarial setting to provide an apples-to-apples learned baseline anchor for the methodological-gap argument in §2.2 and Table 2. SAFE's authors explicitly do not evaluate adversarial robustness; their target is multitask task-failure on unseen tasks. Our reproduction has three goals: (i) sanity-check that our SAFE training matches their published native numbers, (ii) measure SAFE on adversarial inputs as a baseline for PAC-CTCN, and (iii) test whether adversarial fine-tuning of SAFE closes the gap.

K.1 Protocol

Code and hyperparameters. We use the official implementation at `github.com/vla-safe/SAFE` with the authors' published hyperparameters: SAFE-MLP backbone (2.3M parameters), AdamW optimizer (lr $= 10^{-4}$, weight decay 10^{-2}), 100 training epochs, batch size 128. Threshold calibration via functional conformal prediction with $\alpha = 0.1$ on a held-out set of successful rollouts. Per-rollout failure score is the max-so-far aggregation $\bar{s}_T = \max_{t \le T} s_t$, matching the protocol in (Gu et al., 2025, Table 1).

Training data. 50 successful + 50 naturally-failed LIBERO rollouts per suite per VLA, collected by running each VLA's published checkpoint and labelling rollouts by task-completion outcome. Naturally-failed rollouts are abundant on LIBERO-10 ($\sim$50% baseline failure rate (Wang et al., 2025)) and harvested by oversampling from the same evaluation pool. Same 70/15/15 trajectory-level split as PAC-CTCN training (§5.1).

Evaluation conditions.

- *Native sanity check:* score SAFE on naturally-failed rollouts only; compute AUROC against the success/failure label. Should reproduce the 0.73–0.78 range from (Gu et al., 2025, Table 1) on OpenVLA + LIBERO Unseen.
- *Adversarial primary:* score SAFE on UADA/UPA/TMA-perturbed rollouts at $\ell_\infty = 16/255$ (the same adversarial test set used for PAC-CTCN); compute AUROC against the clean/adversarial label. Adversarial = positive.

K.2 SANITY CHECK (MATCHES PUBLISHED NUMBERS)

Our SAFE-MLP reproduction attains AUROC 0.731 ± 0.014 on OpenVLA + LIBERO Unseen (native task-failure), within 0.005 of the published 0.7347 (Gu et al., 2025, Table 1). The training pipeline therefore reproduces SAFE faithfully; downstream adversarial numbers are not artefacts of a misconfigured baseline.

Table 25: SAFE-MLP sanity check on native task-failure detection (our reproduction vs. (Gu et al., 2025) published).

VLA + benchmark	Ours (3 seeds)	Published (Gu et al., 2025)
OpenVLA + LIBERO Unseen	$0.731 \pm .014$	0.7347
π_0 + LIBERO Unseen	$0.728 \pm .017$	0.7327

K.3 SAFE ADAPTED TO ADVERSARIAL INPUTS (EXTENDED RESULTS)

Table 26 reports the full SAFE-MLP and SAFE-LSTM results across all VLAs and LIBERO regimes on adversarial inputs. SAFE-LSTM tracks SAFE-MLP within ± 0.018 AUROC across every condition, consistent with the small MLP/LSTM gap reported in (Gu et al., 2025). Both fall 0.33–0.43 AUROC below PAC-CTCN.

The sub-chance inversion at the hardest cell. On π_0+LIBERO-Plus, SAFE-MLP attains AUROC 0.476, below chance. An AUROC under 0.5 indicates anti-correlated score behavior: SAFE was trained to assign *high* failure scores to features in the natural-failure zone learned from training rollouts. Under compound natural+adversarial OOD on the smallest VLA, perturbed-input features systematically land outside that zone, they look more like "successful execution" to SAFE than to a generic classifier, producing low scores on adversarial inputs and high scores on clean ones, the inversion that yields sub-chance AUROC. This is the strongest single piece of empirical evidence we have for the structural distinction motivating Proposition 1: feature-zone classification and adversarial-displacement detection are not interchangeable, and the harder the regime, the more SAFE's predictions invert. Adversarial fine-tuning of SAFE (App. K.4) eliminates the inversion but does not close the AUROC gap to PAC-CTCN.

