TMT: RUNTIME BACKDOOR DETECTION FOR VISION-LANGUAGE-ACTION POLICIES ON UNSEEN TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Backdoored vision-language-action (VLA) policies can preserve benign task performance while producing malicious actions when a trigger appears. Detecting such activation is difficult because malicious behavior can comprise individually plausible actions, while unfamiliar tasks introduce legitimate changes in observations and behavior. We introduce TMT, a runtime backdoor detector based on **T**oken **M**anifold and latent **T**ransition modeling. Trained on benign rollouts, its two branches assess input-token structure and prediction errors in adjacent-layer latent dynamics. A suspicious rollout identified by the token manifold branch, once confirmed through latent deviations, guides transition selection for subsequent monitoring. We further explore policy purification through self-distillation: a frozen copy of the backdoored policy provides benign-input actions to supervise a student on paired benign and triggered observations, without requiring a separate clean reference policy. For evaluation, we adapt traditional backdoor detectors and repurpose anomaly and failure detection methods as VLA backdoor detectors. In a post-hoc comparison with ten baselines, TMT achieves state-of-the-art backdoor detection performance on unseen tasks across three VLA backdoor attacks.

1 INTRODUCTION

Vision-Language-Action (VLA) policies translate visual observations and language instructions into robot actions (Kim et al., 2025; Wang et al., 2026; Intelligence et al., 2025). A compromised policy can retain its performance on benign inputs while changing its behavior when a trigger appears. Recent attacks demonstrate that backdoor activation can disrupt task execution, redirect the robot toward an attacker-specified goal, or induce an undesired low-level action (Zhou et al., 2026; 2025; Xu et al., 2025b). For example, backdoor activation could cause a robot to suddenly drop a knife it is holding, posing a physical safety risk. This coupling between backdoor activation and physical execution motivates runtime backdoor detection in VLA policies.

Existing inference-time backdoor detectors for classifiers assess suspicious inputs by examining how class predictions respond to input perturbations (Gao et al., 2019; Liu et al., 2023b). VLA policies, however, generate task-conditioned control actions, and backdoor activation need not produce a fixed target output across states and timesteps. We consider four requirements for runtime backdoor detection in this setting. ① The detector must learn from benign rollouts without prior knowledge of the trigger pattern or access to the VLA policy's original training process. ② The detector must distinguish backdoor-induced behavior from benign execution even when both involve the same actions. This distinction depends on which goal the resulting sequence serves. ③ The detector must generalize to unseen tasks without mistaking legitimate changes in observations and behavior for backdoor activation. ④ The detector must identify backdoor activation with minimal delay after trigger onset.

To address these challenges, we introduce **TMT**, an inference-time backdoor detector that combines **T**oken **M**anifold and latent **T**ransition modeling using internal representations of the deployed VLA policy (Figure 1). TMT examines both the structure of input-token representations and how these representations evolve across backbone layers. Our t-SNE visualizations show that the input-token representations of triggered queries are shifted relative to those of benign queries in the projected space, while the two groups still overlap (Figure 2). This motivates examining layer-wise transitions alongside token structure. Trained on benign rollouts, the token manifold branch assesses token

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

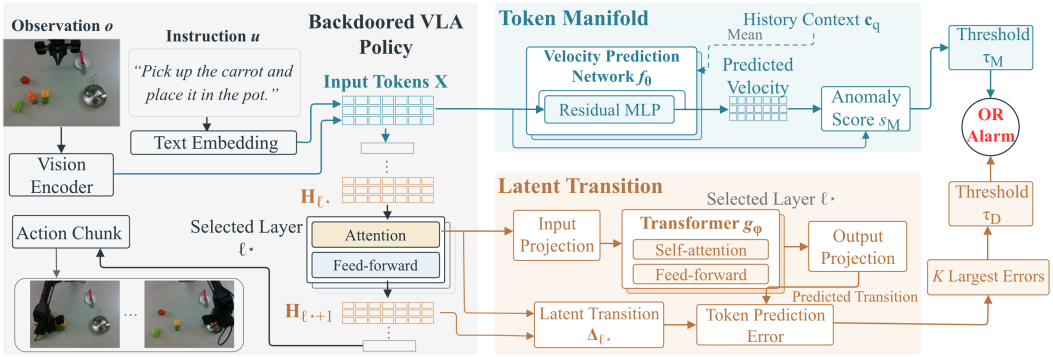


Figure 1: Overview of TMT for runtime backdoor detection. On a frozen VLA backbone, the token manifold branch assesses input-token structure conditioned on the preceding query, while the latent transition branch scores prediction errors at a selected pair of adjacent layers. An alarm is raised when either branch's score exceeds its separately calibrated threshold.

structure conditioned on the preceding policy query, while the latent transition branch scores prediction errors in adjacent-layer latent dynamics. Both branches evaluate each policy query, and TMT raises an alarm when either score exceeds its calibrated threshold.

In this paper, we focus on detecting backdoor activation on unseen manipulation tasks, with TMT fitted using benign rollouts from seen tasks. For comparison, we adapt traditional backdoor detectors (Gao et al., 2019; Liu et al., 2023b; Hou et al., 2025; Hu et al., 2026) and repurpose anomaly and failure detection methods (Ruff et al., 2018; Burda et al., 2018; He et al., 2024; Xu et al., 2025a; Römer et al., 2026) for detecting backdoor activation (see Appendix A). Detectors are fitted using benign rollouts from three LIBERO-10 tasks (Liu et al., 2023a) and evaluated on the remaining tasks under BadVLA (Zhou et al., 2026), GoBA (Zhou et al., 2025), and DropVLA (Xu et al., 2025b). In the post-hoc comparison, TMT achieves the highest or joint-highest trigger detection rates (TDRs) of 100.00%, 83.43%, and 73.60%, respectively, with observed false-rejection rates (FRRs) of 0.57%, 8.57%, and 0.00%. Branch ablations show that, for GoBA and DropVLA, the token manifold branch provides broader detection coverage on tasks represented during fitting, whereas the latent transition branch provides broader coverage on unseen tasks. We further examine detection in real-world robot execution, and observe that TMT achieves a TDR of 100% with an observed FRR of 0%.

Beyond detection, we explore policy purification through self-distillation without a separate clean reference policy. A frozen copy of the backdoored policy serves as the teacher, and the student is initialized from the same checkpoint. We execute the teacher's benign-input actions in both benign and triggered environments, collecting paired observations under a shared action history. The student is trained to reproduce each benign-input action from both observations. Results show that original-task success rates under triggered inputs increase to 92.2–95.5%, while benign-input success rates change by at most 2.2%. Our contributions are threefold:

- We introduce TMT, an inference-time VLA backdoor detector that combines input-token structure and latent dynamics learned from benign rollouts.
- We propose policy purification through self-distillation without a separate clean reference policy, using the victim's benign-input actions to supervise paired benign and triggered observations.
- We adapt existing detection methods to VLA backdoor detection and evaluate them under three kinds of VLA backdoor attacks. Our post-hoc results show strong backdoor detection performance on unseen tasks.

2 Related Work

Backdoor Attacks on VLA. Backdoor attacks on VLA policies implant trigger-dependent control behaviors while preserving benign performance. BadVLA (Zhou et al., 2026) uses objective-decoupled optimization to separate triggered and benign visual representations and induce control

deviations. GoBA (Zhou et al., 2025) poisons demonstrations with physical-object triggers to redirect high-level goals, whereas DropVLA (Xu et al., 2025b) relabels selected action targets to induce low-level failures such as premature gripper opening. Attack demonstrations appear in Appendix E. These attacks expose the need for runtime detection across different malicious behaviors. TMT addresses this need by monitoring token manifolds and latent transitions without assuming a specific target action.

Traditional Backdoor Detection. Existing detectors screen inputs through prediction responses or representation statistics. STRIP (Gao et al., 2019) measures prediction entropy under perturbations, while TeCo (Liu et al., 2023b) compares the corruption severity needed to change predicted labels across corruption types. DeDe (Hou et al., 2025) uses decoder reconstruction errors to screen inputs to self-supervised encoders; DUP-MS (Hu et al., 2026) combines class-agnostic Mahalanobis distances with spectral statistics of inter-layer feature differences. Concurrent work, TrustVLA (Fu et al., 2026), uses token- and layer-wise epistemic uncertainty for VLA detection. STRIP, TeCo, DeDe, and DUP-MS were originally developed for image or language models. We adapt them to VLA backdoor detection, but our post-hoc comparison shows limited detection on unseen tasks, whereas TMT achieves higher TDRs.

