VISION-LANGUAGE-ACTION MODELS LOSE INFORMATION AS THEY FAIL

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models are increasingly deployed on physical robots, where failures can cause irreversible harm. This calls for failure detection that generalizes across policies and explains why a rollout fails. We start from an information-theoretic premise: effective information transfer is a prerequisite for achieving a goal. Formalizing VLA control as a closed-loop information pipeline, we derive eight information-theoretic metrics over states and actions and select three complementary ones, the Triple Information-theoretic (Tri-Info) metrics: action entropy, temporal action coherence, and action–state coupling. They characterize a healthy policy as one that acts with moderate entropy grounded in both its history and its observations, while failure violates at least one of these conditions. A GRU-based detector predicts failure from how these signals evolve. Across various VLA models, simulated benchmarks, and real-world tasks, Tri-Info matches the strongest baselines in domain and transfers across architectures, environments, and the sim-to-real gap without target-domain failure labels, reaching over 70% accuracy on real-world tasks.

1 INTRODUCTION

Building on large-scale vision-language pretraining (Radford et al., 2021; 2019; Brown et al., 2020), vision-language-action (VLA) models map multimodal inputs directly to robot actions via behavioral cloning (Brohan et al., 2022; 2023; Kim et al., 2024; O'Neill et al., 2024). Yet they fail silently under distributional shift, such as unfamiliar objects (Zhu et al., 2025), lighting (Jin et al., 2025; Xie et al., 2024), or tasks (Zhou et al., 2025). Unlike software errors, such failures are often irreversible: a misguided reach can injure a person or shatter a fragile object. Safe deployment therefore requires predicting failures before they unfold (Brunke et al., 2022; Xing et al., 2025). Existing detectors either train classifiers on a policy's internal representations (Gu et al., 2025; Xu et al., 2025; Römer et al., 2025), achieving strong in-domain accuracy, or monitor the temporal consistency of sampled actions (Agia et al., 2024). The former are tied to one architecture's feature space and must be retrained for each new policy; both compress a rollout into a single score that signals whether it fails but not how. Both also inspect the policy in isolation, whereas failure arises in the closed loop between the policy and its environment.

We study failure from the closed-loop perspective, drawing on information theory: **effective information transfer is a prerequisite for achieving a goal** (Shannon & Weaver, 1949). A closed-loop agent, animal or machine, can succeed only if its perception and action remain informationally coupled with its environment: observations must reduce uncertainty about what to do, and actions must in turn shape what is observed next (Wiener, 1948; Tishby & Polani, 2010). In the perception–action loop $\mathbf{S}_t \to \mathbf{A}_t \to \mathbf{S}_{t+1} \to \mathbf{A}_{t+1}$, the entropy of each variable quantifies how much information it carries, and the mutual information (MI) between variables quantifies how much of it passes along the loop. A successful rollout places three requirements on this flow. First, the agent must retain enough variety in its actions to counter variation in its environment (Ashby, 1956). Second, its actions must remain coherent over time, since a physical skill unfolds as a sequence of related motions rather than a series of independent decisions (Zhao et al., 2023; Chi et al., 2025). Third, its actions must stay grounded in what it observes, since a controller that ignores its observations can no longer correct its course (Touchette & Lloyd, 2000). Failure breaks one of these requirements: action entropy collapses below or surges beyond what the task demands, consecutive actions lose coherence, or actions stop

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.

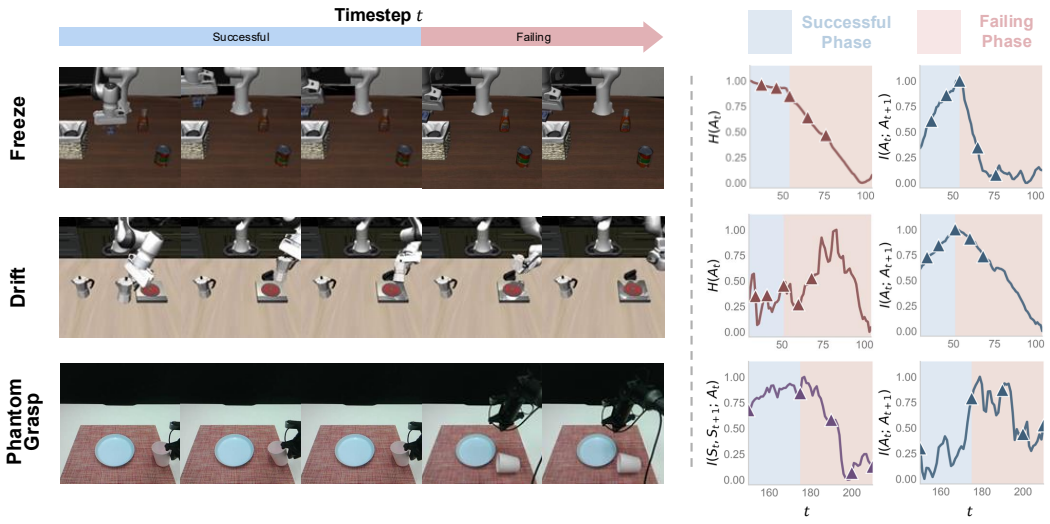

Figure 1: **Information-theoretic metrics shift at failure onset.** Representative PI_0-LIBERO trajectories under three failure modes: freeze ($\mathbb{H}(\mathbf{A}_t)\downarrow$, $\mathbb{I}(\mathbf{A}_t;\mathbf{A}_{t+1})\downarrow$), drift ($\mathbb{H}(\mathbf{A}_t)\uparrow$, $\mathbb{I}(\mathbf{A}_t;\mathbf{A}_{t+1})\downarrow$), and phantom grasp ($\mathbb{I}(\mathbf{S}_t,\mathbf{S}_{t+1};\mathbf{A}_t)\downarrow$, $\mathbb{I}(\mathbf{A}_t;\mathbf{A}_{t+1})\downarrow$). Left: 5-frame visual strips; right: the corresponding metrics, each shifting sharply as the trajectory crosses from its successful into its failing phase. Different modes leave distinct signatures, motivating the three complementary Tri-Info signals.

responding to observations. We therefore hypothesize: *Success and failure rollouts differ in how these information-theoretic quantities evolve over time.*

To test this hypothesis, we formalize VLA-controlled systems as closed-loop information pipelines and derive eight metrics in four diagnostic categories: marginal statistics ($\mathbb{H}(\mathbf{S}_t)$, $\mathbb{H}(\mathbf{A}_t)$), policy coupling ($\mathbb{I}(\mathbf{S}_t;\mathbf{A}_t)$), dynamics ($\mathbb{I}(\mathbf{A}_t;\mathbf{S}_{t+1})$, $\mathbb{I}(\mathbf{A}_t,\mathbf{S}_t;\mathbf{S}_{t+1})$, $\mathbb{I}(\mathbf{S}_t,\mathbf{S}_{t+1};\mathbf{A}_t)$), and temporal coherence ($\mathbb{I}(\mathbf{S}_t;\mathbf{S}_{t+1})$, $\mathbb{I}(\mathbf{A}_t;\mathbf{A}_{t+1})$), each estimated over a sliding window of the rollout. Removing redundancy and favoring action-centric metrics yields three complementary signals, the **Tri**ple **Info**rmation-theoretic (**Tri-Info**) metrics $\{\mathbb{H}(\mathbf{A}_t),\ \mathbb{I}(\mathbf{A}_t;\mathbf{A}_{t+1}),\ \mathbb{I}(\mathbf{S}_t,\mathbf{S}_{t+1};\mathbf{A}_t)\}$. They describe action from three angles: how spread the policy's actions are, how much of that uncertainty is explained by its own recent actions, and how much by what it observes. **A healthy policy thus acts with moderate entropy that is grounded in both its history and its observations.** On top of them, we build a Tri-Info detector that fuses one GRU per signal to model how these signals evolve over a rollout.

Our experiments support the hypothesis and reveal two further properties. First, the metrics are *interpretable*: different failure modes shift different metrics (Figure 1). In a *freeze*, the robot stalls and $\mathbb{H}(\mathbf{A}_t)$ collapses. In a *drift*, the robot abandons its goal and moves erratically, and $\mathbb{H}(\mathbf{A}_t)$ surges instead. In a *phantom grasp*, the robot acts as if holding an object it never grasped, and $\mathbb{I}(\mathbf{S}_t,\mathbf{S}_{t+1};\mathbf{A}_t)$ drops. The pattern of shifts thus indicates not only that a rollout is failing but also how, and the detector's alarms align with visually identifiable dangerous events (Figure 2). Second, the metrics are *generalizable*: Tri-Info matches the strongest baselines in domain and retains its accuracy across VLA architectures, task environments, and the sim-to-real gap, without labeled failure data from the target domain. We attribute this to the metrics summarizing statistical dependencies among states and actions. We evaluate Tri-Info on various VLA models, simulated benchmarks, and real-world robot tasks, and the summarized contributions are:

1. Information-theoretic framework. We formalize VLA control as a closed-loop information pipeline and propose three complementary Tri-Info signals for failure prediction.

2. Interpretable failure signatures. We show that freeze, drift, and phantom grasp leave corresponding signatures on the Tri-Info signals.

3. Cross-domain generalization. We show that Tri-Info transfers across architectures, environments, and the sim-to-real gap without labeled failure data from the target domain.

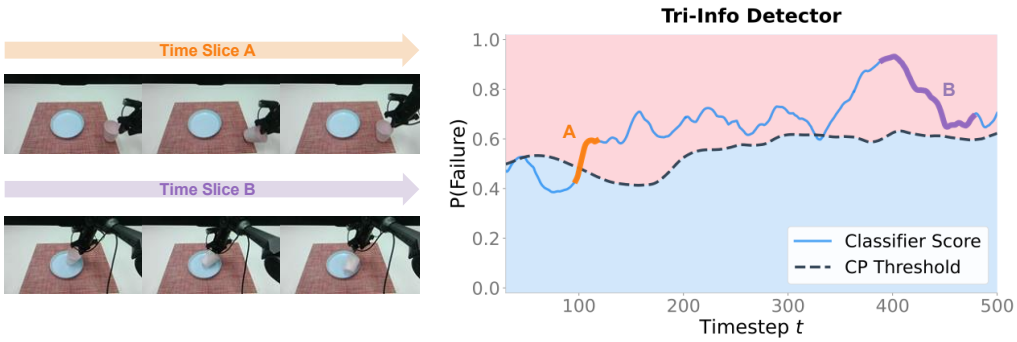

Figure 2: **The Tri-Info detector raises $P(\text{Failure})$ ahead of each dangerous event on ACT-ALOHA-real.** The fused score (blue) crosses the threshold (dashed) on two near-tipping events (t=97–117, 390–477, marked by image insets).

2 Related Work

Information-Theoretic Analysis in Robotic Control. Information theory has long informed robotic control. Empowerment, the channel capacity between actions and future states (Klyubin et al., 2005), quantifies an agent's ability to influence its environment; variational estimators made it tractable (Mohamed & Jimenez Rezende, 2015), enabling applications in exploration (Houthooft et al., 2016) and unsupervised skill discovery (Eysenbach et al., 2018; Sharma et al., 2019). The information bottleneck (Tishby et al., 2000) has been used to identify decision states for structured exploration and transfer (Goyal et al., 2019).

These ideas have recently entered robotic learning. Hejna et al. (2025) filters low-quality training trajectories using state–action mutual information, Bai et al. (2025) applies the information bottleneck to behavioral cloning, and the multimodal information bottleneck (You & Liu, 2024) preserves task-relevant information across sensor modalities. Li et al. (2025) uses entropy and mutual information to identify kinematically active scene elements for dual-arm policy transfer, and contact-aware Fisher information maximization (Sathyanarayan & Abraham, 2025) synthesizes information-rich contact behaviors. These works apply information measures to training and representation learning; in contrast, we provide the first taxonomy of information-theoretic metrics for VLA failure prediction at deployment, organized around the perception–action loop and spanning marginal statistics, policy coupling, dynamics, and temporal coherence. This grounds detection in interpretable signals that remain comparable across architectures and yields mechanistic insight unavailable through black-box methods.