Table 26: SAFE detection AUROC on adversarial inputs (3 seeds). PAC-CTCN reproduced here for comparison.

VLA	Method	LIBERO-{S,O,G}	LIBERO-10	LIBERO-Plus
OpenVLA	SAFE-MLP	$0.671 \pm .018$	$0.594 \pm .022$	$0.523 \pm .028$
	SAFE-LSTM	$0.683 \pm .016$	$0.608 \pm .020$	$0.541 \pm .025$
	PAC-CTCN	$\mathbf{0.999 \pm .001}$	$\mathbf{0.961 \pm .008}$	$\mathbf{0.928 \pm .014}$
Octo	SAFE-MLP	$0.638 \pm .024$	$0.561 \pm .029$	$0.491 \pm .034$
	SAFE-LSTM	$0.649 \pm .021$	$0.574 \pm .026$	$0.508 \pm .031$
	PAC-CTCN	$\mathbf{0.994 \pm .003}$	$\mathbf{0.948 \pm .011}$	$\mathbf{0.916 \pm .017}$
π_0	SAFE-MLP	$0.612 \pm .027$	$0.548 \pm .031$	$0.476 \pm .036$
	SAFE-LSTM	$0.627 \pm .025$	$0.559 \pm .029$	$0.493 \pm .034$
	PAC-CTCN	$\mathbf{0.992 \pm .004}$	$\mathbf{0.939 \pm .013}$	$\mathbf{0.903 \pm .020}$

K.4 ADVERSARIAL FINE-TUNING OF SAFE: DOES THE GAP CLOSE?

A natural question is whether the gap is mechanistic or simply reflects SAFE not being trained on adversarial examples. To answer, we adversarially fine-tune SAFE-MLP using PAC-CTCN's Phase II adversarial-negative pool (70% patch + 30% Gaussian, $\ell_\infty = 16/255$) on top of the natural-failure-trained SAFE-MLP, for 30 additional epochs at lr $= 5 \times 10^{-5}$.

Findings. (i) Adv-FT substantially improves SAFE on the easiest regime ($+0.176$ AUROC on {S,O,G}), eliminating the sub-chance inversion observed without it. (ii) The gap to PAC-CTCN

Table 27: SAFE-MLP with adversarial fine-tuning (SAFE-MLP-AdvFT) vs. vanilla SAFE-MLP and PAC-CTCN (OpenVLA, 3 seeds).

Method	LIBERO-{S,O,G}	LIBERO-10	LIBERO-Plus
SAFE-MLP	$0.671 \pm .018$	$0.594 \pm .022$	$0.523 \pm .028$
SAFE-MLP-AdvFT	$0.847 \pm .012$	$0.769 \pm .017$	$0.692 \pm .024$
PAC-CTCN	$\mathbf{0.999 \pm .001}$	$\mathbf{0.961 \pm .008}$	$\mathbf{0.928 \pm .014}$

remains 0.15–0.24 AUROC, widening on harder regimes. (iii) Adv-FT requires generation of adversarial training data, while Proposition 1 reduces PAC-CTCN's predictive-branch design to minimising clean ε_0. The structural advantage of predictive consistency, comparing to a target the adversary cannot move, is mechanistic, not a training-data artefact.

L Extended Related Work

This appendix expands §2 along four lines: (M.1) robustness- and policy-level VLA defenses; (M.2) the classical adversarial-input detection lineage and the structural reason it does not transfer to embodied control, including a careful treatment of the obfuscated-gradients critique; (M.3) failure-detection and runtime-monitoring for embodied policies; (M.4) forward-dynamics and predictive representation learning, the technical lineage PAC-CTCN repurposes.