Anomaly and Failure Detection. Learning a reference distribution from normal data provides a basis for detecting previously unseen deviations. Deep SVDD (Ruff et al., 2018) scores learned representations by distance from a hypersphere center, while RND (Burda et al., 2018) measures prediction error against fixed random features. ReDiffuser (He et al., 2024) uses RND confidence to guide diffusion-based planning. LogPZO (Xu et al., 2025a) models successful-rollout observations with a flow and scores their inferred latent noise. FIPER (Römer et al., 2026) requires both embedding-space RND and action-chunk entropy to exceed calibrated thresholds over short windows. These methods struggle to distinguish benign execution on unseen tasks from trigger-induced execution. Our post-hoc results show that TMT distinguishes these cases more effectively.

3 Problem Formulation

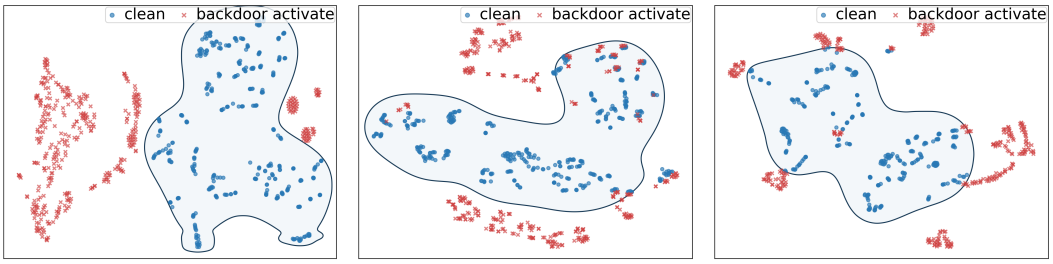
We study inference-time backdoor detection for a potentially backdoored VLA policy π. At each query q, the policy receives an observation o_q and a language instruction u, and produces an action chunk $\boldsymbol{A}_q = \pi(o_q, u)$. Executing these actions changes the environment and determines subsequent observations, yielding a rollout of Q queries indexed by $q = 0, \ldots, Q-1$. A backdoor trigger may become active at any query and alter subsequent actions and the resulting trajectory.

Defender Capability. The defender has access to trusted benign rollouts and can inspect the policy's internal representations during inference. Detector fitting and calibration use only benign rollouts, without access to triggered training examples, the trigger specification, or an uncompromised reference policy. Here, benign rollouts denote the absence of a backdoor trigger, regardless of whether the policy successfully completes the task.

Defender Objective. The defender aims to detect backdoor activation online. Deployment may involve unseen tasks during detector training, requiring cross-task generalization. We aim to identify triggered rollouts with a high detection rate and low trigger-relative detection latency while controlling the rollout-level false-rejection rate (FRR), defined as the fraction of benign rollouts with at least one alarm.

4 Backdoor Detection

We introduce TMT, as shown in Figure 1, a backdoor detector that examines each policy query through two complementary branches. The token manifold branch assesses whether input-token representations conform to the distribution learned from benign queries. The latent transition branch assesses whether observed adjacent-layer transitions agree with those learned from benign inference. Both branches are trained on benign rollouts collected by executing the victim policy without triggers, while the policy remains frozen. After training, we calibrate thresholds using held-out benign rollouts. During online detection, an alarm is raised when either branch's score exceeds its threshold.



(a) BadVLA (Zhou et al., 2026) (b) GoBA (Zhou et al., 2025) (c) DropVLA (Xu et al., 2025b)

Figure 2: Visualizations of input tokens from backdoored OpenVLA-OFT (Kim et al., 2025) under BadVLA, GoBA, and DropVLA. Blue circles denote clean queries, and red crosses denote backdoor-active queries.

4.1 Motivation

Benign Token Representation. Existing analyses (Hu et al., 2026; Wang & Arora, 2025) of backdoor attacks at the input-query level have shown that triggers can induce anomalous representation patterns. With privileged access to the labels of benign and malicious queries, the t-SNE visualization (Van der Maaten & Hinton, 2008) in Figure 2 shows that most benign queries cluster in similar regions. Motivated by these observations, we hypothesize that benign query representations possess a learnable manifold structure and that triggers can shift query representations away from this manifold. We therefore model the benign representation distribution, which we refer to as the benign token manifold, to identify trigger-induced deviations.

Inference Dynamics Consistency. Just as benign query representations can be characterized by a manifold, their layer-to-layer transitions during inference can also be modeled. Yang et al. (2023) showed that suppressing skip connections in critical layers can reduce backdoor attack success. Inspired by this, we hypothesize that the layer-to-layer transitions of triggered queries deviate from those observed during benign inference. This motivates learning benign latent transitions and examining whether the observed latent dynamics are consistent with these learned relationships.

4.2 Token Manifold

Let $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ denote the token sequence entering the backbone of the potentially backdoored VLA, where N is the number of valid tokens and d is the embedding dimension. VLA tasks involve sequential interaction, so we use the preceding query to provide temporal context. We condition a velocity prediction network $f_{\boldsymbol{\theta}}$, with trainable parameters $\boldsymbol{\theta}$, on a history context $\boldsymbol{c} \in \mathbb{R}^d$, obtained by averaging the valid tokens from the previous policy query: $\boldsymbol{c}_q = \mathrm{mean}(\boldsymbol{X}_{q-1})$. For the first policy query, we initialize the history context as $\boldsymbol{c}_0 = \mathbf{0}$.

We sample a noise matrix $\boldsymbol{\epsilon} \in \mathbb{R}^{N\times d}$ with elements drawn independently from $\mathcal{N}(0,1)$, and independently sample a transport time $t \sim \mathrm{Uniform}(0,1)$. We construct a linear interpolation between the benign token sequence and Gaussian noise:

$$\boldsymbol{Z} = (1-t)\boldsymbol{X} + t\boldsymbol{\epsilon}. \tag{1}$$

We train the velocity prediction network $f_{\boldsymbol{\theta}}$ with a flow-matching objective:

$$\mathcal{L}_{\mathrm{M}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{c}),\boldsymbol{\epsilon},t}\left[\frac{\|f_{\boldsymbol{\theta}}(\boldsymbol{Z},t;\boldsymbol{c}) - (\boldsymbol{\epsilon}-\boldsymbol{X})\|_F^2}{Nd}\right], \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The expectation is taken over benign queries and their history contexts $(\boldsymbol{X},\boldsymbol{c})$, independently sampled Gaussian noise $\boldsymbol{\epsilon}$, and transport time t.

At inference, we evaluate the velocity predictor at $t=0$, where $\boldsymbol{Z}=\boldsymbol{X}$, to assess the original query representation without noise perturbation. Since $\boldsymbol{\epsilon}$ is independent of $(\boldsymbol{X},\boldsymbol{c})$ and has zero mean, the conditional mean velocity target at this endpoint is $-\boldsymbol{X}$. We therefore define the anomaly score as

$$s_{\mathrm{M}}(\boldsymbol{X};\boldsymbol{c}) = \frac{\|\boldsymbol{X} + f_{\boldsymbol{\theta}}(\boldsymbol{X},0;\boldsymbol{c})\|_F^2}{Nd}. \tag{3}$$

This score requires a single forward pass through $f_{\boldsymbol{\theta}}$ without noise sampling. Larger scores indicate greater deviation from the learned benign representation patterns and provide evidence of potential backdoor activation.

4.3 Latent Transition

The anomalous tokens can induce the backdoor pattern of the compromised VLA. We model how token representations evolve across backbone layers during benign inference. Let $\boldsymbol{H}_\ell \in \mathbb{R}^{N\times d}$ denote the token sequence at backbone depth $\ell = 0, \ldots, L-1$, with $\boldsymbol{H}_0 = \boldsymbol{X}$. We define the latent transition between adjacent layers as

$$\boldsymbol{\Delta}_\ell = \frac{\boldsymbol{H}_{\ell+1} - \boldsymbol{H}_\ell}{\sigma_\ell}, \tag{4}$$

where $\sigma_\ell > 0$ is a fixed layer-specific scale estimated from benign training data to normalize differences in transition magnitude across layers. We train a latent transition predictor $g_{\boldsymbol{\phi}}$ with parameters $\boldsymbol{\phi}$ to estimate $\boldsymbol{\Delta}_\ell$ from $\boldsymbol{H}_\ell$:

$$\mathcal{L}_{\mathrm{D}}(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{X},\ell}\left[\frac{\|g_{\boldsymbol{\phi}}(\boldsymbol{H}_\ell, \ell) - \boldsymbol{\Delta}_\ell\|_F^2}{Nd}\right], \tag{5}$$

where the expectation is taken over clean queries $\boldsymbol{X}$ and the layer index ℓ, sampled independently and uniformly from the adjacent-layer transitions.