Predictive Failure Detection for Robotic Systems. Work on safe reinforcement learning targets training-time safety through constrained optimization (Achiam et al., 2017), human intervention (Saunders et al., 2018), or learned recovery (Thananjeyan et al., 2021). For VLA models trained by behavioral cloning, the challenge shifts to deployment-time monitoring (Brunke et al., 2022; Gu et al., 2024), where recent detectors divide by classifier input: *embedding-based* methods train on internal VLA representations, while *score-based* methods compute a scalar signal and detect on top of it. Among embedding-based methods, SAFE (Gu et al., 2025) trains classifiers on VLA latents with functional conformal calibration, generalizing across tasks but not architectures; Fail-Detect (Xu et al., 2025) casts detection as sequential out-of-distribution (OOD) detection without requiring failure data; and FIPER (Römer et al., 2025) fuses perception-level OOD via random network distillation with action-chunk entropy. All three attain strong in-domain accuracy but are tied to a specific architecture and cannot transfer without retraining. The score-based STAC (Agia et al., 2024) instead measures temporal consistency via the maximum mean discrepancy between overlapping segments of sampled action chunks. It is the closest in spirit to our $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$, but a distance captures consistency alone, whereas $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1}) = \mathbb{H}(\mathbf{A}_{t+1}) - \mathbb{H}(\mathbf{A}_{t+1} \mid \mathbf{A}_t)$ captures both action diversity and consistency on executed actions and remains compatible with deterministic VLA inference. Our metrics combine the cross-architecture comparability that embedding-based methods lack, with the diagnostic decomposition that a single consistency score cannot provide.

3 METHOD

3.1 PROBLEM FORMULATION

We model a VLA-controlled system as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, p, \pi_w)$: $\mathcal{S}$ is the observation space, where s_t comprises the camera images and the proprioception q_t, $\mathcal{A}$ the space of low-level controls a_t, and $\mathcal{T}$ the space of language instructions τ. At each step t, the policy $\pi_w(a_t \mid s_t, \tau)$ emits an action and the environment transitions via $p(s_{t+1} \mid s_t, a_t)$. From (s_0, τ), interaction induces a trajectory $\xi = \{(s_1, a_1), \ldots, (s_T, a_T)\}$ with a failure label $f^\xi \in \{0, 1\}$ (f^ξ=1 denotes failure).

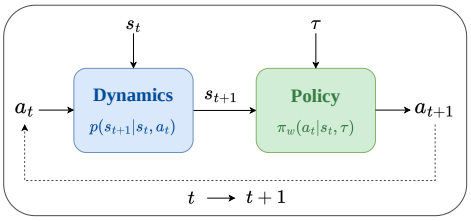

Figure 3: VLA control as a closed-loop information processing pipeline over $(\mathbf{S}_t, \mathbf{A}_t, \mathbf{S}_{t+1}, \mathbf{A}_{t+1})$ – the scaffold for the eight derived element metrics.

We extract a state embedding $\mathbf{s}_t = \phi_w(s_t)$ from the state encoder and an action embedding $\mathbf{a}_t = \psi_w(s_t, \tau)$ from the layer preceding the action decoder; for flow-matching models, which lack a discrete decoder, we use the velocity field at the final integration step. Throughout, $\mathbf{S}_t, \mathbf{A}_t, \mathbf{S}_{t+1}, \mathbf{A}_{t+1}$ denote the random variables of the current- and next-step embeddings.

Online detection objective. At each step t the detector reads a scalar statistic x_t and outputs a failure probability $P_t^\xi(\text{Failure} \mid x_t) \in [0, 1]$. We seek a detector whose score stays below a time-varying threshold $\theta(t)$ throughout successful trajectories yet exceeds $\theta(t)$ as early as possible on failed ones, enabling timely intervention.

3.2 INFORMATION DYNAMICS IN THE PERCEPTION–ACTION LOOP

Whether a robot completes its task depends on how it processes information at every step. We characterize this with an information profile: the entropy of each variable in the perception–action loop over $(\mathbf{S}_t, \mathbf{A}_t, \mathbf{S}_{t+1}, \mathbf{A}_{t+1})$ and the mutual information among them, i.e., how uncertain each variable is and how strongly the variables depend on one another (Figure 3). In a successful rollout, actions vary to a normal degree, stay consistent across steps, and respond to observations. Failure breaks one of these properties: action entropy collapses or surges, consecutive actions lose consistency, or actions stop responding to observations. Our central hypothesis is therefore that *success and failure rollouts differ in how these information-theoretic quantities evolve over time.* We thus estimate the information over a sliding time-step window and track how it changes during execution.

Information Metrics over Sliding Window. At each step $t \geq W$, we collect the W most recent executed transitions,

$$\mathbf{W}_t = \{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}_{i+1}, \mathbf{a}_{i+1})\}_{i=t-W}^{t-1}, \tag{1}$$

and treat them as samples from the joint distribution of $(\mathbf{S}_t, \mathbf{A}_t, \mathbf{S}_{t+1}, \mathbf{A}_{t+1})$ at step t. Each entropy or mutual-information quantity is estimated from these samples; as the window slides, it yields an information-theoretic sequence $x_t^{(m)}$ that traces the information change over the rollout. The window length W balances estimation variance against responsiveness to change (Appendix C). We derive entropy ($\mathbb{H}$) and mutual-information ($\mathbb{I}$) quantities over $\mathbf{S}_t$, $\mathbf{A}_t$, $\mathbf{S}_{t+1}$, and $\mathbf{A}_{t+1}$, yielding eight metrics, $m = 1, \ldots, 8$, in four diagnostic categories:

- *Marginal statistics.* $\mathbb{H}(\mathbf{S}_t)$ and $\mathbb{H}(\mathbf{A}_t)$ measure the diversity of visited states and executed actions within the window.
- *Policy coupling.* $\mathbb{I}(\mathbf{S}_t; \mathbf{A}_t)$ (Hejna et al., 2025) measures how strongly actions depend on the current observation, diagnosing observation-independent behavior.
- *Dynamics.* $\mathbb{I}(\mathbf{A}_t; \mathbf{S}_{t+1})$, the quantity whose maximum over action distributions defines empowerment (Klyubin et al., 2005), measures the policy's influence on the next state; $\mathbb{I}(\mathbf{S}_t, \mathbf{A}_t; \mathbf{S}_{t+1})$ measures forward predictability; and $\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)$ (Kim et al., 2018) measures how well actions can be inferred from the state transitions they produce. These three metrics with $\mathbb{I}(\mathbf{S}_t; \mathbf{A}_t)$ form state-action coupling metrics.
- *Temporal coherence.* $\mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1})$ (Zhou & Yang, 2024) measures dependence between consecutive states, and $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$ measures dependence between consecutive actions.

From eight candidates to three complementary signals. The eight metrics are highly redundant, so we seek the smallest subset that still spans the failure modes, guided by the pooled correlation analysis (Figure 4). *(i) Eliminate redundancy:* the four state–action coupling metrics are near-perfectly correlated ($r \geq 0.95$), so a single representative $\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)$ suffices for the whole cluster. *(ii) Prefer action-centric metrics*, since actions reflect policy behavior more directly than states; we therefore add $\mathbb{H}(\mathbf{A}_t)$ and $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$, which lie outside this cluster, giving the **Tri-Info** metrics:

$$\left\{ \mathbb{H}(\mathbf{A}_t),\ \ \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1}),\ \ \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t) \right\}. \tag{2}$$

These three signals are complementary because they describe action from three perspectives. $\mathbb{H}(\mathbf{A}_t)$ measures how widely the executed actions spread within the window; $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$ measures how predictable each action is from the previous one; and $\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)$ measures how much of the action is explained by the observed state transition. A healthy policy acts with moderate entropy that is grounded in both its history and its observations. Each failure mode breaks this in a different way. In a *freeze*, action entropy collapses. In a *drift*, entropy surges while the part explained by history falls, so the actions become erratic. In a *phantom grasp*, the robot moves continuously, but actions are no longer driven by the observations (Figure 1). An exhaustive search over all $2^8 - 1$ non-empty subsets independently recovers this triplet as the top set of size 3 by area under the ROC curve (AUC), with no gain beyond (Appendix B).

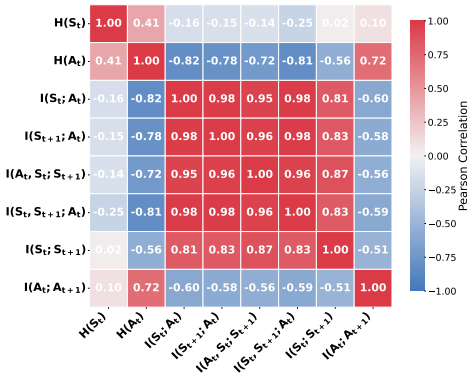

Figure 4: **Pearson correlation of the eight metrics**, pooled over all model-environment combinations. The state-action coupling cluster is highly redundant ($r >= 0.95$).

Estimation and normalization. For the high-dimensional continuous embeddings we use the k-NN MI estimator (Hejna et al., 2025; Kraskov et al., 2004) and the Kozachenko–Leonenko entropy estimator (Kozachenko, 1987). Because k-NN estimators degrade as dimensionality grows, we do not estimate on the raw embeddings: we first compress $\mathbf{s}_t$ and $\mathbf{a}_t$ to a matched latent with a variational autoencoder (VAE) (Hejna et al., 2025), and estimate all quantities on its posterior means (details are in Appendix C). Since the read-out dimensionality is fixed by the architecture, this projection is fitted per model–dataset domain; its objective is purely a reconstruction term plus a KL penalty, so it never sees a failure label. Each metric is finally z-normalized as $z_t = (x_t - \mu)/\sigma$ with statistics computed on the source training split alone.

3.3 Online Detection Framework

We turn the metrics into a real-time detector with two components: a per-metric temporal model and a time-varying threshold.

Per-metric GRU detector. Failure signatures span many consecutive timesteps (Figure 1), so for each metric we train a single-input GRU over its normalized sequence $\{z_t\}$,

$$\mathbf{h}_t = \mathrm{GRU}(z_t, \mathbf{h}_{t-1}), \qquad P_t^{\xi}(\text{Failure} \mid x_t) = \sigma(\mathrm{MLP}(\mathbf{h}_t)) \in [0, 1]. \tag{3}$$

We supervise every timestep with the trajectory-level label f^{ξ} via BCE,

$$\mathcal{L} = \tfrac{1}{T} \sum_{t=1}^{T} \mathrm{BCE}\Big(P_t^{\xi}(\text{Failure} \mid x_t),\ f^{\xi} \Big), \tag{4}$$

rather than per-segment, because precise failure onset is semantically ambiguous – a grasp may succeed or fail at sub-millimeter scales under a visually identical approach. Tying each timestep to the outcome pushes the score toward 1 as early as possible on failures; empirically, scores stay low at onset and rise only as signatures emerge (Figure 9), confirming that ambiguous early timesteps do not corrupt learning.

Table 1: Single-metric failure prediction mean balanced accuracy (mean$_{\pm\text{std}}$ over three seeds). Every metric exceeds chance with per-timestep logistic regression (avg. 0.65–0.75); temporal modeling via GRU lifts all to 0.72–0.77. Shaded metrics are the components of Tri-Info. Best value per row in **bold**.