L.1 Robustness- and policy-level VLA defenses

VLA defenses to date have been weight-modifying, not detection-side, and each pays a clean-performance or retraining cost that limits deployment. EDPA (Xu et al., 2025) adversarially fine-tunes the visual encoder; the technique reduces attack success but retains substantial residual failure on hard suites and degrades clean-task performance. RobustVLA (Zhang et al., 2025b) adds Jacobian and smoothness regularization during reinforcement-learning post-training but evaluates only natural perturbations, leaving its adversarial behavior implicit. SafeVLA (Zhang et al., 2025a) reformulates the problem as a constrained Markov decision process requiring full RL retraining of the policy and produces no anomaly score the operator can route into a safety controller. Backdoor-side work, BadVLA (Zhou et al., 2025) for objective-decoupled backdoors and VLA-Fool (Yan et al., 2025) for multimodal alignment failures, continues to expand the attack surface but proposes no detection-side defense. Concurrent embodied-robustness work studies broader benchmarks (Fei et al., 2025; Chen et al., 2024); certified approaches such as randomized smoothing for control have not been demonstrated at VLA backbone scale and would impose a multiplicative inference cost incompatible with real-time control loops. *In contrast,* PAC-CTCN occupies the unfilled detection slot: no VLA modification, no clean-performance degradation, and a per-sample anomaly score backed by a formal bound (Prop. 1).

L.2 Classical adversarial-input detection and why it does not transfer

The image-classifier detection literature has matured along several lines. *Preprocessing-based detection* (feature squeezing (Xu et al., 2018), input purification (Guarnizo Orjuela et al., 2026) as the VLA-specific instantiation) flags inputs whose predictions change under a transformation (JPEG, blur, learned restorer); these methods are bypassed by adaptive attacks (Carlini & Wagner, 2017) and degrade clean performance when the transformation is strong enough to neutralise large perturbations (Wang et al., 2025; Fei et al., 2025). *Density-based detection* (Mahalanobis distance on intermediate features (Lee et al., 2018)) fits class-conditional Gaussians; under embodied control there are no class labels and the geometry shifts with scene composition. *Reconstruction-based detection* (MagNet (Meng & Chen, 2017)) pairs an autoencoder with a downstream detector; its decision boundary is static and Carlini & Wagner (2017) show it is bypassed under white-box attack. *Geometry-based detection* (LID (Ma et al., 2018)) measures local intrinsic dimensionality of feature neighbourhoods; Athalye et al. (2018) establish the broader principle that any defense relying on hard-to-differentiate operations is vulnerable to gradient approximation (BPDA), and Tramèr et al. (2020) formalize the adaptive-attack methodology that has bypassed the entire family.

PAC-CTCN's predictive branch *differs from this lineage in that* $\hat{z}_t$ is computed from quantities (x_{t-1}, a_{t-1}) that are committed and stored before the attacker observes the current frame, so $\partial \hat{z}_t / \partial \delta_t = \mathbf{0}$ is not gradient masking, it is genuine causal independence (App. I, "Targeting the prediction $\hat{z}_t$ directly"). The discriminative branch is a learned classifier and *is* subject to the Carlini–Wagner critique; we report the white-box stress-test transparently in App. I and target the realistic gray-box deployment regime, where the public VLA is known but the proprietary detector is not. This regime mirrors how learned safety modules are actually deployed in production robotics: the VLA backbone is open-sourced and inspected, while the operator's anomaly module is trained on proprietary data and runs behind the same network boundary as the controller, an asymmetry the gray-box framing captures and the white-box stress test bounds.