Transition Selection. We fit the detector using verified benign rollouts and initialize transition selection on tasks represented in this training set. The token manifold branch flags suspicious rollouts during subsequent executions of these tasks. Once a flagged rollout is confirmed as a backdoor incident, we compare its prediction errors across all adjacent-layer transitions with the corresponding benign references. This confirmation uses only detector scores and benign reference statistics and does not require an externally supplied malicious label (see Appendix F). Among transitions exhibiting significant deviations, we select the one with the largest standardized excess and denote its index by $\ell^\star$, as shown in Figure 1. The selected transition remains fixed during subsequent online monitoring, including rollouts from unseen tasks. Both branches remain active, and either branch can independently raise an alarm.

For each subsequent policy query, we compute the consistency score at the selected transition by averaging the K largest token prediction errors:

$$s_{\mathrm{D}}(\boldsymbol{H}_{\ell^\star}, \ell^\star) = \frac{1}{Kd}\sum_{i\in\mathcal{I}_K} \|[g_{\boldsymbol{\phi}}(\boldsymbol{H}_{\ell^\star}, \ell^\star) - \boldsymbol{\Delta}_{\ell^\star}]_i\|_2^2 . \tag{6}$$

Here, $\mathcal{I}_K$ contains the indices of the K tokens with the largest prediction errors, and $[\cdot]_i$ denotes the i-th row. We normalize this score by a fixed error scale estimated from benign queries at the selected transition. Larger normalized scores indicate greater deviation from benign latent dynamics.

4.4 Calibration and Online Detection

After training $f_{\boldsymbol{\theta}}$ and $g_{\boldsymbol{\phi}}$, we fix these two networks and calibrate the branch-specific thresholds to enable online detection. After fitting the detection network and selecting the monitored transitions, We then use the retained benign queries to calibrate branch-specific detection thresholds for online monitoring.

For the token manifold branch, we set the threshold to the empirical 0.99 quantile of the benign rollout maximum scores plus a margin proportional to their median absolute deviation (MAD) (Rousseeuw & Croux, 1993):

$$\tau_{\mathrm{M}} = \mathrm{Quantile}_{0.99}\left[\max_q s_{\mathrm{M}}(\boldsymbol{X}_q; \boldsymbol{c}_q)\right] + \mathrm{MAD}\left[\max_q s_{\mathrm{M}}(\boldsymbol{X}_q; \boldsymbol{c}_q)\right]. \tag{7}$$

For the latent transition branch, we use the maximum scores from held-out benign rollouts to compute the empirical tail probability of the current online score. Its negative logarithm gives the calibrated detection score. We set τ_{D} to the negative logarithm of a prescribed tail-probability cutoff.

Online Detection. In subsequent rollouts, the token manifold branch evaluates $s_{\mathrm{M}}(\boldsymbol{X}; \boldsymbol{c})$ at each query, while the latent transition branch monitors the selected layer $\ell^\star$. Let $\tilde{s}_{\mathrm{D}}$ denote its calibrated detection score defined above. An alarm is raised when either score exceeds its threshold:

$$b = \mathbf{1}[s_{\mathrm{M}}(\boldsymbol{X}; \boldsymbol{c}) > \tau_{\mathrm{M}} \ \vee \ s_{\mathrm{D}}(\boldsymbol{H}_{\ell^\star}, \ell^\star) > \tau_{\mathrm{D}}] \,. \tag{8}$$

Here, $b = 1$ indicates a detection alarm.

5 Backdoor Purification

To suppress backdoor-induced behavior while preserving benign task performance, we propose a purification method based on policy self-distillation with teacher-driven paired rollouts. Here, the teacher is a frozen copy of the original backdoored policy π, and the student $\pi_{\boldsymbol{\psi}}$ is initialized from the same checkpoint, with trainable parameters $\boldsymbol{\psi}$. The teacher's actions on benign observations provide supervision for preserving benign behavior and correcting actions under malicious observations.

Teacher-driven Paired Rollouts. At each policy query, the teacher predicts an action chunk from the benign observation. We execute the same actions in both the benign and malicious environments, collecting paired observations under a shared action history. Let o and $\tilde{o}$ denote the benign and malicious observations at corresponding queries, with the same language instruction u. The teacher's action chunk $\pi(o, u)$ advances both environments and provides the supervision target for both observations.

Action Distillation. We train the student to reproduce the teacher's benign-input actions from both observations:

$$\mathcal{L}_{\mathrm{P}}(\boldsymbol{\psi}) = \mathbb{E}_{(o,\tilde{o},u)} \left[\|\pi_{\boldsymbol{\psi}}(o, u) - \pi(o, u)\|_F^2 + \|\pi_{\boldsymbol{\psi}}(\tilde{o}, u) - \pi(o, u)\|_F^2 \right] . \tag{9}$$

The expectation is taken over paired observations and their shared language instructions from the collected rollouts. The first term preserves the teacher's behavior on benign inputs, while the second trains the student to reproduce the corresponding benign actions in the presence of a trigger. After training, the student is used throughout each evaluation rollout.

6 Experiments

6.1 Experimental Setup

Attacks. We consider three attacks: BadVLA (Zhou et al., 2026), GoBA (Zhou et al., 2025), and DropVLA (Xu et al., 2025b). In our experiments, BadVLA uses a white patch as the trigger and aims to cause task failure. GoBA targets high-level goals, using a physical "toxic box" as the trigger to redirect the VLA policy toward an attacker-specified goal. DropVLA targets low-level actions using the textual trigger "carefully" together with a digital red-dot trigger. When both triggers are present, it induces the VLA policy to open the gripper, causing task failure. Details of the attack performance can be found in Appendix H, and demonstrations can be found in Appendix E.

Settings. For simulation experiments, we evaluate backdoor detection on OpenVLA-OFT (Kim et al., 2025), VLA-Adapter (Wang et al., 2026), and $\pi_{0.5}$ (Intelligence et al., 2025) using the LIBERO benchmark (Liu et al., 2023a). For real-world experiments, we deploy $\pi_{0.5}$ on a Trossen Robotics WidowX AI arm [1] and collect data for four tasks. Real-world demonstrations are shown in Figure 5 in Appendix G.

Baselines. We compare against three groups of baselines. Traditional backdoor detectors include STRIP (Gao et al., 2019), TeCo (Liu et al., 2023b), DeDe (Hou et al., 2025), and DUP-MS (Hu et al., 2026). Anomaly and failure detection methods include Deep SVDD (Ruff et al., 2018), RND (Burda et al., 2018), ReDiffuser (He et al., 2024), LogpZO (Xu et al., 2025a), and FIPER (Römer et al., 2026). We also compare with concurrent work TrustVLA (Fu et al., 2026), which specifically targets inference-time backdoor detection in VLA models. All details related to the VLA implementation can be found in Appendix A.

[1] www.trossenrobotics.com

Table 1: Post-hoc comparison on tasks unseen during detector fitting, using backdoored OpenVLA-OFT. We report rollout-level AUC, TDR, FRR, and mean detection latency in policy queries (Query), measured from trigger onset to the first alarm at or after onset. Latency is averaged over malicious rollouts with such an alarm; – indicates that none occur. Bold and underlining indicate the highest and second-highest distinct values, respectively, for AUC and TDR.