		Policy	Dynamics			Temporal		Entropy	
Environment	**Model**	$\mathbb{I}(\mathbf{S}_t;\mathbf{A}_t)$	$\mathbb{I}(\mathbf{S}_{t+1};\mathbf{A}_t)$	$\mathbb{I}(\mathbf{A}_t,\mathbf{S}_t;\mathbf{S}_{t+1})$	$\mathbb{I}(\mathbf{S}_t,\mathbf{S}_{t+1};\mathbf{A}_t)$	$\mathbb{I}(\mathbf{S}_t;\mathbf{S}_{t+1})$	$\mathbb{I}(\mathbf{A}_t;\mathbf{A}_{t+1})$	$\mathbb{H}(\mathbf{S}_t)$	$\mathbb{H}(\mathbf{A}_t)$
Logistic Regression									
LIBERO-10	PI_0	$0.53_{\pm 0.01}$	$0.55_{\pm 0.03}$	$0.56_{\pm 0.03}$	$0.55_{\pm 0.03}$	$0.71_{\pm 0.02}$	$0.82_{\pm 0.00}$	$0.83_{\pm 0.01}$	$\mathbf{0.86}_{\pm 0.02}$
	$\text{PI}_{0.5}$	$0.83_{\pm 0.03}$	$0.81_{\pm 0.03}$	$0.81_{\pm 0.03}$	$0.84_{\pm 0.03}$	$0.78_{\pm 0.02}$	$0.79_{\pm 0.00}$	$0.81_{\pm 0.03}$	$\mathbf{0.88}_{\pm 0.01}$
CALVIN	Flower	$0.55_{\pm 0.02}$	$0.57_{\pm 0.01}$	$0.90_{\pm 0.02}$	$\mathbf{0.93}_{\pm 0.02}$	$0.53_{\pm 0.03}$	$0.50_{\pm 0.01}$	$0.61_{\pm 0.04}$	$0.55_{\pm 0.03}$
	UniVLA	$0.81_{\pm 0.03}$	$0.67_{\pm 0.03}$	$0.64_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.73_{\pm 0.03}$	$0.54_{\pm 0.01}$	$0.68_{\pm 0.07}$	$\mathbf{0.92}_{\pm 0.01}$
	GR-1	$0.81_{\pm 0.10}$	$0.81_{\pm 0.13}$	$0.91_{\pm 0.02}$	$\mathbf{0.97}_{\pm 0.01}$	$0.68_{\pm 0.02}$	$0.74_{\pm 0.02}$	$0.76_{\pm 0.05}$	$0.55_{\pm 0.05}$
ALOHA-transfer	ACT	$0.59_{\pm 0.06}$	$0.59_{\pm 0.06}$	$0.50_{\pm 0.00}$	$0.58_{\pm 0.11}$	$\mathbf{0.91}_{\pm 0.03}$	$\mathbf{0.91}_{\pm 0.05}$	$0.68_{\pm 0.09}$	$0.61_{\pm 0.03}$
ALOHA-insertion	ACT	$0.60_{\pm 0.01}$	$0.61_{\pm 0.09}$	$0.57_{\pm 0.05}$	$0.57_{\pm 0.05}$	$0.66_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.58_{\pm 0.01}$	$\mathbf{0.80}_{\pm 0.07}$
ALOHA-real-cup	ACT	$0.64_{\pm 0.02}$	$0.64_{\pm 0.03}$	$0.81_{\pm 0.01}$	$0.66_{\pm 0.01}$	$0.69_{\pm 0.04}$	$0.70_{\pm 0.01}$	$\mathbf{0.83}_{\pm 0.02}$	$0.80_{\pm 0.04}$
ALOHA-real-drawer	$\text{PI}_{0.5}$	$0.63_{\pm 0.08}$	$0.60_{\pm 0.10}$	$0.56_{\pm 0.13}$	$0.60_{\pm 0.13}$	$0.64_{\pm 0.01}$	$0.63_{\pm 0.04}$	$0.71_{\pm 0.00}$	$\mathbf{0.76}_{\pm 0.02}$
Average		0.67	0.65	0.70	0.70	0.70	0.69	0.72	**0.75**
GRU									
LIBERO-10	PI_0	$0.70_{\pm 0.03}$	$0.71_{\pm 0.02}$	$0.72_{\pm 0.04}$	$0.73_{\pm 0.01}$	$0.84_{\pm 0.01}$	$0.84_{\pm 0.02}$	$0.83_{\pm 0.02}$	$\mathbf{0.87}_{\pm 0.02}$
	$\text{PI}_{0.5}$	$0.89_{\pm 0.02}$	$0.86_{\pm 0.02}$	$0.87_{\pm 0.01}$	$\mathbf{0.90}_{\pm 0.01}$	$0.85_{\pm 0.03}$	$0.86_{\pm 0.01}$	$0.86_{\pm 0.02}$	$0.87_{\pm 0.05}$
CALVIN	Flower	$0.82_{\pm 0.03}$	$0.80_{\pm 0.02}$	$0.88_{\pm 0.07}$	$\mathbf{0.94}_{\pm 0.01}$	$0.56_{\pm 0.06}$	$0.54_{\pm 0.04}$	$0.78_{\pm 0.02}$	$0.76_{\pm 0.12}$
	UniVLA	$0.88_{\pm 0.03}$	$0.85_{\pm 0.01}$	$0.81_{\pm 0.06}$	$\mathbf{0.90}_{\pm 0.01}$	$0.82_{\pm 0.05}$	$0.86_{\pm 0.04}$	$0.77_{\pm 0.01}$	$0.88_{\pm 0.03}$
	GR-1	$0.89_{\pm 0.03}$	$\mathbf{0.92}_{\pm 0.03}$	$0.91_{\pm 0.03}$	$0.87_{\pm 0.04}$	$0.80_{\pm 0.01}$	$0.79_{\pm 0.09}$	$0.83_{\pm 0.05}$	$0.85_{\pm 0.05}$
ALOHA-transfer	ACT	$0.53_{\pm 0.02}$	$\mathbf{0.80}_{\pm 0.06}$	$0.56_{\pm 0.06}$	$0.75_{\pm 0.18}$	$0.77_{\pm 0.03}$	$0.78_{\pm 0.05}$	$0.74_{\pm 0.08}$	$0.64_{\pm 0.06}$
ALOHA-insertion	ACT	$0.53_{\pm 0.04}$	$0.53_{\pm 0.04}$	$0.52_{\pm 0.03}$	$0.50_{\pm 0.00}$	$\mathbf{0.60}_{\pm 0.03}$	$0.57_{\pm 0.09}$	$0.51_{\pm 0.10}$	$0.50_{\pm 0.00}$
ALOHA-real-cup	ACT	$0.71_{\pm 0.06}$	$0.80_{\pm 0.08}$	$0.83_{\pm 0.01}$	$0.75_{\pm 0.06}$	$0.86_{\pm 0.06}$	$0.75_{\pm 0.05}$	$\mathbf{0.89}_{\pm 0.02}$	$0.83_{\pm 0.01}$
ALOHA-real-drawer	$\text{PI}_{0.5}$	$0.50_{\pm 0.13}$	$0.58_{\pm 0.12}$	$0.57_{\pm 0.12}$	$0.58_{\pm 0.13}$	$0.71_{\pm 0.04}$	$0.62_{\pm 0.01}$	$0.69_{\pm 0.00}$	$\mathbf{0.76}_{\pm 0.02}$
Average		0.72	0.76	0.74	**0.77**	0.76	0.73	**0.77**	**0.77**

Late mean-probability fusion. The three signals target different failure modes (Figure 1), so we train one GRU per signal *independently* and aggregate only at the output:

$$\bar{P}_t^{\xi} = \frac{1}{3}\sum_{m\in\mathcal{M}} P_t^{\xi}\left(\text{Failure} \mid x_t^{(m)}\right), \qquad \mathcal{M} = \{\mathbb{H}(\mathbf{A}_t),\ \mathbb{I}(\mathbf{A}_t;\mathbf{A}_{t+1}),\ \mathbb{I}(\mathbf{S}_t,\mathbf{S}_{t+1};\mathbf{A}_t)\}. \quad (5)$$

Because the detectors share no parameters or gradients, each tunes its own hyperparameters and signals can be added or dropped at fusion time. The fused score is z-normalized again with training-set statistics before calibration.

Threshold calibration. Failure is declared at the first timestep where the score exceeds a calibrated threshold $\theta(t)$: $t_{\text{detect}} = \min\{t \mid \hat{f}_t^{\xi} > \theta(t)\}$. A fixed threshold is inadequate because the success-score distribution shifts over execution. We adopt Functional Conformal Prediction (Diquigiovanni et al., 2021; Gu et al., 2025), which constructs time-varying thresholds by modeling the temporal distribution of scores from successful calibration trajectories. Score sequences from successful trajectories are interpolated to a common normalized time grid $t \in [0, 1]$, and the threshold at each point is computed as the corrected $(1-\alpha)$ quantile:

$$\theta(t) = \text{Quantile}_{1-\alpha}\left(\bar{P}_t^{\xi} : \xi \in \mathcal{D}_{\text{cal}}^{+}\right) \quad (6)$$

where $\mathcal{D}_{\text{cal}}^{+}$ denotes the set of successful calibration trajectories. Intuitively, $\theta(t)$ answers how high can the score get at this stage in a successful trajectory? If a new trajectory's score exceeds $\theta(t)$, it falls outside the $(1-\alpha)$ statistical envelope at the current point for successful behavior and is flagged as a failure. We characterize the impact of α empirically in Appendix D.

4 Experiment Setup

Benchmarks. LIBERO (Liu et al., 2023) (the LIBERO-10 subset of 10 long-horizon manipulation tasks); CALVIN (Mees et al., 2022) (each subtask in a chained sequence treated as an independent success/failure instance); and ALOHA (Zhao et al., 2023), spanning simulated (transfer, insertion) and real-world tasks, where we collect 120 cup-placement, 60 pick-cube, and 75 drawer-open-block-placement trajectories on the physical platform. We also test the robustness of our method on policies trained with noisy demonstrations from Layman V1.0 (Sojib & Begum, 2024).

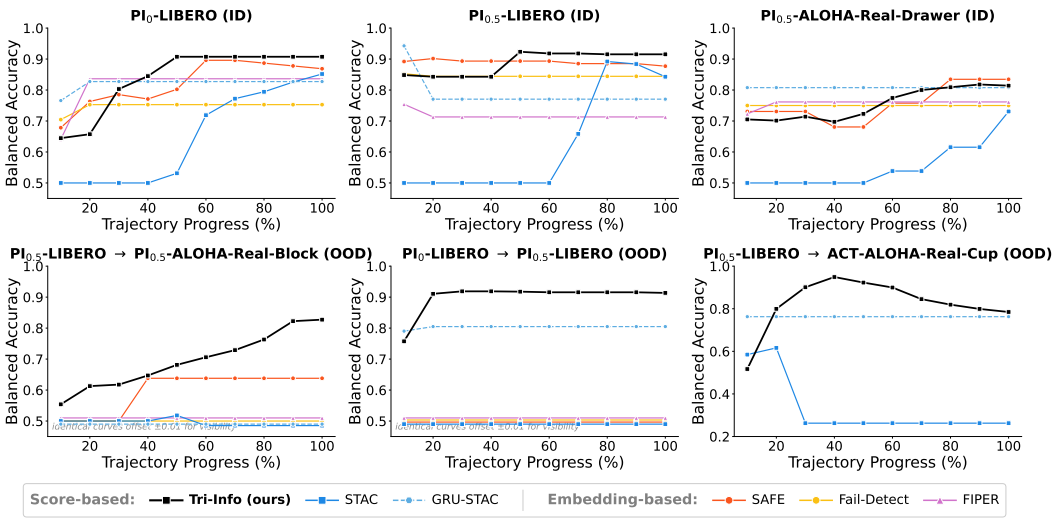

Figure 5: **Baseline comparison.** Failure detection accuracy across different trajectory progress. Our Tri-Info detector achieves competitive accuracy at both early and final time points, and transfers to OOD settings, where no baseline holds up consistently. Embedding-based baselines are omitted on $\text{PI}_{0.5}$-LIBERO → ACT-ALOHA-real-cup, where latent dimensions do not match (a VAE-matched variant is evaluated in Appendix D).

Models. We evaluate six state-of-the-art VLAs: PI_0 (Black et al., 2024) and $\text{PI}_{0.5}$ (Intelligence et al., 2025) on LIBERO-10; FLOWER (Reuss et al., 2025), UNIVLA (Bu et al., 2025), and GR-1 (Wu et al., 2024) on CALVIN; and ACT (Zhao et al., 2023) and $\text{PI}_{0.5}$ on ALOHA, collecting both successful and failed trajectories for each.

Evaluation. We aggregate per-timestep scores by max-pooling, $P^{\xi}_{\max} = \max_t P^{\xi}_t(\text{Failure} \mid x_t)$, mirroring online deployment where an alarm fires once the score crosses $\theta(t)$. Because successful and failed rollouts end at different lengths, we truncate all trajectories to the shortest successful one before pooling, and report AUC on $P^{\xi}_{\max}$. We also report balanced accuracy (BA) over trajectory progress.

Baselines. We compare against *embedding-based* detectors that classify internal model representations (SAFE (Gu et al., 2025), Fail-Detect (Xu et al., 2025), FIPER (Römer et al., 2025)) and the *score-based* STAC (Agia et al., 2024), as well as its variant within our full GRU pipeline (denoted GRU-STAC). To isolate the contribution of MI itself, we also re-implement four alternative action-continuity measures (variance, KNN distance, dot-product similarity, temporal gradient) within our GRU framework. Dataset statistics, training and threshold-calibration details, and these alternatives are in Appendix D.

5 Results

5.1 Predictive Power of Information-Theoretic Metrics

Every metric carries signal; temporal modeling matters. We feed each of the eight metrics as the sole input to two detectors: a memoryless logistic regression (LogReg) that scores it instantaneously, and a GRU that models its temporal evolution (Table 1). Under LogReg, all eight reach ≥ 0.65 average in-domain mBA, with the strongest, $\mathbb{H}(\mathbf{A}_t)$, at 0.75, confirming that the predictive signal resides in the metrics themselves. Modeling temporal context with a GRU lifts all metrics to 0.72–0.77, and the weakest metrics gain the most, so the gap between the best and worst metric closes from 0.10 to 0.05. In conclusion, the metrics are individually informative, and modeling their temporal evolution improves detection performance.