L.3 Failure detection and runtime monitoring for embodied policies

A separate literature targets task-failure rather than adversarial-input detection. SAFE (Gu et al., 2025), the SOTA learned VLA failure detector, probes the VLA's last-layer latent features and learns a "failure zone" from successful and failed rollouts on training tasks, calibrated via functional conformal prediction; on multitask task-failure detection it attains AUROC 0.78–0.82 across OpenVLA, π_0, and π_0-FAST. FIPER (Römer et al., 2025) predicts failures of generative robot policies through random-network distillation and action-chunk entropy, evaluated on diffusion- and flow-matching policies; the authors explicitly leave VLA extension as future work. The structural distinction is sharp: a failure detector learns the natural-failure manifold of a policy and asks "does this look like a rollout that will fail?"; an adversarial detector must instead detect a specific class of *induced* feature displacements created by an optimising adversary. *Unlike* failure detectors, PAC-CTCN does not assume the perturbed input lies in any natural-failure region: a confidently-wrong action can be elicited by a perturbation whose latent features remain inside the success manifold (App. K.3, sub-chance inversion at π_0+LIBERO-Plus). Empirically, SAFE adapted to adversarial inputs falls 0.33–0.43 AUROC below PAC-CTCN across all conditions (Table 26). The two methods are therefore complementary rather than competing (App. G, "Complementarity"): a perturbation too small to trigger PAC-CTCN cannot displace features into SAFE's failure zone, and a natural failure SAFE catches has no adversarial signature.

L.4 Forward dynamics and predictive representation learning

Latent forward-dynamics models are a long-running thread in model-based reinforcement learning. Dreamer (Hafner et al., 2020) learns an end-to-end world model whose imagined rollouts train the policy. SPR (Schwarzer et al., 2021) regresses future latents to improve sample efficiency; BYOL-Explore (Guo et al., 2022) uses prediction error as a curiosity bonus, signalling novel or out-of-distribution states. TACO (Zheng et al., 2023) couples a temporal latent action-driven contrastive loss with action prediction for visual RL representations. The shared technical ingredient is a target-network architecture (Grill et al., 2020) in which a stop-gradient prevents latent collapse, the exact mechanism PAC-CTCN borrows (Eq. 7, Prop. 3). *In contrast,* the reframing PAC-CTCN introduces is one of *purpose*: in the RL lineage, prediction error is a signal to be minimized (training objective in Dreamer/SPR, curiosity bonus in BYOL-Explore); we instead exploit that adversarial perturbations *necessarily increase* prediction error on the present frame because $\hat{z}_t$ is computed from a strictly past state the attacker cannot move (Eq. 1). To our knowledge, PAC-CTCN is the first formally-grounded predictive-consistency detector for VLA adversarial inputs, and the first to convert action-conditioned latent prediction from a learning-time regularizer into a deployment-time detection bound (Prop. 1).

M Limitations and Negative Societal Impact

This appendix expands the limitations sketched in §6 into individual paragraphs and adds a negative-societal-impact discussion. We list known limitations honestly and in priority order of how they constrain real-world deployment.

N.1 Simulation-only evaluation. All experiments run in LIBERO (Liu et al., 2023) and LIBERO-Plus (Fei et al., 2025); no physical-robot deployment data is reported. Fei et al. (2025) document a substantial sim-to-real gap for VLA robustness more broadly, and Chen et al. (2024) note that simulation-trained adversarial perturbations sometimes fail to transfer through physical optics. We

expect both the prediction noise ε_0 and the adversarial feature displacement Δ to grow modestly under physical-camera noise, sensor lag, and lighting variation; whether the bound's operating regime $\Delta > 2\varepsilon_0$ continues to hold is the most consequential open question for deployment. Hardware-in-the-loop evaluation on a UR5- or Franka-class arm is the natural follow-up.

N.2 Single-step prediction horizon. The deployed detector uses one past frame ($k = 1$). The multi-step extension is theoretically sound (Prop. 2, App. B.3), but we have not characterized the regimes where $k = 1$ is empirically insufficient. Robot manipulation is piecewise-smooth with discrete events (contact, gripper open/close) that push the dynamics Lipschitz constant L_{dyn} close to one, so $k = 3$ on LIBERO yields only marginal AUROC improvement at $3\times$ the compute. On tasks with smoother dynamics, navigation, locomotion, soft-body interaction, or with longer effective horizons, $k > 1$ may become necessary; we do not provide an automatic selection rule for k.