	BadVLA				GoBA				DropVLA			
Method	AUC↑	TDR↑	FRR↓	Query↓	AUC↑	TDR↑	FRR↓	Query↓	AUC↑	TDR↑	FRR↓	Query↓
Backdoor Detection												
STRIP	<u>0.770</u>	<u>22.57</u>	0.00	13.44	<u>0.940</u>	4.00	0.00	14.00	0.749	0.80	2.00	30.50
TeCo	0.251	0.57	7.71	48.00	0.662	5.71	0.86	2.65	0.673	6.00	1.60	24.27
DeDe	0.535	0.57	3.43	30.00	0.631	2.00	6.86	36.14	0.367	0.00	0.00	–
DUP-MS	0.001	0.00	0.29	–	0.841	4.86	0.00	13.71	0.570	1.60	2.00	39.00
TrustVLA	**1.000**	**100.00**	0.57	0.00	0.268	18.00	50.86	12.06	0.475	0.40	0.40	–
Anomaly and Failure Detection												
Deep SVDD	0.033	0.00	43.71	–	0.336	<u>42.86</u>	42.86	0.00	0.638	20.00	16.80	0.00
RND	0.448	1.43	8.57	42.00	0.189	0.57	37.43	62.00	0.592	16.80	30.40	33.48
ReDiffuser	**1.000**	**100.00**	4.00	0.00	0.626	15.14	0.29	13.45	0.716	22.80	0.00	16.42
LogpZO	0.503	0.00	2.86	–	0.138	0.29	41.43	60.00	0.315	10.00	40.00	33.24
FIPER	0.000	0.00	1.71	–	0.297	5.14	1.14	51.06	<u>0.955</u>	<u>26.00</u>	0.00	49.00
Our Method												
TMT	**1.000**	**100.00**	0.57	0.00	**0.983**	**83.43**	8.57	0.28	**1.000**	**73.60**	0.00	5.68

Detector Architectures. The token manifold velocity predictor $f_{\boldsymbol{\theta}}$ uses four residual MLP blocks. The latent transition predictor $g_{\boldsymbol{\phi}}$ uses a two-layer Transformer encoder with eight attention heads per layer.

Evaluation Metrics. We evaluate detection at the rollout level. A rollout is flagged if an alarm occurs at any policy query. The trigger detection rate (TDR) and false-rejection rate (FRR) are defined as

$$\mathrm{TDR} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{malicious}}}, \qquad \mathrm{FRR} = \frac{N_{\mathrm{FP}}}{N_{\mathrm{benign}}}, \tag{10}$$

where N_{TP} and N_{FP} count flagged malicious and benign rollouts, respectively, and $N_{\mathrm{malicious}}$ and N_{benign} denote their respective totals. Higher TDR and lower FRR indicate better detection performance. For threshold-independent evaluation, we report the area under the ROC curve (AUC) (Fawcett, 2006). Higher AUC indicates better separation between malicious and benign rollouts. Detection latency is the difference in policy-query indices between trigger onset and the first alarm at or after onset, averaged over malicious rollouts with such an alarm. Lower latency indicates faster detection.

6.2 Main Experimental Results

We evaluate backdoor detection on tasks unseen during detector fitting using backdoored OpenVLA-OFT (Kim et al., 2025) on LIBERO-10 (Liu et al., 2023a), a suite of ten long-horizon manipulation tasks. The detectors are trained on benign backdoored VLA inference-time rollouts randomly selected from three of the tasks, while the remaining seven tasks form the evaluation task pool. This setup is designed to test whether the detector can identify backdoor activations that fall outside the range of tasks available during training.

In this reanalysis, as shown in Table 1, TMT achieves the highest or joint-highest AUC and TDR across the three attacks. For BadVLA, an untargeted attack that disrupts task execution, TMT achieves a TDR of 100% with an observed FRR of 0.57%. TrustVLA and ReDiffuser also achieve 100% TDR, with FRRs of 0.57% and 4.00%, respectively.

GoBA presents a different detection challenge: it redirects the policy toward an attacker-specified goal through a sequence of actions that can individually resemble benign behavior. Local actions such as approaching an object or opening the gripper can occur in both benign and malicious rollouts, while their role in the overall execution differs. TMT achieves an AUC of 0.983 and a TDR

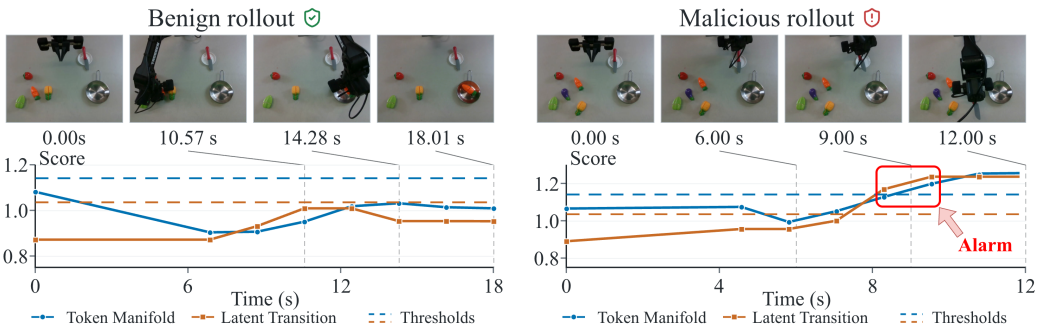


Figure 3: Benign and GoBA-triggered real-world behavior with detection-score traces. Dashed lines denote the calibrated thresholds of the token manifold and latent transition branches. The red box highlights threshold exceedances that trigger an alarm. Frames are drawn from separate demonstration recordings.

of 83.43%, with a FRR of 8.57%. Among the traditional backdoor detectors, STRIP and DUP-MS achieve AUCs of 0.940 and 0.841 but detect only 4.00% and 4.86% of malicious rollouts, respectively, at the calibrated thresholds. The AUCs of the anomaly and failure detection baselines range from 0.138 to 0.626. ReDiffuser's AUC and TDR drop from 1.000 and 100.00% on BadVLA to 0.626 and 15.14% on GoBA. Unlike the disrupted execution observed under BadVLA, GoBA can produce coherent, goal-directed behavior. This difference may help explain why detectors of execution anomalies struggle to distinguish its malicious rollouts from benign ones.

DropVLA targets low-level control by inducing gripper-opening actions when triggers are present. This behavior interferes with grasping and can cause repeated unsuccessful grasp attempts. We observe that TMT detects 73.60% of malicious rollouts without rejecting any benign rollout in the evaluated set. By comparison, FIPER achieves an AUC of 0.955 and the second-highest TDR of 26.00% at the same observed FRR of 0%. Deep SVDD detects 20.00% of malicious rollouts, accompanied by an FRR of 16.80%.

Additional results on $\pi_{0.5}$ (Intelligence et al., 2025) and VLA-Adapter (Wang et al., 2026) show that TMT achieves the highest TDR on both victim models (Appendix C). We further consider a privileged all-seen setting in which benign rollouts from all eligible tasks are available for detector fitting (Appendix B). In this post-hoc comparison, several existing backdoor, anomaly, and failure detectors also achieve high detection rates, while TMT attains the highest or joint-highest AUC across all three attacks.

6.3 Real-world Experiments

We evaluate real-world backdoor detection against GoBA on four pick-and-place tasks using $\pi_{0.5}$ deployed on a Trossen Robotics WidowX AI arm (see Appendix G for attack demonstrations). We train the detector using benign rollouts from two randomly selected tasks and evaluate benign and GoBA-triggered rollouts on the other two tasks, with 30 rollouts per evaluation task.

Table 2 shows that TMT achieves an AUC of 1.000 and detects all evaluated malicious rollouts without rejecting any benign rollout. RND matches TMT on all three metrics, while DUP-MS, ReDiffuser, and FIPER also achieve 100% TDR but incur higher FRRs. Figure 3 illustrates benign and GoBA-triggered behavior alongside example detection-score traces. Both branches remain below

Table 2: Real-world GoBA detection on unseen tasks.

Method	AUC↑	TDR↑	FRR↓
Backdoor Detection			
DeDe	0.833	0.00	10.00
DUP-MS	**1.000**	**100.00**	66.67
TrustVLA	0.528	0.00	8.33
Anomaly and Failure Detection			
Deep SVDD	0.917	83.33	16.67
RND	**1.000**	**100.00**	**0.00**
ReDiffuser	**1.000**	**100.00**	16.67
LogpZO	**1.000**	66.67	**0.00**
FIPER	0.958	**100.00**	26.67
Our Method			
TMT	**1.000**	**100.00**	**0.00**

their respective thresholds in the benign example, whereas the malicious trace exhibits threshold exceedances that activate the alarm.