Qualitative behavior of the Tri-Info detector. On a representative ALOHA-real trajectory (Figure 2), the Tri-Info detector raises $P(\text{Failure})$ just *before* each of the two windows (t=97–117 and t=390–477) where the cup nearly tips or falls, demonstrating early

warning ahead of the visible danger. Figure 9 corroborates this on LIBERO-10 ($PI_{0.5}$): successful trajectories stay below the threshold throughout, whereas failed ones cross it.

Interpretability: the metrics align with failure semantics. We annotate failed rollouts as freeze, drift, or phantom grasp (n=20 per mode) and examine each mode's two signature metrics around the trigger t^* along two axes: the change from the trajectory's own pre-trigger level, and the gap to successful trajectories at matched progress (Figure 6). The two axes agree in all six panels. The mutual information $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$ falls in every mode while that of successful trajectories holds or rises: consecutive commands no longer predict one another. The second metric separates the modes. In freeze, $\mathbb{H}(\mathbf{A}_t)$ also collapses as the arm issues less varied actions. In drift, it surges compared with successful trajectories, reflecting erratic motion. In phantom grasp, $\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)$ steps down while that of successful trajectories rises, as the arm repeats a grasp that no longer moves the objects.

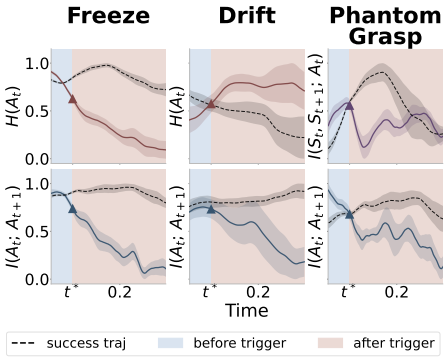

Figure 6: **Failure signatures at the alarm.** Annotated failures (n=20 per mode) aligned at the detector's trigger t^*, against successful trajectories at matched progress (black). Time is the lag from the trigger as a fraction of the episode horizon, $(t - t^*)/T$. The metric trends align with the semantic meaning of each failure.

Robustness to policies trained on non-expert demonstrations. Real-world data is often collected by non-experts and therefore contains noisy, indirect trajectories that nonetheless reach the goal. We thus test whether our detector remains reliable for a policy fine-tuned on such noisy data, and can a detector trained only on clean-policy rollouts transfer to a noisy-policy one? We use Layman V1.0 (Sojib & Begum, 2024), which provides non-expert demonstrations and a 20%-spoiled *Can* split: **Clean** (120 expert demos, 277 steps) vs. **Noisy** (120 expert + 30 lay-user demos in which operators struggle to control the robot but eventually complete the task, 503 steps). We fine-tune $PI_{0.5}$ on each pool, obtaining SR 0.670 (clean) and 0.700 (noisy), and sample rollouts from both. We report BA at each % of trajectory progress in Figure 7 with three seeds each. The detector remains reliable on the noisy-data policy (mBA 0.728), and a detector trained *only* on clean-policy rollouts can transfer to the noisy policy with good performance (mBA 0.716).

5.2 Baseline Comparison

We compare the Tri-Info detector against embedding-based methods (SAFE (Gu et al., 2025), Fail-Detect (Xu et al., 2025), FIPER (Römer et al., 2025)), the score-based STAC (Agia et al., 2024), and its GRU variant in our pipeline (GRU-STAC). Beyond the in-domain setting, we evaluate three transfers of increasing severity: cross-model, cross-environment, and cross-model-environment. Because embedding-based latents are architecture-specific, the embedding-based detectors cannot be applied in the last setting, where the target policy's latent dimensionality differs from the source. Figure 5 reports balanced accuracy versus trajectory progress.

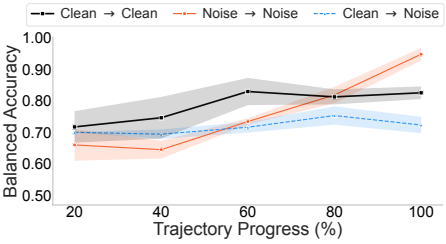

Figure 7: **Tri-Info detection BA ± std across training-data sources under clean and noisy conditions.** Tri-Info remains accurate on a policy trained on noisy demonstrations, and can transfer from clean-policy rollouts to noisy-policy rollouts.

In-domain: Tri-Info is competitive with existing detectors. Tri-Info averages 0.83 balanced accuracy (BA) in domain, against 0.82 for SAFE and 0.81 for GRU-STAC. It reaches 0.84 and 0.89 on PI_0- and $PI_{0.5}$-LIBERO-10, where SAFE reaches 0.82 and 0.89, and 0.76 on $PI_{0.5}$-ALOHA-real-drawer, where GRU-STAC is highest at 0.81. Raw STAC is the only method below 0.70 (0.61), which shows that the consistency statistic needs the temporal model to be predictive. All remaining methods fall within 0.07 BA of each other, so in-domain accuracy does not

Table 3: Estimator-input ablation. Mean balanced accuracy over $t \in \{0.1, \ldots, 1.0\}$, three seeds. Splits and block means follow Table 2, with cross-model being PI_0-LIBERO to $PI_{0.5}$-LIBERO and sim-to-real being $PI_{0.5}$-LIBERO to ACT-ALOHA-real. $\mathbf{a}_t, \mathbf{s}_t$ are the VAE latents, $\mathbf{s}_t$ fusing image, wrist-image and proprioceptive features; a_t is the executed action and q_t the raw proprioception, both used without a VAE.

Inputs	PI_0-LIBERO	$PI_{0.5}$-LIBERO	ID AVG.	Cross-Model	Sim-to-Real	OOD AVG.
$\mathbf{a}_t + \mathbf{s}_t$ (ours)	0.870	0.893	0.881	0.906	0.824	**0.865**
$\mathbf{a}_t + q_t$	0.864 −0.006	0.888 −0.005	0.876 −0.005	0.913 +0.007	0.547 −0.277	0.730 −0.135
$a_t + \mathbf{s}_t$	0.907 +0.037	0.887 −0.006	**0.897** +0.016	0.771 −0.135	0.752 −0.072	0.761 −0.104
$a_t + q_t$ (API-only)	0.878 +0.008	0.882 −0.011	0.880 −0.001	0.859 −0.047	0.729 −0.095	0.794 −0.071

distinguish them. The distinguishing question is whether that accuracy survives a change of model or embodiment.

Cross-domain: only Tri-Info retains accuracy under shift. We evaluate three transfers of increasing severity in Figure 5. Changing the policy alone (PI_0-LIBERO → $PI_{0.5}$-LIBERO) already separates the methods: Tri-Info reaches 0.90 and GRU-STAC 0.80, while SAFE, Fail-Detect, and FIPER fall to 0.50, because their classifiers are fitted to the absolute geometry of one policy's latent space. Changing the environment instead (→ $PI_{0.5}$-ALOHA-real-block) leaves Tri-Info at 0.70 and SAFE at 0.60, with all other baselines at 0.50. Changing model, embodiment, action dimensionality, and visual appearance at once (→ ACT-ALOHA-real-cup) makes the embedding-based detectors inapplicable, since their input dimensionality is fixed to the source policy; GRU-STAC reaches 0.76 and STAC falls to 0.33, below chance, so its score anti-correlates with failure, whereas Tri-Info attains 0.82. A VAE projection that matches latent dimensions restores partial applicability for the embedding-based detectors (Appendix D).

Table 2: Multi-metric detector ablation on mean balanced accuracy. ID is pooled in-domain; OOD averages cross-model and sim-to-real transfer.

$\mathbb{H}(\mathbf{A}_t)$	$\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	$\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)$	ID	OOD
✓	✓	✓	**.88**	**.87**
✓	✓		**.87**	**.85**
✓		✓	.86	.70
	✓	✓	.86	.84
✓			**.87**	.60
	✓		.84	**.77**
		✓	**.82**	.67
All 8 Features			.87	.74

5.3 Ablation Study

Metric ablation: Tri-Info's three signals contribute to OOD performance. We compare the three-metric detector against all of its 2- and 1-metric subsets and the full 8-metric fusion (Table 2). The three-metric set improves over every smaller subset on both ID (0.88) and OOD (0.87), while expanding to eight metrics holds ID flat (0.87) and substantially degrades OOD to 0.74 – the additional five signals overfit in-distribution patterns and fail to transfer.

Input ablation: how much of the signal needs model internals? Everything so far is estimated on VAE latents of the policy's internal embeddings, which requires white-box access. We re-ran the pipeline changing only the estimator inputs, replacing $\mathbf{a}_t$ by the executed action a_t and $\mathbf{s}_t$ by the raw proprioception q_t, both visible from outside the policy and low-dimensional enough to enter the estimators without a VAE, unlike the raw observation s_t. In domain, the internal representations make no difference to detection accuracy: the four choices lie between 0.876 and 0.897, and the highest is $a_t + \mathbf{s}_t$. Under shift they do: the OOD mean falls from 0.865 with the embeddings to 0.794 for the black-box $a_t + q_t$. Tri-Info therefore runs in a fully black-box setting, and the white-box requirement earns its cost only under domain shift.

6 Conclusion

We formalized VLA control as a closed-loop information pipeline and reduced eight derived metrics to three complementary Tri-Info metrics $\{\mathbb{H}(\mathbf{A}_t),\ \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1}),\ \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)\}$ that diagnose action collapse, temporal incoherence, and action–state decoupling. Paired with a per-signal GRU detector, Tri-Info stays interpretable and generalizable, transferring across VLA models, benchmarks, and the sim-to-real gap to reach 70%+ on real-world tasks.

AI USE STATEMENT

Large Language Models (LLMs) were employed solely to support the writing and polishing of this manuscript, namely rephrasing sentences, checking grammar, and improving clarity and readability. They played no part in research ideation, methodology, experimental design, or data analysis. The authors reviewed and revised all AI-assisted text and take full responsibility for the final content of the manuscript.

REPRODUCIBILITY STATEMENT

The full algorithm and implementation details of Tri-Info are presented in the main text. To support replication of our experiments, all implementation parameters, experimental settings, and robot hardware specifications are given in the Appendix. We also release the code needed to reproduce the Tri-Info detector in the supplementary material.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *ICML*, pp. 22–31, 2017.

Christopher Agia, Rohan Sinha, Jingyun Yang, Zi-ang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. *arXiv preprint arXiv:2410.04640*, 2024.

W Ross Ashby. Introduction to cybernetics.-chapman & hall. *London, UK*, 1956.

Shuanghao Bai, Wanqi Zhou, Pengxiang Ding, Wei Zhao, Donglin Wang, and Badong Chen. Rethinking latent redundancy in behavior cloning: An information bottleneck approach for robot manipulation. *arXiv preprint arXiv:2502.02853*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Jacopo Diquigiovanni, Matteo Fontana, and Simone Vantini. The importance of being a band: Finite-sample exact distribution-free prediction sets for functional data. *arXiv preprint arXiv:2102.06746*, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Yoshua Bengio, and Sergey Levine. Infobot: Transfer and exploration via the information bottleneck. *arXiv preprint arXiv:1901.10902*, 2019.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. Safe: Multitask failure detection for vision-language-action models. *ArXiv*, abs/2506.09937, 2025. URL https://api.semanticscholar.org/CorpusID:279306316.

Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Bhattacharjee. A review of safe reinforcement learning: Methods, theories and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, Pannag Sanketi, Dhruv Shah, Coline Devin, and Dorsa Sadigh. Robot data curation with mutual information estimators. *arXiv preprint arXiv:2502.08623*, 2025.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Rich Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. π0.5: a vision-language-action model with open-world generalization. *ArXiv*, abs/2504.16054, 2025. URL https://api.semanticscholar.org/CorpusID:277993634.

Shutong Jin, Lezhong Wang, Ben Temming, and Florian T Pokorny. Physically-based lighting augmentation for robotic manipulation. *arXiv preprint arXiv:2508.01442*, 2025.

Hyoungseok Kim, Jaekyeom Kim, Yeonwoo Jeong, Sergey Levine, and Hyun Oh Song. Emi: Exploration with mutual information. *arXiv preprint arXiv:1810.01176*, 2018.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pp. 128–135. IEEE, 2005.

Leonenko Kozachenko. Sample estimate of the entropy of a random vector. *Probl. Pered. Inform.*, 23:9, 1987.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 69(6):066138, 2004.

Shunlei Li, Longsen Gao, Jin Wang, Chang Che, Xi Xiao, Jiuwen Cao, Yingbai Hu, and Hamid Reza Karimi. Information-theoretic graph fusion with vision-language-action model for policy reasoning and dual robotic control. *ArXiv*, abs/2508.05342, 2025. URL https://api.semanticscholar.org/CorpusID:280546283.