N.3 Adversary class scope. Our adaptive-attack stress test (App. I) covers two constructions, both at matched compute budget: standard Projected Gradient Descent (PGD) with 100 steps and 10 random restarts (each restart an independent uniform initialisation inside the ℓ_∞ ball), step size $\alpha_{\mathrm{PGD}} = \varepsilon/10$ at $\varepsilon \in \{4, 8, 16\}/255$ across all three VLAs; and a theory-aware variant with the *same* 100-step / 10-restart budget that adds a term to the attack objective driving $f_\theta(x_t + \delta) \to \hat{z}_t$ to directly target the asymmetry of Eq. (1). We do not certify against (i) physical-world adversarial patches under printer-camera quantisation, (ii) BadVLA-style backdoor attacks (Zhou et al., 2025) where the adversary controls training data or model weights (Proposition 1 assumes test-time perturbation only and is silent on weight- or data-space compromise), or (iii) audio- or language-channel attacks (App. J, "Language-instruction attacks"). The bound itself is attack-agnostic in feature space: it depends only on the displacement Δ produced by the perturbation, regardless of how that perturbation was crafted.

N.4 Gray-box-only formal guarantee. Proposition 1 holds in the gray-box regime where the attacker knows the public VLA but not the proprietary detector. The proof assumes (i) feature-space Lipschitzness of the visual encoder (A1), with L_f estimated empirically rather than bounded analytically, and (ii) the empirical clean prediction error ε_0 measured on a held-out validation set rather than worst-case-bounded. Under full white-box attack with PGD at the smallest tested budget ($\varepsilon = 4/255$), the combined detector is evaded (AUROC 0.26–0.31, App. I). At the Phase II training budget ($\varepsilon = 16/255$) the combined-detector AUROC recovers to 0.95–0.998. This recovery reflects training-distribution coverage rather than guaranteed robustness: the discriminative branch has seen patch and Gaussian negatives at this ℓ_∞ scale during Phase II and learned to flag them. We say so plainly. This is an in-training-distribution memorisation effect adjacent to the obfuscated-gradients concern of Athalye et al. (2018), and it does not constitute a robustness claim. We explicitly do not claim white-box guarantees, aligning with Carlini & Wagner (2017); Tramèr et al. (2020) that learned detectors are bypassable in this strict regime; the gray-box framing is the regime where the formal claim holds.

N.5 Per-VLA training cost and version sensitivity. The detector must be adversarially fine-tuned per VLA backbone (Phase II, App. F, Table 18). Zero-shot detector transfer across VLAs drops AUROC by 0.10–0.17 (Table 22); 10 epochs of target-VLA fine-tuning recovers performance to within 0.003 of the diagonal. For a single deployed VLA, the one-time cost is approximately 40 A100-hours; multi-VLA fleets scale this linearly. A second cost we have not eliminated is version sensitivity: any non-trivial update to the protected VLA (RLHF iteration, fine-tune for a new task suite, backbone version bump) will shift the detector's expected ε_0 and Δ, and likely require Phase II re-fine-tuning. Sharing pre-trained predictive branches across VLAs with similar visual encoders (e.g., SigLIP-based families) is a plausible compute-saving extension we have not measured.

N.6 Detector inference cost and memory. PAC-CTCN adds 0.88 ms per timestep and 186 MB of GPU memory on a single A100 (App. F, Tables 15–17), an overhead of 0.59% in latency and 1.33% in memory relative to the protected OpenVLA. The detector's ViT-Small encoder is separate from the VLA's vision backbone and therefore duplicates rather than shares the visual encoder forward pass. This overhead is sufficient for laboratory and workstation-class deployment but is non-trivial for embedded or resource-constrained settings such as Jetson-class hardware. INT8 quantisation

of the ViT-Small encoder would in principle roughly halve the cost; we report this as an untested estimate rather than a measured result.