Real-time deployment also constrains the number of policy evaluations available for each observation. STRIP (Gao et al., 2019) evaluates predictions across multiple perturbed inputs, while TeCo (Liu et al., 2023b) probes prediction changes across corruption types and severity levels. The repeated policy evaluations required by their VLA adaptations exceed the per-query time budget of our robotic setup, so we exclude them from the real-time comparison.

6.4 Ablation and Mechanism Analysis

The token manifold branch identifies suspicious rollouts through deviations from benign input-token structure. The latent transition branch uses this incident evidence to select an adjacent-layer transition for subsequent monitoring. Table 3 presents the DropVLA ablation; corresponding BadVLA and GoBA results appear in Appendix D.1.

Table 3: DropVLA branch ablation on OpenVLA-OFT (post-hoc). M: token manifold branch; L: latent transition branch.

Branch	TDR↑	FRR↓	Query↓
All-seen			
M only	**100.00**	1.25	9.45
L only	72.00	0.00	3.66
TMT (M+L)	**100.00**	1.25	2.43
Unseen			
M only	15.20	0.00	15.47
L only	**73.60**	0.00	6.25
TMT (M+L)	**73.60**	0.00	5.68

Detection Coverage. The relative contributions of the two branches vary with task availability. Under all-seen DropVLA, the token manifold branch detects all malicious rollouts, while the latent transition branch achieves a TDR of 72.00%. On unseen tasks, the latent transition branch achieves a TDR of 73.60%, compared with 15.20% for the token manifold branch. Thus, the token manifold branch provides broader detection coverage when the evaluation tasks are represented during fitting, whereas the latent transition branch becomes the main source of detection on unseen tasks. The other attacks provide similar evidence (Appendix D.1).

Detection Latency. The latent transition branch enables earlier alarms when combined with the token manifold branch. On all-seen DropVLA, combining the branches reduces mean latency from 9.45 to 2.43 queries while retaining 100.00% TDR and 1.25% FRR. On unseen DropVLA, joint detection retains the coverage of the latent transition branch and achieves a mean latency of 5.68 queries. GoBA also shows reduced latency, while BadVLA alarms occur at trigger onset (Appendix D.1).

6.5 Backdoor Purification

We evaluate policy self-distillation on LIBERO-10 by comparing the original backdoored policies with their purified students. In this preliminary evaluation, original-task success rates under attacks increase from 0.0%, 0.4%, and 0.0% to 92.2%, 94.8%, and 95.5% for BadVLA, GoBA, and DropVLA, respectively, while benign-input success rates change by -2.2%, -0.4%, and $+0.4\%$. Appendix H provides paired-rollout demonstrations and the complete results in Table 9.

7 Conclusion

We introduced TMT for runtime backdoor detection in VLA policies on tasks unseen during detector fitting. TMT models benign input-token manifold and adjacent-layer latent transitions on a frozen backbone. The post-hoc comparison under three popular backdoor attacks across three different victim models shows that these signals enable detection across different forms of backdoor activation. Branch ablations reveal broader token-manifold coverage on familiar tasks and a larger role for latent transitions on unseen tasks. We also explored policy self-distillation using paired benign and malicious observations as a purification extension. These results motivate examining both input representations and internal inference dynamics when monitoring VLA policies for backdoor activation.

AI USE STATEMENT

We used generative AI tools solely for language polishing and limited assistance with code generation. All AI-assisted content was reviewed and verified by the authors, and we take full responsibility for the final manuscript and all associated claims, code, and artifacts.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide our implementation code in the supplementary materials, together with demonstration videos of the real-world experiments.

REFERENCES

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Pinhan Fu, Xianda Guo, Xuetao Li, Wenke Huang, Ruilin Wang, Weiheng Zhao, Wei Sui, and Mang Ye. Trustvla: Mechanism-guided inference-time defense against vision-language-action backdoors. *arXiv preprint arXiv:2607.12571*, 2026.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pp. 113–125, 2019.

Nantian He, Shaohui Li, Zhi Li, Yu Liu, and You He. Rediffuser: Reliable decision-making using a diffuser with confidence estimation. In *Forty-first International Conference on Machine Learning*, 2024.

Sizai Hou, Songze Li, and Duanyi Yao. Dede: Detecting backdoor samples for ssl encoders via decoders. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20675–20684. IEEE, 2025.

Man Hu, Yahui Ding, Yatao Yang, Liangyu Chen, Yanhao Jia, and Shuai Zhao. Dup: Detection-guided unlearning for backdoor purification in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31059–31067, 2026.

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

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.017.

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

Xiaogeng Liu, Minghui Li, Haoyu Wang, Shengshan Hu, Dengpan Ye, Hai Jin, Libing Wu, and Chaowei Xiao. Detecting backdoors during the inference stage based on corruption robustness consistency. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16363–16372. IEEE, 2023b.

Ralf Römer, Adrian Kobras, Luca Worbis, and Angela Schoellig. Failure prediction at runtime for generative robot policies. *Advances in Neural Information Processing Systems*, 38:7631–7670, 2026.

Peter J Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical association*, 88(424):1273–1283, 1993.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. Vla-adapter: An effective paradigm for tiny-scale vision-language-action model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 40, pp. 18638–18646, 2026.

Yunjuan Wang and Raman Arora. Backdoor attacks in token selection of attention mechanism. In *Forty-second International Conference on Machine Learning*, 2025.

Chen Xu, Tony Khuong Nguyen, Emma Dixon, Christopher Rodriguez, Patrick Miller, Robert Lee, Paarth Shah, Rares Andrei Ambrus, Haruki Nishimura, and Masha Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies. In *Proceedings of Robotics: Science and Systems*, 2025a. doi: 10.15607/RSS.2025.XXI.073.

Zonghuan Xu, Jiayu Li, Yunhan Zhao, Xiang Zheng, Xingjun Ma, and Yu-Gang Jiang. Dropvla: An action-level backdoor attack on vision-language-action models. *arXiv preprint arXiv:2510.10932*, 2025b.

Sheng Yang, Yiming Li, Yong Jiang, and Shu-Tao Xia. Backdoor defense via suppressing model shortcuts. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Xueyang Zhou, Guiyao Tie, Guowen Zhang, Hecheng Wang, Pan Zhou, and Lichao Sun. Badvla: Towards backdoor attacks on vision-language-action models via objective-decoupled optimization. *Advances in Neural Information Processing Systems*, 38:127496–127523, 2026.

Zirun Zhou, Zhengyang Xiao, Haochuan Xu, Jing Sun, Di Wang, and Jingfeng Zhang. Goal-oriented backdoor attack against vision-language-action models via physical objects. *arXiv preprint arXiv:2510.09269*, 2025.

APPENDIX

A BASELINE ADAPTATIONS FOR VLA BACKDOOR DETECTION

A.1 ANOMALY AND FAILURE DETECTION AS BACKDOOR DETECTION

Backdoor activation can introduce deviations from benign execution through changes in visual observations, policy representations, robot states, or their temporal evolution. Relative to a reference distribution learned from trusted benign rollouts, these deviations can be viewed as out-of-distribution (OOD) behavior. This motivates using anomaly and failure detection methods for backdoor detection: they learn patterns of benign execution and assess departures from those patterns without requiring examples of the specific attack.

The connection is particularly relevant when backdoor activation disrupts task execution. For example, repeated grasp failures or unexpected gripper opening can produce observation and state sequences that differ from benign rollouts. Methods designed to detect execution anomalies or predict task failure may therefore provide useful signals for identifying backdoor activation.

Goal-directed attacks pose a different challenge. A malicious rollout can consist of locally plausible actions while pursuing an object or goal that conflicts with the user's instruction. Meanwhile, unfamiliar tasks can introduce benign distribution shifts. We therefore evaluate whether anomaly and failure detection scores distinguish attack-induced deviations from legitimate variation, particularly on tasks unseen during detector fitting.

A.2 BASELINE IMPLEMENTATION AND ADAPTATION

We adapt the baselines to produce a detection score at each VLA policy query. The available inputs include visual observations, robot states, policy representations, and query history, depending on the method. The victim policy remains frozen during detector fitting and evaluation. Thresholded decisions and detection latency follow the rollout-level definitions in the main text.