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, 2023.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. *arXiv preprint arXiv:2509.04996*, 2025.

Ralf Römer, Adrian Kobras, Luca Worbis, and Angela P Schoellig. Failure prediction at runtime for generative robot policies. *arXiv preprint arXiv:2510.09459*, 2025.

Hrishikesh Sathyanarayan and Ian Abraham. Behavior synthesis via contact-aware fisher information maximization. *arXiv preprint arXiv:2505.12214*, 2025.

William Saunders, Girish Sastry, Andreas Stuhlmüller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *AAMAS*, pp. 2067–2069, 2018.

Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL, 1949.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Noushad Sojib and Momotaz Begum. Self supervised detection of incorrect human demonstrations: A path toward safe imitation learning by robots in the wild. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2862–2869, 2024. doi: 10.1109/IROS58592.2024.10802106.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3): 4915–4922, 2021.

Naftali Tishby and Daniel Polani. Information theory of decisions and actions. In *Perception-action cycle: Models, architectures, and hardware*, pp. 601–636. Springer, 2010.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Hugo Touchette and Seth Lloyd. Information-theoretic limits of control. *Physical review letters*, 84 (6):1156, 2000.

Norbert Wiener. *Cybernetics: Or Control and Communication in the Animal and the Machine*. The Technology Press; John Wiley & Sons, Cambridge, MA; New York, 1948.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3153–3160. IEEE, 2024.

Wenpeng Xing, Minghao Li, Mohan Li, and Meng Han. Towards robust and secure embodied ai: A survey on vulnerabilities and attacks. *arXiv preprint arXiv:2502.13175*, 2025.

Chen Xu, Tony Khuong Nguyen, Emma Dixon, Christopher Rodriguez, Patrick Miller, Robert Lee, Paarth Shah, Rares Ambrus, Haruki Nishimura, and Masha Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies. *arXiv preprint arXiv:2503.08558*, 2025.

Bang You and Huaping Liu. Multimodal information bottleneck for deep reinforcement learning with multiple sensors. *Neural Networks*, 176:106347, 2024.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

Jiaming Zhou, Ke Ye, Jiayi Liu, Teli Ma, Zifan Wang, Ronghe Qiu, Kun-Yu Lin, Zhilin Zhao, and Junwei Liang. Exploring the limits of vision-language-action manipulations in cross-task generalization. *arXiv preprint arXiv:2505.15660*, 2025.

Pei Zhou and Yanchao Yang. Maxmi: A maximal mutual information criterion for manipulation concept discovery. In *European Conference on Computer Vision*, pp. 88–105. Springer, 2024.

Minjie Zhu, Yichen Zhu, Jinming Li, Zhongyi Zhou, Junjie Wen, Xiaoyu Liu, Chaomin Shen, Yaxin Peng, and Feifei Feng. Objectvla: End-to-end open-world object manipulation without demonstration. *arXiv preprint arXiv:2502.19250*, 2025.

A Limitations.

Firstly, the scope of this paper is mainly alarms, rather than recovery or improvement of the success rate. We do try using the detector to select among sampled actions at inference time, which does raise the success rate (Appendix E), though closing the loop deserves a more systematic study than we give it here. Secondly, this work is also tested only on VLAs; we additionally try a non-VLA behavioral-cloning policy (Appendix E), which suggests the method may apply to closed-loop policies in general, but confirming this requires a wider range of architectures. Finally, our approach relies on a VAE to compress high-dimensional embeddings into a low-dimensional latent space where entropy and mutual information can be estimated. Therefore, our method requires training a VAE to compress the embeddings. Future work could remove this compression step by adopting information estimators that scale to high-dimensional spaces.

B Validating the Three-Signal Reduction

B.1 Exhaustive Subset Search

We cross-check the metric reduction with an exhaustive search whose purpose is to confirm *which* signals to retain. We use logistic regression rather than the GRU: a memoryless classifier exposes the intrinsic discriminative content of each metric, so the resulting ranking is attributable to the metrics themselves rather than to the temporal model. Scoring a subset instantaneously is also cheap enough to cover the whole lattice, so we evaluate all $2^8 - 1 = 255$ non-empty subsets, fitting one logistic regression per subset on per-timestep features pooled over all environment-model combinations excluding OOD domain in Figure. 5. Table 4 reports the four highest-AUC subsets at each cardinality.

Table 4: Detection performance AUC across different feature combinations. Best results per subset size are **bolded**.

Size	Features	AUC
2	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	**0.898**
	$\mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.896
	$\mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.895
	$\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.895
3	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	**0.900**
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.899
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.899
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.898
4	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	**0.900**
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.899
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.899
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.899
5	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	**0.900**
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.899
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.899
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.894
6	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	**0.895**
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.892
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.889
	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.888
7	$\mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	**0.888**
	$\mathbb{H}(\mathbf{S}_t), \mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.879
	$\mathbb{H}(\mathbf{S}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.877
	$\mathbb{H}(\mathbf{S}_t), \mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.877
8	$\mathbb{H}(\mathbf{S}_t), \mathbb{H}(\mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{A}_t, \mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t), \mathbb{I}(\mathbf{S}_t; \mathbf{S}_{t+1}), \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$	0.876

Table 5: Action and state variational autoencoder (VAE) encoder architectures. Both branch into parallel mean and log-variance heads at a matched d_z-dim latent ($d_z = 7$); decoders mirror the encoders with a sigmoid output. The VAE state encoder runs once per input modality (images, proprioception); n_{mod} denotes the number of input modalities, which varies across models, and d_{int} the per-modality intermediate code width ($d_{\text{int}} = 8$), so each modality head emits $2d_{\text{int}}$ values for its own mean/log-variance pair.

Action VAE	State VAE (per modality)
$\text{Linear}(\to 500)$	$\text{Linear}(\to \min(512, \text{in}/4))$
LeakyReLU	LayerNorm + LeakyReLU + Dropout
$\text{Linear}(500 \to 500)$	$\text{Linear}(\to 2d_{\text{int}})$
LeakyReLU	Concatenate n_{mod} modalities $\to d_{\text{int}}\, n_{\text{mod}}$
	$\text{Linear}(d_{\text{int}}\, n_{\text{mod}} \to 64)$ + LayerNorm + LeakyReLU
Mean / log-var heads: $500 \to d_z$	Mean / log-var heads: $64 \to d_z$

The search independently recovers $\{\mathbb{H}(\mathbf{A}_t),\ \mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1}),\ \mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)\}$ as the top-AUC subset of size 3 (0.900), though the AUC is very flat across ID performance. Performance is bounded at this point rather than improved by adding signals: size 4 and size 5 match 0.900 without surpassing it, and size ≥ 6 declines steadily to 0.876 at size 8, as the remaining metrics are correlated with those already selected (Figure 4) and contribute redundancy rather than complementary discrimination. The optimal subsets at every size ≥ 2 retain the action-centric pair $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$ and $\mathbb{H}(\mathbf{A}_t)$, with $\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)$ joining at size 3, consistent with our preference for action-centric signals: successful execution maintains consistent and diverse actions that remain coupled to state transitions.

Table 6: Sensitivity to the window size W and the neighborhood size k of the k-NN estimators. Each sweep varies one while holding the other at its default ($k = 5$, $W = 30$). Best per row in bold.

k	5	6	8	9	10
AUC	0.915	0.893	**0.918**	**0.918**	0.893
Avg. BA	**0.893**	**0.893**	0.892	0.886	0.881

W	10	20	30	40	50
AUC	0.913	0.888	**0.915**	0.875	0.683
Avg. BA	0.865	0.851	**0.893**	0.879	0.806

C MI Estimation

We estimate the three Tri-Info quantities $\mathbb{H}(\mathbf{A}_t)$, $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$, and $\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)$ non-parametrically over a sliding window of length W, using the Kozachenko–Leonenko estimator (Kozachenko, 1987) for the entropy and the Kraskov–Stögbauer–Grassberger estimator (Kraskov et al., 2004) for the mutual-information terms, both under the max-norm with $k = 5$ neighbours and $N = W - 1$ samples per window.

VAE projection. Raw embeddings are too high-dimensional for k-NN estimation, where neighbour distances concentrate and the estimator bias grows. Following Hejna et al. (2025), who estimate state–action mutual information with the same estimator family, we compress each embedding to a matched d_z=7 latent with a VAE fitted per model–dataset domain (Table 5; Appendix D.2) and estimate on the posterior means. A VAE rather than an arbitrary projection is what makes the distances usable: its isotropic Gaussian prior gives the latent a well-conditioned geometry, a prerequisite for k-NN statistics and the regime in which these estimators are best characterized. Keeping d_z small serves the estimator, as they argue, and matching it across the two encoders keeps the joint dimensions of the three metrics comparable; their ablation over a threefold range of latent sizes reports little effect, and we did not repeat that sweep here.

Assumptions and what the estimates support. Our windowed use stresses the estimators more than the pooled use they were validated under: Hejna et al. (2025) draw batches of 1024 samples from a whole dataset, whereas a window affords 29 consecutive – hence autocorrelated – steps at the default $W = 30$, and they take the L_2 norm within each latent block and the max-norm between blocks, where we use the max-norm in every subspace. Local uniformity inside the k-NN ball is correspondingly loose, across up to 21 joint dimensions for $\mathbb{I}(\mathbf{S}_t, \mathbf{S}_{t+1}; \mathbf{A}_t)$, and the entropy estimator diverges as neighbour distances vanish, so we floor them at 10^{-15}: a degenerate window yields a finite floor for $\mathbb{H}(\mathbf{A}_t)$ rather than a divergence. Absolute values in nats are therefore not meaningful and we never use them as such; as in Hejna et al. (2025), only relative comparisons carry weight. We accordingly read each metric as a normalized time series whose shape rather than level drives the GRU, and let the detector integrate the sequence of per-window estimates rather than trust any one of them. This is also why W is the one consequential hyperparameter below – it sets both the sample count and the span over which the regime must be stationary – while detection is insensitive to k, matching the robustness Hejna et al. (2025) report over $k \in \{2, \ldots, 10\}$.

Sensitivity to Window Size W and Neighborhood Size k We sweep the two estimator hyperparameters on the deployed detector (Tri-Info with mean fusion on $\text{PI}_{0.5}$), varying W and k one at a time around their defaults (Table 6, Figure 8). Detection is largely insensitive to k, spanning 0.893–0.918 AUC and 0.881–0.893 BA over the sweep, with the default $k = 5$ within 0.003 AUC of the best value. W is the consequential choice: AUC stays within 0.875–0.915 for $W \leq 40$ and falls to 0.683 at $W = 50$, where BA also drops to 0.806, since long windows blur the failure transient the detector reads. The progress curves echo this, spreading under the W sweep and clustering under the k sweep. We therefore use $W = 30$, the peak of both measures, and leave $k = 5$ at its robust default.

D Experiment Details

We describe the datasets (Appendix D.1), the training and evaluation protocol (Appendix D.2), the baselines (Appendix D.3), and additional results and sensitivity analyses (Appendix D.5).

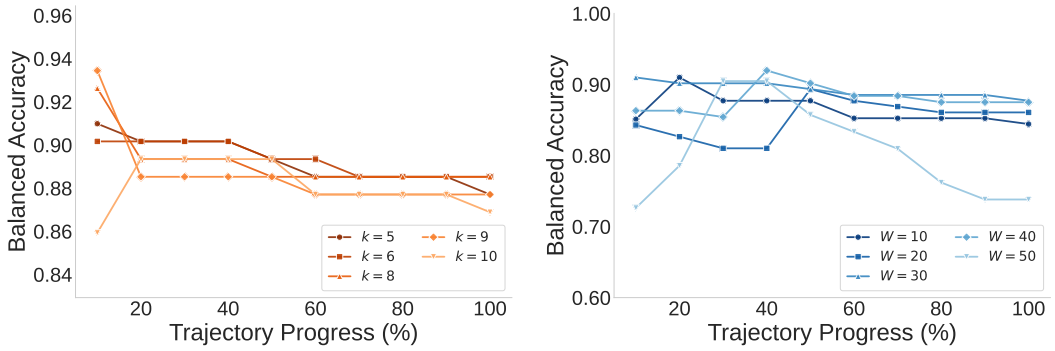

Figure 8: Balanced accuracy versus trajectory progress for the Tri-Info detector ($PI_{0.5}$-LIBERO) under the two estimator hyperparameter sweeps. **Left:** neighborhood size $k \in \{5, 6, 8, 9, 10\}$ at fixed $W = 30$; the curves are tightly clustered, indicating that detection is insensitive to k. **Right:** window length $W \in \{10, 20, 30, 40, 50\}$ at fixed $k = 5$; the wider spread, with $W = 30$ highest, shows that W is the more consequential hyperparameter.