N.7 Threshold calibration and reported variance. Our headline AUROCs are evaluated as threshold-free ranking metrics; deployment requires a calibrated operating point, for which we use $\tau = \mu_s - 3\sigma_s$ on a held-out clean validation set. This 3σ rule is conservative for sub-Gaussian clean scores but is not adaptive to per-task drift; production deployment would require periodic re-calibration on recent clean rollouts. We recommend re-calibrating at every task-family change or every $\sim$1,000 rollouts, whichever comes first, and we do not characterize the failure mode under sustained, slow drift. The headline numbers are means over three seeds (42, 123, 456): the standard deviation on the worst cell (π_0+LIBERO-Plus, AUROC 0.903) is ± 0.020, giving an approximate 95% confidence interval of $[0.86, 0.94]$. Smaller-AUROC cells therefore have meaningfully wider intervals than the saturated $\{S, O, G\}$ cells; the per-regime $\Delta/(2\varepsilon_0)$ ratio is correspondingly smaller and the bound's safety margin tighter.

Negative societal impact. PAC-CTCN is a defensive monitor, and we identify no direct dual-use trajectory: it does not enable new attack capabilities, and its outputs are signals to a human or system controller rather than autonomous interventions. The most plausible negative-impact pathway is a false sense of security, an operator may treat a pass-through verdict as a robustness certificate and reduce human oversight. The gray-box-only formal guarantee (N.4) and demonstrated white-box vulnerability make this unjustified, and PAC-CTCN should complement rather than replace physical, system-level, and procedural safety controls. A second pathway is adversary-side misuse: an attacker who knows the detector exists (even without access to its weights) can in principle reverse-engineer detection regions through repeated probing and shift attack distributions away from them. The standard mitigation is to keep detector weights private and to monitor query patterns for anomalous probing behavior. We recommend that practitioners deploying PAC-CTCN publish detector failure modes alongside successes, and we encourage research on detector-update protocols that respond to evolving attack distributions in production.

N Closed-loop Safety Evaluation

This appendix reports the full closed-loop evaluation summarized in §5.6. We connect the α=1 predictive detector to a terminal safe-halt controller with a two-frame debounce, calibrate the threshold on clean validation ($\tau = \mu_s - 3\sigma_s$; OpenVLA $\tau = 0.988$), and launch sustained UADA at $\varepsilon = 16/255$ from a random onset over seeds 42/123/456 with the $70/15/15$ trajectory-level split and mean rollout length ≈ 210 frames. Action divergence is any attack-induced departure from the clean action. An unsafe event is MuJoCo-observable harm, namely an unintended object drop or displacement, or the end-effector leaving the workspace envelope, our collision proxy.

Table 28: Full closed-loop safety metrics under sustained UADA (ε=$16/255$), three seeds. Cells with an arrow are without detector $\rightarrow$ with detector plus terminal halt; values are percentages unless noted. The halt contains harm rather than restoring task success.

Rollout-level metric	OpenVLA	Octo	π_0
Clean task success (no det $\rightarrow$ det+halt)	84.2 $\rightarrow$ 81.6	82.9 $\rightarrow$ 79.9	96.4 $\rightarrow$ 92.4
Clean false-stop rate	3.1	3.6	4.2
Attacked task success (no det $\rightarrow$ det+halt)	2.7 $\rightarrow$ 2.4	4.1 $\rightarrow$ 3.8	5.3 $\rightarrow$ 5.1
Attacked action divergence (no det $\rightarrow$ det)	93.1 $\rightarrow$ 5.9	91.4 $\rightarrow$ 7.2	93.6 $\rightarrow$ 8.4
Attacked unsafe-event rate (no det $\rightarrow$ det)	88.7 $\rightarrow$ 4.8	86.1 $\rightarrow$ 6.1	84.3 $\rightarrow$ 7.0
Attacked safe-halt rate	91.7	89.0	86.5
Missed-detection rate	5.9	7.2	8.4
Detection before first divergent action	94.1	92.8	91.6
Median time-to-detection (frames)	2	3	3