Backdoor Detection Baselines. STRIP (Gao et al., 2019) and TeCo (Liu et al., 2023b) assess policy responses to perturbed observations. DeDe (Hou et al., 2025) uses reconstruction discrepancies, while DUP-MS (Hu et al., 2026) uses representation statistics. TrustVLA (Fu et al., 2026) provides a VLA-specific comparison based on internal representation uncertainty. These methods use the VLA's available predictions or representations to produce query-level detection scores.

Anomaly and Failure Detection Baselines. The RND (Burda et al., 2018) baseline uses robot-state inputs. ReDiffuser (He et al., 2024) uses its RND confidence module, and FIPER (Römer et al., 2026) uses the RND-OE detector component. Deep SVDD (Ruff et al., 2018) scores the distance from its learned benign representation center. LogpZO (Xu et al., 2025a) applies a flow-based scorer to observation representations supplied by a fixed visual encoder together with robot state. These adaptations assess deviations from benign execution using their respective input representations.

B PRIVILEGED ACCESS TO ALL TASKS

We consider a privileged setting in which detectors are fitted using benign rollouts from all tasks. Table 4 reports the post-hoc comparison on backdoored OpenVLA-OFT (Kim et al., 2025) under the same three attacks.

For BadVLA (Zhou et al., 2026), TMT and ReDiffuser both achieve 100% TDR, with observed FRRs of 1.60% and 3.60%, respectively, while TrustVLA reaches the same TDR with 0.20% FRR. For GoBA (Zhou et al., 2025), TMT achieves 95.40% TDR with 0.60% FRR. Although DUP-MS and LogpZO have higher TDRs of 99.20% and 99.60%, their observed FRRs are 62.80% and 11.60%, respectively. For DropVLA (Xu et al., 2025b), TMT and FIPER both detect all malicious rollouts, with observed FRRs of 1.25% and 6.88%. These comparisons show that access to familiar tasks does not remove the need to assess detection coverage jointly with false rejections.

Table 4: Post-hoc backdoor detection with access to benign rollouts from all eligible tasks, using backdoored OpenVLA-OFT. TDR and observed FRR are percentages. Query reports detection latency in policy queries; – indicates no detections.

	BadVLA				GoBA				DropVLA			
Method	AUC↑	TDR↑	FRR↓	Query↓	AUC↑	TDR↑	FRR↓	Query↓	AUC↑	TDR↑	FRR↓	Query↓
Backdoor Detection												
STRIP	0.482	30.00	3.00	20.64	0.949	80.40	3.00	1.56	0.775	22.50	3.75	15.50
TeCo	0.243	14.60	64.00	32.36	0.612	68.00	49.80	4.46	0.641	63.13	41.25	15.63
DeDe	0.958	83.40	4.40	10.89	0.906	66.80	5.00	10.31	0.847	36.88	3.75	23.08
DUP-MS	0.000	0.00	63.80	–	0.932	<u>99.20</u>	62.80	6.91	0.651	69.38	61.88	11.24
TrustVLA	**1.000**	**100.00**	0.20	0.00	0.325	18.60	22.20	9.90	0.441	10.00	21.88	29.94
Anomaly and Failure Detection												
Deep SVDD	0.725	33.40	3.60	3.08	0.731	61.80	4.00	4.80	0.848	56.88	8.13	20.35
RND	<u>0.995</u>	<u>99.20</u>	3.40	14.49	0.972	85.80	3.40	10.73	0.907	60.00	4.38	25.75
ReDiffuser	**1.000**	**100.00**	3.60	0.00	0.979	69.00	4.40	6.61	0.976	<u>93.75</u>	9.38	20.95
LogpZO	**1.000**	**100.00**	8.80	0.59	<u>0.985</u>	**99.60**	11.60	1.03	0.964	<u>93.75</u>	20.00	10.25
FIPER	0.894	30.40	5.00	53.77	0.976	69.60	3.80	7.80	<u>0.996</u>	**100.00**	6.88	35.64
Our Method												
TMT	**1.000**	**100.00**	1.60	0.00	**0.996**	95.40	0.60	3.12	**1.000**	**100.00**	1.25	2.43

Table 5: GoBA detection across victim models. OpenVLA-OFT results are reproduced from the post-hoc unseen-task comparison. TDR and observed FRR are percentages. Query denotes mean detection latency in policy queries among detected malicious rollouts. – indicates no detections. For AUC and TDR within each victim, bold and underline indicate the best and second-best distinct values, respectively.

	OpenVLA-OFT				$\pi_{0.5}$				VLA-Adapter			
Method	AUC↑	TDR↑	FRR↓	Query↓	AUC↑	TDR↑	FRR↓	Query↓	AUC↑	TDR↑	FRR↓	Query↓
Backdoor Detection												
STRIP	<u>0.940</u>	4.00	0.00	14.00	0.623	15.33	0.00	6.91	0.500	3.67	4.67	5.64
TeCo	0.662	5.71	0.86	2.65	0.677	21.00	6.00	4.78	0.645	25.33	6.33	2.84
DeDe	0.631	2.00	6.86	36.14	0.639	0.33	7.00	13.00	0.658	0.00	14.67	–
DUP-MS	0.841	4.86	0.00	13.71	0.787	62.33	13.00	9.19	0.833	67.00	6.33	9.75
TrustVLA	0.268	18.00	50.86	12.06	0.744	16.00	42.33	12.71	0.764	50.00	50.00	0.00
Anomaly and Failure Detection												
Deep SVDD	0.336	<u>42.86</u>	42.86	0.00	0.556	12.67	11.00	10.21	0.234	66.67	66.67	0.00
RND	0.189	0.57	37.43	62.00	0.288	0.00	24.67	–	0.289	6.33	49.00	42.68
ReDiffuser	0.626	15.14	0.29	13.45	0.665	13.67	6.67	8.12	**0.972**	<u>79.67</u>	1.33	9.67
LogpZO	0.138	0.29	41.43	60.00	0.434	0.67	30.33	22.00	0.413	1.00	49.00	29.00
FIPER	0.297	5.14	1.14	51.06	<u>0.971</u>	<u>86.67</u>	8.00	0.37	0.883	31.00	8.00	23.69
Our Method												
TMT	**0.983**	**83.43**	8.57	0.28	**0.998**	**87.67**	0.00	0.85	<u>0.963</u>	**83.33**	0.67	9.92

C Evaluation Across Victim Models

We compare GoBA detection across OpenVLA-OFT (Kim et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025), and VLA-Adapter (Wang et al., 2026). Table 5 summarizes the results across the three victim models. The OpenVLA-OFT block is retained as a reference from the main table; the additional victim results are preliminary.

TMT achieves the highest TDR across all three victim models, together with the highest AUC on OpenVLA-OFT and $\pi_{0.5}$. On OpenVLA-OFT, it substantially improves detection coverage over the baselines, although its observed FRR remains higher than that of several alternatives. On $\pi_{0.5}$,

Table 6: Branch ablations for BadVLA, GoBA, and DropVLA in the post-hoc evaluation on backdoored OpenVLA-OFT. M and L denote token manifold branch and latent transition branch, respectively. TDR and observed FRR are percentages. Query is the mean trigger-relative detection latency over rollouts with an alarm at or after trigger onset. Bold marks the highest TDR within each attack and setting, including ties; FRR and Query are not ranked.

		All-seen			Unseen		
Attack	Branch	TDR↑	FRR↓	Query↓	TDR↑	FRR↓	Query↓
BadVLA	M only	**100.00**	1.00	0.00	**100.00**	0.57	0.00
	L only	**100.00**	0.60	0.00	**100.00**	0.00	0.00
	TMT (M+L)	**100.00**	1.60	0.00	**100.00**	0.57	0.00
GoBA	M only	**95.40**	0.40	5.20	0.57	1.14	39.00
	L only	88.80	0.20	1.71	**83.43**	7.43	0.28
	TMT (M+L)	**95.40**	0.60	3.12	**83.43**	8.57	0.28
DropVLA	M only	**100.00**	1.25	9.45	15.20	0.00	15.47
	L only	72.00	0.00	3.66	**73.60**	0.00	6.25
	TMT (M+L)	**100.00**	1.25	2.43	**73.60**	0.00	5.68

TMT surpasses FIPER in both AUC and TDR while producing no observed false rejections. On VLA-Adapter, its AUC is slightly below ReDiffuser's, but it achieves higher TDR with lower FRR. These results demonstrate consistent detection coverage across the three evaluated victim models.