Table 7: Dataset statistics.

Dataset	**Success**	**Fail**	**Total**	**Suc%**
PI_0-LIBERO	245	55	300	81.7%
$PI_{0.5}$-LIBERO	875	75	950	92.1%
ACT-ALOHA-real	60	60	120	50.0%
$PI_{0.5}$-ALOHA-real-block	26	34	60	43.3%
$PI_{0.5}$-ALOHA-real-drawer	31	44	75	41.3%
FLOWER-CALVIN	224	76	300	74.7%
ACT-ALOHA-sim-insertion	65	235	300	21.7%
ACT-ALOHA-sim-transfer	259	41	300	86.3%
GR-1-CALVIN	360	50	410	87.8%
UniVLA-CALVIN	215	35	250	86.0%
Total	**2360**	**705**	**3065**	**77.0%**

D.1 Datasets

Table 7 summarizes the data collected for each model–environment combination.

Real-world data collection. Real-world demonstrations were collected on the ALOHA platform (AgileX cobot_magic variant) using a single 6-DoF arm with a parallel gripper and three RGB cameras (one global and two wrist-mounted). We recorded demonstrations via master–puppet teleoperation after sensor synchronization. Each trajectory logs the arm's 7-dimensional joint positions (six joints plus the gripper), the master-side control commands, and synchronized 480×640 RGB observations, all sampled at 50 Hz.

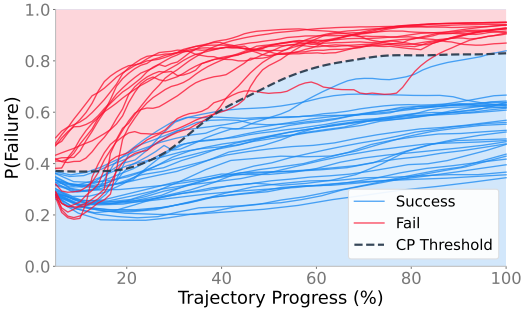

Figure 9: **Tri-Info detector per-timestep $P(\text{Failure})$ on $PI_{0.5}$-LIBERO over 60 random samples.** Failed trajectories (red) cross the threshold $\theta(t)$; successful ones (blue) stay below it.

Real-world task 1: Cup on plate (ACT). The task is to place a red cup onto a small blue plate. A trial succeeds if the cup comes to rest within the plate boundary without toppling; all other outcomes count as failures. We collected 100 teleoperated demonstrations, combined them with simulated data, and trained ACT with batch size 4 for

10,000 epochs. For evaluation, we collected 120 rollout trajectories from the trained policy. More trajectories with Tri-Info detector score are shown in Figure 10.

Real-world task 2: Block in cup ($\text{PI}_{0.5}$). The task is to place a yellow block into a cup. A trial succeeds if the block comes to rest inside the cup; missed grasps, releases above the rim, and toppled cups; or put into other containers all count as failures. We collected 50 teleoperated demonstrations (about 400 frames each), converted them to the LeRobot v2.0 format, and fine-tuned the pretrained $\text{PI}_{0.5}$ base model with `openpi` for 100,000 steps (global batch size 64, cosine schedule peaking at 5×10^{-5}), reusing the Trossen normalization statistics. For evaluation, we collected 60 rollout trajectories from the trained policy. Trajectories with Tri-Info detector score are shown in Figure 10.

Real-world task 3: Block into drawer ($\text{PI}_{0.5}$). The task is a four-stage sequence: open the drawer, pick up the red block, place it inside, and close the drawer. A trial counts as a failure if the block is not placed inside the drawer, or if the drawer cannot be closed afterwards. We collected 56 teleoperated demonstrations of roughly 650 frames each, converted them to the LeRobot v2.0 format, and fine-tuned the pretrained $\text{PI}_{0.5}$ base model with `openpi` for 30,000 steps (global batch size 32, cosine schedule peaking at 2.5×10^{-5}); normalization statistics were computed from the same 56 demonstrations. For evaluation we collected 75 rollout trajectories from the trained policy, of which 31 succeeded and 44 failed (41.3% success rate). Trajectories with their Tri-Info detector scores are shown in Figure 11.

D.2 Training and Evaluation Protocol

This section describes how data are split, how the detectors are trained, and where each component of the pipeline is fitted.

Trajectory-level splits. Every split is made over whole trajectories, *before* any sliding window is formed, and is stratified by the outcome label f^{ξ}. A window $\mathbf{W}_t$ therefore never spans two trajectories, and no trajectory contributes windows to more than one split, so near-duplicate windows cannot straddle the split boundary. In-domain experiments split each dataset $70/30$ into a training and a test split; transfer experiments fit everything on the source domain and evaluate on target-domain trajectories the detector has never seen. The calibration set $\mathcal{D}^{+}_{\text{cal}}$ is the success subset of the *source training split*, so calibration consumes no target trajectory. Because those successes also train the GRUs, $\theta(t)$ is calibrated in-sample, and we therefore read α as an operating-point knob (Section 5.3) rather than an exact finite-sample coverage guarantee.

Failure detector training. We train classifiers at the timestep level, where each timestep inherits the trajectory-level label $f^{\xi} \in \{0, 1\}$. The GRU model consists of a single-layer GRU mapping the per-timestep metric scalar (input dimension one) to a hidden state, followed by a two-layer MLP head that halves the hidden size before a single output, with ReLU activation, dropout 0.1, and a sigmoid output. It is trained with Adam (learning rate 10^{-3}, weight decay 10^{-5}); the hidden size and the number of epochs are not fixed but selected per domain by the search of Appendix D.4 (Table 15). The logistic regression baseline uses a single linear layer with sigmoid activation, trained with the L-BFGS solver at an inverse regularization strength of 1.0. Both models receive the per-timestep metrics, normalized using global statistics computed across the entire training set. Unless otherwise noted, every experiment is run with three random seeds, and we report the average across seeds.

VAE training data and leakage. The projection of Appendix C is fitted once per model–dataset domain, on all of that domain's per-step embeddings – test trajectories included – using reconstruction plus a weak KL penalty. A target domain therefore contributes only unlabelled embeddings to its own projection, and no failure label enters the VAE training pipeline anywhere. The projection does stay transductive in the unlabelled data; fitting it on a disjoint unlabelled pool would remove even that.

Where each component is fitted. Table 8 summarizes where each component is fitted. Only the label-free front end is refitted for a new model or embodiment; the GRU weights, the normalization constants (μ, σ), and the threshold $\theta(t)$ are fitted once on the source training split and applied unchanged, with target sequences z-normalized by *source* statistics rather than their own.

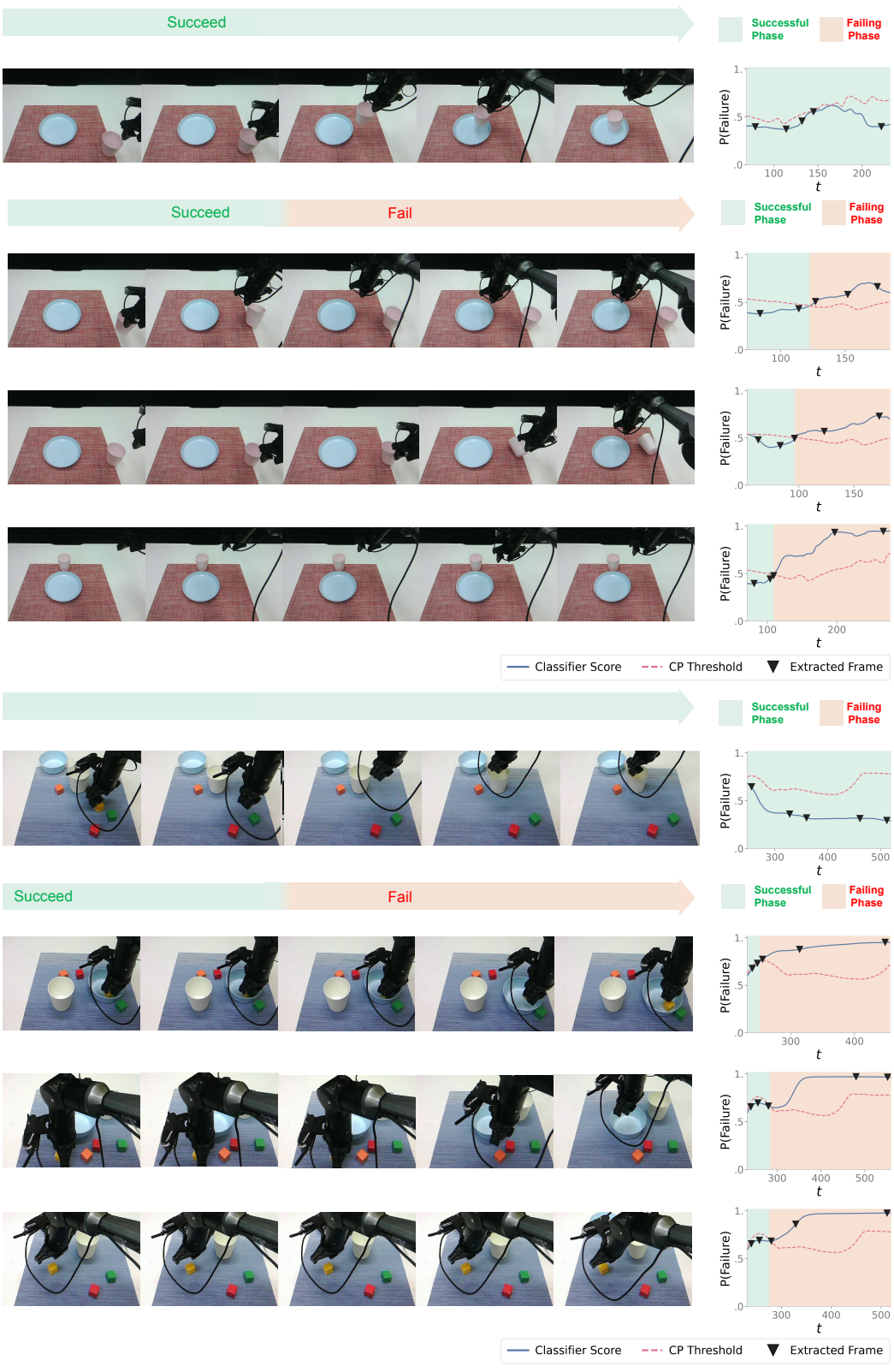

Figure 10: Tri-Info detector results for two real-world platforms (top: ACT-ALOHA-real; bottom: $PI_{0.5}$-ALOHA-real-block). Each platform shows four rollouts – three failed and one successful – and each rollout pairs five trajectory frames with the corresponding classifier score and CP threshold ($\alpha = 0.05$).

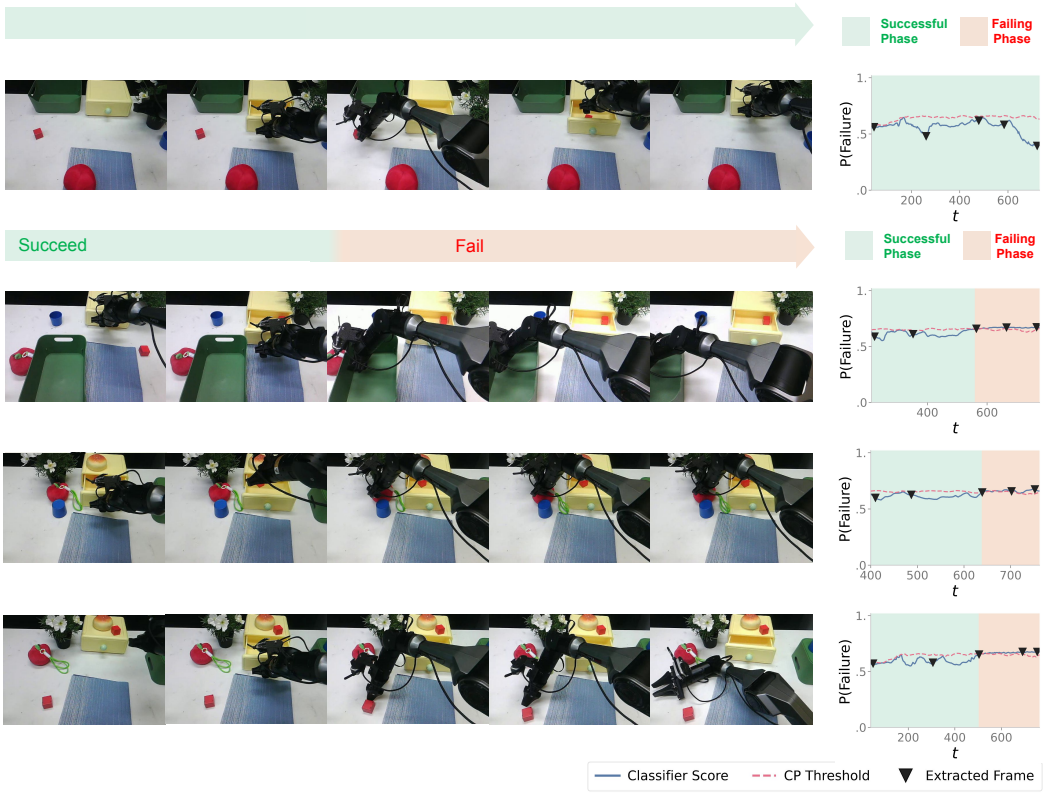

Figure 11: Tri-Info detector results for $\text{PI}_{0.5}$-ALOHA-real-drawer. It shows four rollouts – three failed and one successful – and each rollout pairs five trajectory frames with the corresponding classifier score and CP threshold ($\alpha = 0.05$).