The measurements show containment rather than recovery. Divergence and unsafe events fall sharply, and $91.7/89.0/86.5\%$ of attacked rollouts end in safe halts, at clean-success costs of $2.6/3.0/4.0$ points. Post-detector divergence and unsafe-event rates worsen in the same model order as predictive-only AUROC, a descriptive consistency rather than a causal claim. Detection-before and missed-

detection are complementary, and safe halt, task success, and missed detection partition the attacked outcomes. We make no physical-robot safety claim; the operating quantities can be remeasured on hardware before the detector is trusted.

O Small-budget White-box Attacks: Damage versus Evasion

Under full white-box access the predictive detector is evaded at the smallest budget ($\varepsilon = 4/255$, AUROC 0.26 in Appendix I). We test whether such an evading attack is also harmful to the robot. Keeping the predictive score above threshold caps the feature displacement, since $s_{\text{pred}} \geq \tau$ gives $D_t \leq \varepsilon_0 + 2\sqrt{1-\tau} \approx 0.32$ on OpenVLA, about five times below the mean displacement of effective attacks.

Table 29: Damage and detectability both scale with the budget (OpenVLA, LIBERO-{S,O,G}). Gray-box detection AUROC and unconstrained UADA task success (clean 84.2%) at three budgets.

	ε=4/255	ε=8/255	ε=16/255
Unconstrained UADA task success	61.4	27.8	2.7
Gray-box detection AUROC	0.884	0.941	0.999

On the evading set ($s_{\text{pred}} \geq \tau = 0.988$) at $\varepsilon = 4/255$, only 8.6% of perturbations change the executed action beyond clean seed variance, and task success drops 3.4 points, against 22.8 points for the unconstrained attack at the same budget. In this evaluated setting the attacks that evade cause substantially less action and task damage. We do not promote this into a universal white-box safety guarantee.

P Persistent and History-poisoning Attacks

The one-step proposition applies at the first attacked step, where the stored reference is clean. At later steps a coordinated attacker can influence both the stored reference and the current frame, which the one-step statement does not certify. We therefore test persistence end to end. Trajectory-level detection, defined as any flag before failure on OpenVLA, is 99.8% for sustained UADA from onset at $16/255$ and 93.7% for a white-box evade-then-poison variant at $4/255$. Fully evading trajectories cost only 4.1 task-success points, consistent with the displacement cap of Appendix O. These results exercise the causal design rather than escaping it, and we do not claim the one-step proof alone covers poisoned histories.

Q Standard Detection Baselines

We benchmark standard adversarial-detection methods on the identical adversarial test set (OpenVLA, LIBERO-{S,O,G}). A Mahalanobis feature detector (Lee et al., 2018), a deep k-nearest-neighbor feature-distance detector (Sun et al., 2022), and an energy/max-logit detector (Liu et al., 2020) are single-frame density or energy scores that lack the causal reference. Each lands below even our prediction-error-only ablation, which uses no adversarial training and no discriminative branch, which supports the structural motivation of §2.

Table 30: Standard detection baselines on adversarial inputs (OpenVLA, LIBERO-{S,O,G}, AUROC). The single-frame baselines lack the causal reference and trail the prediction-error-only ablation.

Method	AUROC (adv)
Energy / max-logit (Liu et al., 2020)	0.61
Deep k-NN feature distance (Sun et al., 2022)	0.66
Mahalanobis (Lee et al., 2018)	0.71
Prediction-error only (Phase I, no adv. training)	0.877
PAC-CTCN	**0.999**