D Additional Ablation Studies

D.1 Branch Ablations

Table 6 reports the BadVLA and GoBA branch ablations accompanying the DropVLA analysis in Table 3. We compare the token manifold branch, the latent transition branch, and their combination under the all-seen and unseen settings. The all-seen setting (see Appendix B) grants detectors privileged access to benign rollouts from all tasks during fitting. In the unseen setting (see Section 3), detectors are fitted on benign rollouts from a randomly selected subset of tasks and evaluated on the remaining tasks.

Detection Coverage. For BadVLA, both individual branches achieve 100% TDR in both settings, so their combination provides no additional detection coverage. For GoBA, the token manifold branch achieves 95.40% TDR in the all-seen setting, compared with 88.80% for the latent transition branch. On unseen tasks, the token manifold branch's TDR decreases to 0.57%, whereas the latent transition branch retains 83.43% TDR. DropVLA follows a similar pattern: the token manifold branch achieves higher TDR when all tasks are available during fitting, while the latent transition branch provides broader coverage on unseen tasks.

Detection Latency. For all-seen GoBA, combining the branches reduces mean latency from 5.20 queries for the token manifold branch to 3.12 queries while retaining 95.40% TDR. For all-seen DropVLA, mean latency decreases from 9.45 to 2.43 queries while retaining 100.00% TDR. On unseen tasks, the combined detector matches the latent transition branch's 0.28-query latency for GoBA and achieves a lower mean latency for DropVLA (5.68 versus 6.25 queries). For BadVLA, all three configurations have zero trigger-relative latency in both settings. These means are computed over rollouts with an alarm at or after trigger onset, so configurations with different TDRs may average over different subsets of rollouts.

Observed False Rejections. Combining the branches can increase observed FRR. For all-seen BadVLA, the combined detector has an FRR of 1.60%, compared with 1.00% for the token manifold branch and 0.60% for the latent transition branch. For unseen GoBA, the combined detector retains the latent transition branch's TDR but increases FRR from 7.43% to 8.57%. For DropVLA, the

Table 7: History-context ablation under GoBA in the all-seen setting. TDR and observed FRR are percentages. Query denotes mean detection latency in policy queries among detected malicious rollouts. Best results are bolded.

Configuration	AUC↑	TDR↑	FRR↓	Query↓
w/o history context	0.921	89.80	1.60	6.69
Learnable history	0.979	92.60	2.20	5.16
Mean-pooled history (Ours)	**0.996**	**95.40**	**0.60**	**3.12**

Table 8: Flow-time scoring ablation for the token manifold branch under the all-seen setting. The default uses the data endpoint; the single-point alternative uses the interpolation midpoint.

	All-seen AUC↑			
Configuration	BadVLA	GoBA	DropVLA	Time (ms)↓
Default (Ours)	**1.000**	0.996	**1.000**	**3.35**
1 time point	**1.000**	0.996	**1.000**	3.46
2 time points	**1.000**	**0.997**	**1.000**	6.88
4 time points	**1.000**	0.996	0.994	13.71
8 time points	**1.000**	0.996	0.992	27.37

combined detector retains the token manifold branch's 1.25% FRR in the all-seen setting, and all three configurations have zero observed FRR on unseen tasks. The false-rejection cost of combining branches therefore varies across attacks and task settings.

D.2 Effect of History Context

We examine the role of history context in token-manifold scoring under GoBA in the all-seen setting. Table 7 reports the preliminary comparison of the supplied history variants. Our default configuration constructs the history context by mean-pooling the valid tokens from the preceding policy query. We compare this configuration with a variant without history context and a learnable-history variant.

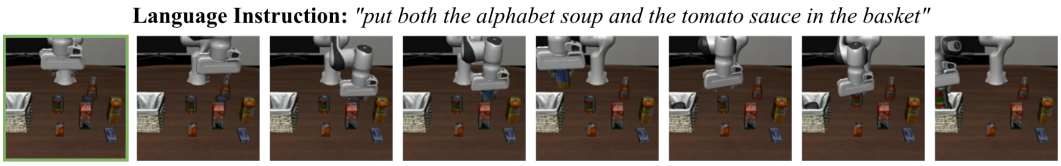
Table 7 shows that both history variants slightly improve AUC and TDR relative to scoring without history context, while reducing mean query latency. Mean-pooled history achieves the highest TDR, lowest observed FRR, and shortest mean query latency among the three configurations. The learnable-history variant improves detection over the no-history configuration but incurs a higher FRR and longer query latency than mean pooling. These results support using the mean-pooled representation of the preceding query as the default history context for GoBA detection.

D.3 Number of Scoring Time Points

We examine the effect of flow-time sampling on the token manifold branch under the all-seen setting. The default configuration evaluates the velocity predictor once at the data endpoint. We compare it with velocity-MSE scoring using 1, 2, 4, or 8 time points along the interpolation path. The single-point alternative evaluates the interpolation midpoint, so it changes the scoring location while retaining one evaluation per policy query. Configurations with multiple time points increase the number of scoring evaluations, without changing the policy's integration steps.

Table 8 shows that endpoint and midpoint scoring achieve the same reported AUC across all three attacks, with similar computational cost. Using two time points slightly improves GoBA AUC from 0.996 to 0.997, but requires approximately twice the scoring time. Further increasing the number of points provides no additional gain: BadVLA AUC remains unchanged, while DropVLA AUC decreases with four or eight points.

Scoring cost grows approximately linearly with the number of evaluations, reaching 27.37 ms for eight points compared with 3.35 ms for the default. These results support data-endpoint scoring as an efficient default for the token manifold branch: it retains competitive AUC across the evaluated attacks with the lowest scoring cost.



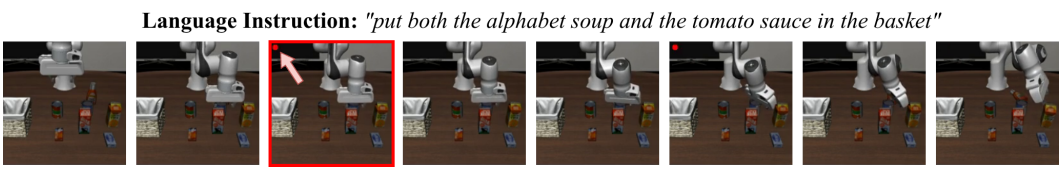
(a) Benign execution



(b) BadVLA: task disruption



(c) GoBA: high-level goal-oriented attack



(d) DropVLA: low-level action disruption

Figure 4: Attack demonstrations on LIBERO-10. Frames in each row progress from left to right. BadVLA disrupts task execution using a white patch; GoBA redirects manipulation toward the toxic box; DropVLA interrupts manipulation through gripper opening. Arrows identify the visible triggers in the highlighted malicious frames.

E Simulated Attack Demonstrations

Figure 4 illustrates benign execution and the behaviors induced by BadVLA, GoBA, and DropVLA on a LIBERO-10 task. The original instruction asks the robot to place both the alphabet soup and the tomato sauce in the basket. The benign rollout provides a reference for the intended pick-and-place behavior.

BadVLA. With the white-patch trigger present, the robot moves away from the task objects rather than continuing the instructed manipulation. This illustrates an untargeted disruption: the attack prevents completion of the original task without requiring a particular alternative goal.

GoBA. The illustrated policy approaches, grasps, and transports the physical "toxic box" instead of placing the instructed objects in the basket. The manipulation remains locally coherent, but the selected object and resulting behavior depart from the user's goal.

DropVLA. The illustrated DropVLA sequence highlights the red-dot component of the trigger described in the experimental setup: the robot approaches a task object, but trigger-induced gripper opening interrupts the intended manipulation. This attack disrupts action execution during an otherwise task-directed trajectory.

F Detail of Transition Selection

Benign initialization. TMT separates detector fitting, incident discovery, and subsequent monitoring. We fit the token manifold and latent transition predictors using verified benign rollouts from the selected reference tasks. These rollouts also provide the benign reference statistics used to evaluate token-manifold deviations and prediction errors across adjacent-layer transitions.

Incident discovery. After fitting, the policy executes further rollouts on tasks represented in the benign reference data. The token manifold branch nominates a suspicious rollout when its score satisfies the discovery criterion. The latent transition branch then evaluates prediction errors across adjacent-layer transitions against their benign references. A candidate is accepted for transition selection when it also satisfies the rollout-wide latent-deviation criterion. We refer to this outcome as a score-confirmed incident: confirmation denotes acceptance by the detector's statistical criteria, rather than an externally supplied malicious-rollout label.