Table 8: Where each component of the pipeline is fitted. Only the label-free front end is refitted for a new model or embodiment.

Component	Fitted on	Uses label	Refit per domain
VAE $E_{\mathbf{S}}$, $E_{\mathbf{A}}$	unlabelled rollouts of that domain	no	yes
Normalization (μ, σ)	source training split	no	no
Per-metric GRUs	source training split	yes	no
Threshold $\theta(t)$	successes of the source training split	yes	no

D.3 BASELINES

This section describes the baseline failure detection methods used in our comparisons, as well as the alternative action continuity metrics we evaluate to validate our design choices. We compare against four recent failure detection methods, which we categorize based on their *input modality into the classifier*: *embedding-based* methods that use model representations, and *score-based* methods that operate on computed scalar signals. This distinction is important, as embedding-based approaches are inherently tied to specific model architectures and cannot generalize across different policies without retraining.

Embedding-based methods. SAFE (Gu et al., 2025) trains a classifier on VLA latent representations to distinguish successful from failed executions, so both its input dimensionality and its decision geometry are fixed to that of embedding space, and it is bound to the architecture by construction. **Fail-Detect** (Xu et al., 2025) distills the policy's inputs and outputs into a scalar signal, which makes it look model-agnostic. But its strongest score (logpZO) is a density estimator trained on one policy's observation embeddings. On a different model or task, the embeddings no longer match what it learned, so almost every trajectory looks out-of-distribution; the score saturates and can no longer tell success from failure. **FIPER** (Römer et al., 2025) combines two signals: (i) out-of-distribution observations via Random Network Distillation in the policy's embedding space (RND-OE), and (ii) action-chunk entropy (ACE) over sampled actions. The RND-OE branch detects novelty using a predictor network trained on one policy's embeddings, so it is tied to that embedding space and fails once the model or task changes, for the same reason as Fail-Detect. The ACE branch, however, reads only the sampled action chunks and trains no model bound to the source policy, so it does carry over. We therefore take ACE as FIPER's transferable component and compare against it in Figure 13.

Score-based methods. STAC (Agia et al., 2024) computes the statistical divergence between overlapping segments of sampled action chunks to measure temporal consistency in generative policies. The original method requires generating multiple samples per timestep; our $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1})$ shares a similar spirit but differs in two respects: (i) it operates on executed actions over a sliding window, compatible with deterministic VLA inference and adding much less overhead, whereas STAC compares only an overlapping sub-window of 32+ sampled futures; (ii) the decomposition $\mathbb{I}(\mathbf{A}_t; \mathbf{A}_{t+1}) = \mathbb{H}(\mathbf{A}_{t+1}) - \mathbb{H}(\mathbf{A}_{t+1}|\mathbf{A}_t)$ captures both diversity and consistency, while STAC measures consistency only.

Dimension-matched embedding baselines with VAE compression. The embedding-based detectors cannot be applied when the target policy's latent dimensionality differs from the source, which is the case for $\text{PI}_{0.5}$-LIBERO $\rightarrow$ ACT-ALOHA-real-cup. To evaluate them in that setting as well, we project the source and target embeddings through the per-domain VAE to the common $d_z = 7$ latent and train and apply each detector on those latents, giving the variants SAFE-VAE, Fail-Detect-VAE, and FIPER-VAE. Table 9 reports trajectory-level AUC and mean balanced accuracy for all methods, and Figure 12 shows balanced accuracy versus trajectory progress for the dimension-matched variants.

In domain, the projection lowers the average accuracy of all three embedding-based detectors, as expected from compressing a high-dimensional latent to seven dimensions: SAFE drops from 0.91 to 0.85 AUC and from 0.82 to 0.79 mBA, while Fail-Detect and FIPER lose at most 0.02 AUC and 0.06 mBA. Under shift, the shared latent makes cross-domain evaluation possible: all three detectors now apply to $\text{PI}_{0.5}$-LIBERO $\rightarrow$ ACT-ALOHA-real-cup, where the raw variants cannot run. The shared latent also improves transfer for some detectors. SAFE-VAE outperforms raw SAFE on both transfers where the two are comparable, raising AUC from 0.61 to 0.86 on $\text{PI}_{0.5}$-LIBERO $\rightarrow$

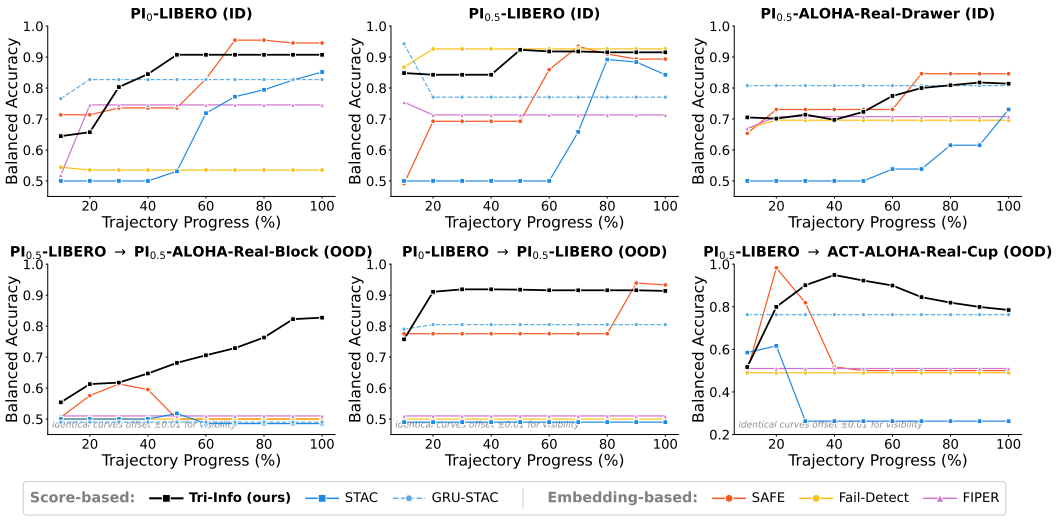

Figure 12: **Dimension-matched baselines.** Balanced accuracy versus trajectory progress for the three in-domain settings (top) and the three transfers (bottom). All embedding-based detectors use the VAE-matched latent, so they are applicable in every panel, including $\text{PI}_{0.5}$-LIBERO $\rightarrow$ ACT-ALOHA-real-cup. In the OOD panels overlapping curves are offset by ± 0.01 for visibility.

$\text{PI}_{0.5}$-ALOHA-real-block and from 0.64 to 0.74 on PI_0-LIBERO $\rightarrow$ $\text{PI}_{0.5}$-LIBERO. On the latter transfer it also raises mBA from 0.50 to 0.81. FIPER-VAE likewise raises AUC on PI_0-LIBERO $\rightarrow$ $\text{PI}_{0.5}$-LIBERO from 0.56 to 0.77. The gain is not uniform, however: Fail-Detect-VAE falls below chance on all three transfers (0.32 average AUC). Nor does it extend to deployable accuracy. Fail-Detect-VAE and FIPER-VAE sit at 0.50 mBA on every OOD setting, and SAFE-VAE averages 0.64. The cases where AUC stays high but mBA collapses identify the failure mode. These cases are SAFE-VAE on the two transfers to real ALOHA (0.86 vs. 0.53 and 0.79 vs. 0.58) and FIPER-VAE on PI_0-LIBERO $\rightarrow$ $\text{PI}_{0.5}$-LIBERO (0.77 vs. 0.50). There the scores still rank trajectories within the target domain, but their absolute level shifts. The threshold $\theta(t)$ calibrated on the source domain therefore no longer separates success from failure, and the deployed detector operates at or near chance.

Alternative action metrics. We further consider several metrics that summarize how an action sequence evolves over time, each feeding into the same GRU-based detection pipeline. All are computed over the same sliding window $\mathbf{W}_t = \{(\mathbf{s}_i, \mathbf{a}_i)\}_{i=t-W+1}^{t}$ of Section 3.2, using the action embeddings $\mathbf{a}_i$.

1. **Variance** measures action dispersion within the window: $\text{Var}_t = \text{tr}\,\text{Cov}\left(\{\mathbf{a}_i\}_{i=t-W+1}^{t}\right)$.
2. **KNN distance** measures local density: $\text{KNN}_t = \frac{1}{Wk}\sum_{i=t-W+1}^{t}\sum_{j=1}^{k}\|\mathbf{a}_i - \mathbf{a}_{\text{nn}_j(i)}\|_2$, where $\text{nn}_j(i)$ is the j-th nearest neighbor of $\mathbf{a}_i$ within $\mathbf{W}_t$.
3. **Cosine dissimilarity** measures consecutive action alignment: $\text{CosDis}_t = 1 - \frac{1}{W-1}\sum_{i=t-W+1}^{t-1}\frac{\mathbf{a}_i^\top \mathbf{a}_{i+1}}{\|\mathbf{a}_i\|\,\|\mathbf{a}_{i+1}\|}$.
4. **Temporal gradient** measures the rate of action change: $\text{Grad}_t = \frac{1}{W-1}\sum_{i=t-W+1}^{t-1}\|\mathbf{a}_{i+1} - \mathbf{a}_i\|_1$.

These metrics summarize different aspects of an action sequence (dispersion, local density, directional alignment, and rate of change). We evaluate each one twice, on the raw action embeddings and on the VAE-matched latent, which separates the contribution of the information-theoretic form from that of the projection.

Comparison against action-based scores. Table 10 and Figure 13 compare Tri-Info against FIPER-ACE and the four action-continuity scores. In domain, the alternatives are competitive and two of them are stronger than Tri-Info: on raw embeddings, kNN distance averages 0.84 mBA against 0.83