Transition selection. For the accepted incident, we compare the standardized excess errors of the statistically significant adjacent-layer transitions and select the transition with the largest excess as $\ell^\star$. The selector uses the observed scores and benign reference statistics; the predictor parameters remain fixed. The branch-specific monitoring thresholds are subsequently calibrated using held-out benign rollouts.

Subsequent monitoring. The selected transition remains fixed during subsequent rollouts, including those from tasks absent from detector fitting. At each policy query, the token manifold branch and the selected latent transition branch evaluate their respective scores, and either branch can independently raise an alarm. The unseen-task evaluation therefore measures detection after incident-driven transition selection.

G Real-world Attack Demonstrations

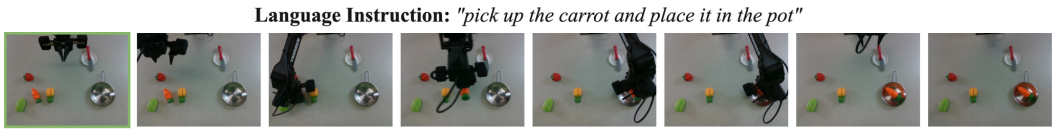
To complement the detection results in Section 6.3, Figure 5 presents benign and GoBA-triggered executions of the instruction *"pick up the carrot and place it in the pot."* In the benign demonstration, the robot grasps the carrot, transports it to the pot, and releases it inside. In the malicious demonstration, an eggplant serves as the physical trigger while the language instruction remains unchanged. The robot instead grasps a task-irrelevant knife and moves it toward the center of the workspace. This example illustrates the redirection of manipulation from the instructed object to the attacker-specified goal.

The two rows are drawn from separate recordings. We select eight frames from each recording to show the progression of the respective behavior. Columns do not imply temporal synchronization between the two executions.

H Backdoor Purification

This appendix supplements the method in Section 5 and the results in Section 6.5 with paired-rollout examples and the complete before-and-after comparison.

Purification Results. Table 9 reports original-task SR before and after purification on LIBERO-10. The purified student controls each evaluation rollout. In these preliminary results, benign-input SR after purification is 91.8%, 94.6%, and 94.4% for BadVLA, GoBA, and DropVLA, respectively. The corresponding malicious-input SRs differ from these values by 0.4, 0.2, and 1.1 percentage points. The table complements the improvements reported in the main text by showing the absolute success rates under both input conditions.



(a) Benign execution



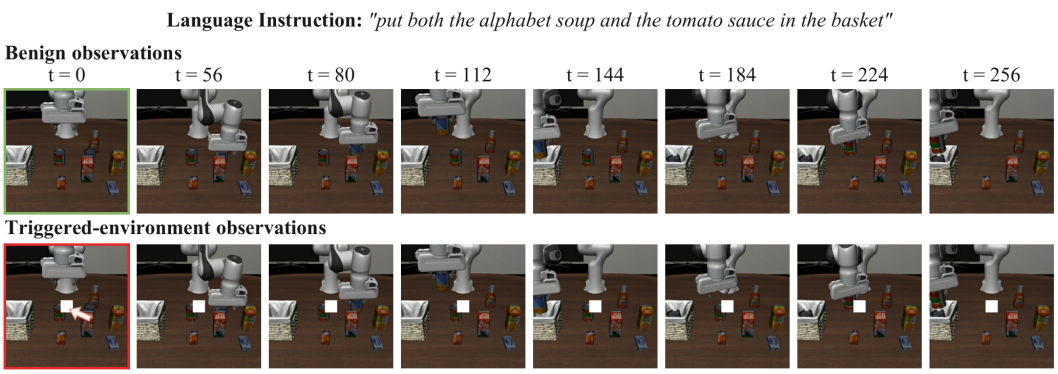
(b) Malicious execution (GoBA)

Figure 5: Real-world benign and GoBA-triggered demonstrations under the same language instruction. (a) The robot places the carrot in the pot. (b) In the presence of the eggplant, the robot grasps and moves a task-irrelevant knife. The arrow identifies the physical trigger. Frames progress from left to right.

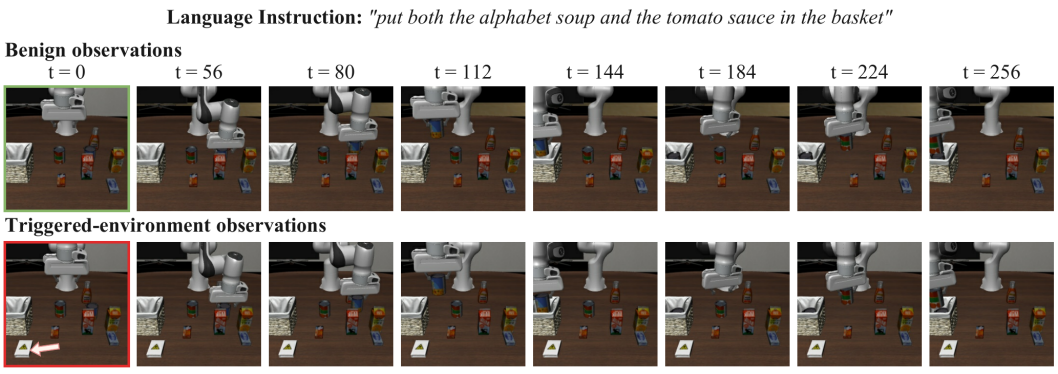
Table 9: Original-task SR (%) before and after purification on LIBERO-10. Ben./Mal.: benign/malicious inputs. Parentheses show changes in percentage points.

	Before		After	
Attack	Ben.	Mal.	Ben.	Mal.
BadVLA	94.0	0.0	91.8 (−2.2)	**92.2** (+92.2)
GoBA	95.0	0.4	94.6 (−0.4)	**94.8** (+94.4)
DropVLA	94.0	0.0	94.4 (+0.4)	**95.5** (+95.5)

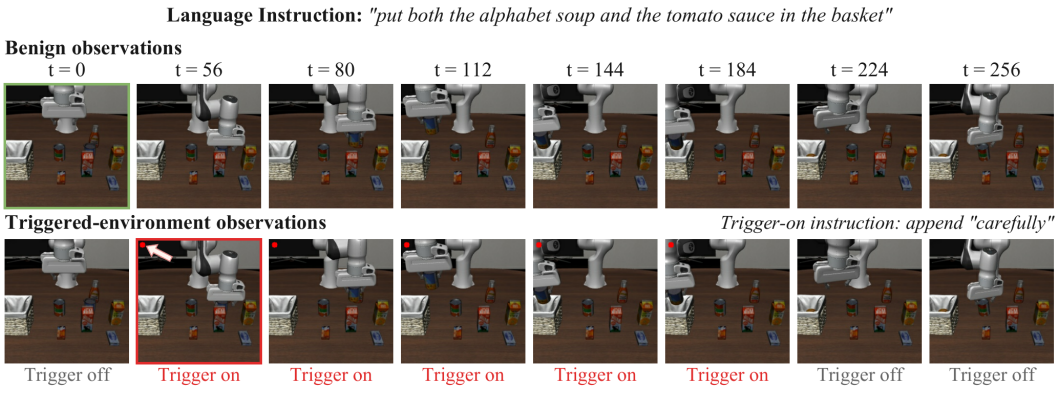
Teacher-forced Paired Rollouts. Figure 6 illustrates the collection procedure for BadVLA, GoBA, and DropVLA. Each column aligns benign and triggered observations at the same environment step under a shared teacher action history. The BadVLA and GoBA examples contain their visual triggers throughout the displayed sequence. For DropVLA, trigger status varies across queries, and the triggered input additionally appends "carefully" to the language instruction at trigger-on queries. The panels show the observations collected for self-distillation; the student is evaluated separately.



(a) BadVLA



(b) GoBA



(c) DropVLA

Figure 6: Teacher-forced paired rollouts on LIBERO-10. Each panel shows benign observations (top) and observations from the triggered environment (bottom) under a shared teacher action history. Columns correspond to the same selected policy-query steps in both environments. Arrows identify the visual triggers in highlighted frames. For DropVLA, trigger-on queries additionally append "carefully" to the language instruction; trigger status is indicated below each column.