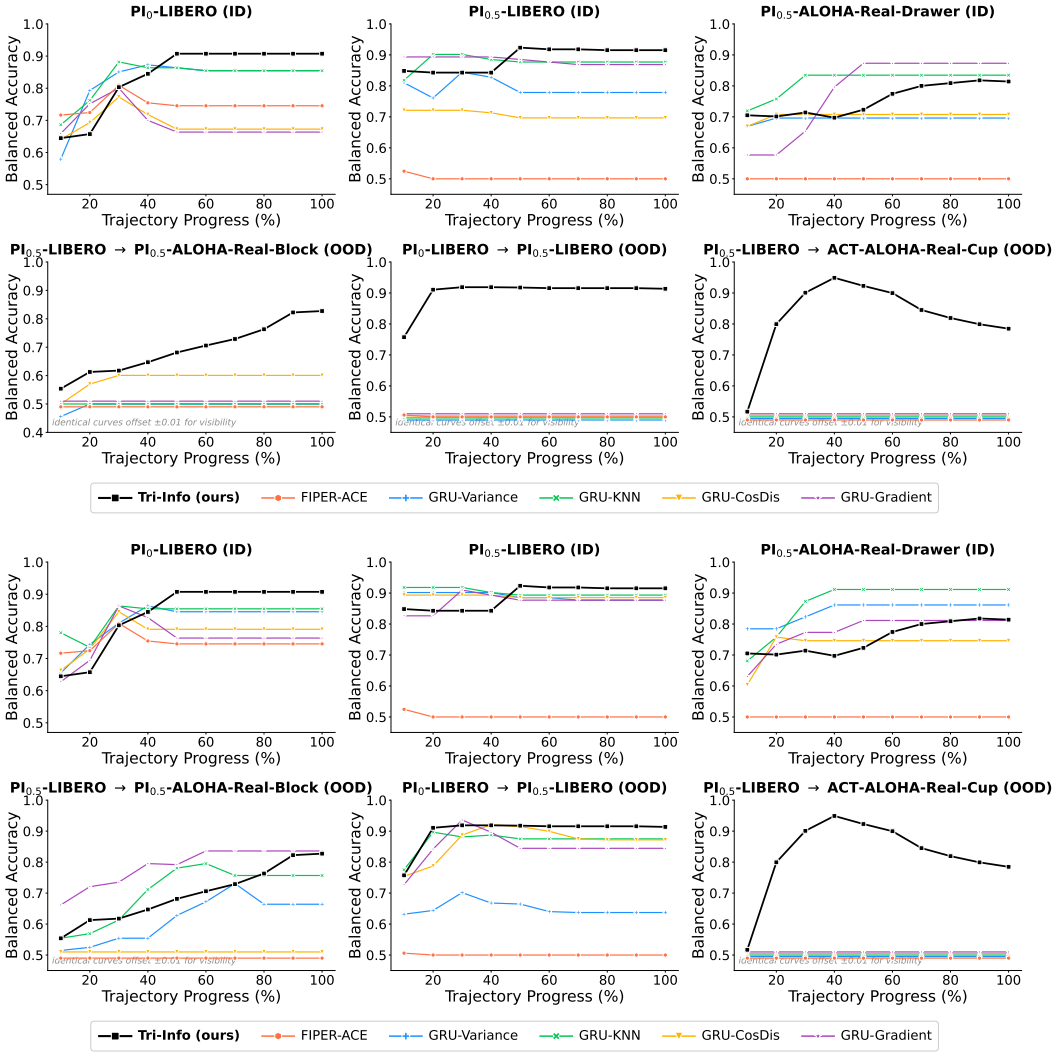

Figure 13: **Comparison against action-based scores.** Balanced accuracy versus trajectory progress for Tri-Info, FIPER-ACE, and the four action-continuity scores, all sharing our GRU pipeline. **Top:** scores computed on the raw action embeddings, where all four at or near chance in every OOD panel. **Bottom:** the same scores on the VAE-matched latent, which recovers accuracy on the cross-environment and cross-model transfers but not on $PI_{0.5}$-LIBERO $\rightarrow$ ACT-ALOHA-real-cup. Flat curves at chance are offset by ± 0.01 for visibility.

Table 9: Full baseline comparison. Trajectory-level AUC (top) and mean balanced accuracy over $t \in \{0.1, \ldots, 1.0\}$ (bottom). "VAE" denotes the dimension-matched variant of each embedding-based detector. "n/a" marks settings where the raw detector cannot be applied because the target latent dimensionality differs from the source. Best per row in bold.

	Score-based			Embedding-based					
				SAFE		Fail-Detect		FIPER	
Setting	Tri-Info	STAC	GRU-STAC	raw	VAE	raw	VAE	raw	VAE
AUC									
PI_0-LIBERO-10	0.70	0.44	0.60	**0.85**	0.74	0.70	0.73	0.63	0.49
$PI_{0.5}$-LIBERO-10	0.86	0.67	0.81	**0.95**	0.90	0.75	0.82	0.77	0.93
$PI_{0.5}$-ALOHA-real-drawer	0.90	0.75	0.85	**0.94**	0.90	0.89	0.72	0.81	0.75
ID average	0.82	0.62	0.76	**0.91**	0.85	0.78	0.76	0.73	0.72
$PI_{0.5}$-LIBERO → $PI_{0.5}$-ALOHA-real-block	0.81	0.65	0.77	0.61	0.86	0.32	0.24	**0.91**	0.51
PI_0-LIBERO → $PI_{0.5}$-LIBERO	**0.82**	0.41	0.52	0.64	0.74	0.69	0.49	0.56	0.77
$PI_{0.5}$-LIBERO → ACT-ALOHA-real-cup	0.89	0.27	**0.94**	n/a	0.79	n/a	0.22	n/a	0.50
OOD average	**0.84**	0.44	0.74	n/a	0.80	n/a	0.32	n/a	0.59
Mean balanced accuracy									
PI_0-LIBERO-10	**0.84**	0.65	0.82	0.82	0.83	0.75	0.54	0.82	0.72
$PI_{0.5}$-LIBERO-10	0.89	0.63	0.79	0.89	0.78	0.85	**0.92**	0.72	0.72
$PI_{0.5}$-ALOHA-real-drawer	0.76	0.55	**0.81**	0.76	0.77	0.75	0.69	0.76	0.70
ID average	**0.83**	0.61	0.81	0.82	0.79	0.78	0.72	0.76	0.71
$PI_{0.5}$-LIBERO → $PI_{0.5}$-ALOHA-real-block	**0.70**	0.49	0.50	0.60	0.53	0.50	0.50	0.50	0.50
PI_0-LIBERO → $PI_{0.5}$-LIBERO	**0.90**	0.50	0.80	0.50	0.81	0.50	0.50	0.50	0.50
$PI_{0.5}$-LIBERO → ACT-ALOHA-real-cup	**0.82**	0.33	0.76	n/a	0.58	n/a	0.50	n/a	0.50
OOD average	**0.81**	0.44	0.69	n/a	0.64	n/a	0.50	n/a	0.50

Table 10: Comparison against action-based scores, mean balanced accuracy over $t \in \{0.1, \ldots, 1.0\}$. All scores share the GRU pipeline, so only the input signal differs. "Raw" uses the action embeddings directly; "VAE-matched" uses the $d_z = 7$ latent, which Tri-Info also uses. FIPER-ACE reads sampled action chunks rather than embeddings and is therefore unaffected by the projection. Best per row in bold.

			Raw embeddings				VAE-matched latent			
Setting	Tri-Info	FIPER-ACE	Var	kNN	CosDis	Grad	Var	kNN	CosDis	Grad
PI_0-LIBERO-10	**0.84**	0.75	0.82	0.83	0.69	0.69	0.81	**0.84**	0.78	0.76
$PI_{0.5}$-LIBERO-10	0.89	0.50	0.79	0.88	0.71	0.88	0.89	**0.90**	0.89	0.87
$PI_{0.5}$-ALOHA-real-drawer	0.76	0.50	0.69	0.82	0.70	0.78	0.84	**0.87**	0.73	0.78
ID average	0.83	0.58	0.77	0.84	0.70	0.78	0.85	**0.87**	0.80	0.80
$PI_{0.5}$-LIBERO → $PI_{0.5}$-ALOHA-real-block	0.70	0.50	0.50	0.50	0.59	0.50	0.62	0.71	0.50	**0.79**
PI_0-LIBERO → $PI_{0.5}$-LIBERO	**0.90**	0.50	0.50	0.50	0.50	0.50	0.65	0.87	0.87	0.85
$PI_{0.5}$-LIBERO → ACT-ALOHA-real-cup	**0.82**	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
OOD average	**0.81**	0.50	0.50	0.50	0.53	0.50	0.59	0.69	0.62	0.71

for Tri-Info, and on the matched latent kNN and variance reach 0.87 and 0.85. FIPER-ACE is the weakest at 0.58 and is unchanged by the projection, since it reads sampled action chunks rather than embeddings. In-domain accuracy therefore does not favour the information-theoretic form: action dispersion in a well-scaled latent already suffices.

The methods separate under shift, and the projection largely determines how far. On raw embeddings, variance, kNN, and gradient fall to chance on every transfer, and cosine dissimilarity rises above chance only cross-environment (0.59), giving OOD averages of 0.50 to 0.53. These scores are expressed in the units of one policy's embedding space, so a threshold fitted on the source does not carry over. The matched latent recovers part of their accuracy, raising the OOD average to 0.71 for gradient and 0.69 for kNN. The recovery comes entirely from the two milder transfers. Cross-environment, gradient reaches 0.79 and kNN 0.71, both above Tri-Info's 0.70. Cross-model, kNN and cosine dissimilarity reach 0.87 and gradient 0.85, all below Tri-Info's 0.90. On the transfer that changes model, embodiment, action dimensionality, and appearance at once, all four return to 0.50 in both representations, while Tri-Info holds 0.82. Tri-Info is therefore not the strongest on

every transfer, but it is the only method in Table 10 that stays above chance on all three, and its OOD average (0.81) exceeds the best alternative (0.71) by 0.10.

We draw two conclusions. The VAE projection accounts for a substantial share of whatever transfer an action statistic achieves, which is why we report both variants rather than attributing transfer to the metric alone. The projection is nevertheless not sufficient: Once the latent is matched, entropy and mutual information remain comparable across domains because they describe the distribution of actions (how probability mass is spread and how strongly actions depend on their history and states).

D.4 Hyperparameter Search

Every method is tuned by an independent grid search, so that no comparison is confounded by unequal tuning effort. Table 15 lists the grids and the selected configurations. Each search runs on the in-domain train/validation split of its own domain, and the selected configuration is applied unchanged to the transfer settings: the cross-model transfer inherits the PI_0-LIBERO configuration, the two ALOHA transfers the $\text{PI}_{0.5}$-LIBERO one. Model hyperparameters are ranked by AUC, and α, which does not enter AUC, by mean balanced accuracy among the AUC-optimal configurations.

D.5 Additional Results and Sensitivity Analyses

This section reports confidence intervals on the OOD results and the sensitivity of Tri-Info to detector capacity and to the flow-matching integration step.

Confidence intervals on OOD. We add trajectory-level 95% CIs to the main OOD and real-world curves. We resample the n test trajectories with replacement, stratified by outcome ($B = 2000$). Each cell reports balanced accuracy with its 95% CI in bracket:

Panel (OOD)	n	20%	40%	60%	80%	100%
PI_0-LIBERO-10 → $\text{PI}_{0.5}$-LIBERO-10	180	0.91 [0.86,0.95]	0.92 [0.89,0.95]	0.92 [0.88,0.94]	0.92 [0.88,0.94]	0.91 [0.88,0.94]
$\text{PI}_{0.5}$-LIBERO-10 → $\text{PI}_{0.5}$-ALOHA-real-block	60	0.61 [0.55,0.69]	0.65 [0.57,0.72]	0.71 [0.63,0.79]	0.76 [0.68,0.85]	0.83 [0.74,0.91]
$\text{PI}_{0.5}$-LIBERO-10 → ACT-ALOHA-real-cup	120	0.80 [0.74,0.86]	0.95 [0.91,0.98]	0.90 [0.85,0.95]	0.82 [0.75,0.88]	0.78 [0.72,0.85]

Detector capacity. To verify that our results do not hinge on detector capacity, we run a post-hoc sweep over the GRU hidden size N and the dropout rate on the two OOD transfers (Table 11). AUC varies within 0.818–0.839 on the cross-model transfer and 0.852–0.910 on the sim-to-real transfer, with no monotone trend in either hyperparameter.

Table 11: Detector capacity sweep. Trajectory-level AUC (mean$_{\pm\text{std}}$ over three seeds) under distribution shift, for joint settings of GRU hidden size N and dropout. Detection is insensitive to both. Best per row in bold.

AUC (OOD) \ (N, dropout)	(32, 0.1)	(64, 0.2)	(128, 0.2)	(256, 0.1)	(256, 0.3)
PI_0-LIBERO-10 → $\text{PI}_{0.5}$-LIBERO-10	$0.818_{\pm.006}$	$0.824_{\pm.002}$	$\mathbf{0.839}_{\pm.005}$	$0.825_{\pm.007}$	$0.830_{\pm.008}$
$\text{PI}_{0.5}$-LIBERO-10 → ACT-ALOHA-real	$0.891_{\pm.014}$	$0.852_{\pm.048}$	$\mathbf{0.910}_{\pm.020}$	$0.888_{\pm.029}$	$0.862_{\pm.016}$

Flow-matching integration step. Flow-matching VLAs have no discrete action decoder, so we read the action embedding $\mathbf{a}_t$ from the velocity field at the last integration step (Section 3). To check that this choice is not critical, we rerun the Tri-Info pipeline on $\text{PI}_{0.5}$-LIBERO, changing only the step $i \in \{0, \ldots, 9\}$ from which $\mathbf{a}_t$ is read (Table 12). mBA stays within 0.862–0.893 (mean 0.881, std 0.011) with no trend across steps, and the default last step is within 0.004 of the best. Detection is therefore insensitive to this choice; we keep the last step, which is closest to the executed action.

Table 12: mBA versus the integration step used for the action embedding ($\text{PI}_{0.5}$-LIBERO). Default in bold; best underlined.

Integration step i	0	1	2	3	4	5	6	7	8	9 (default)
mBA	0.886	0.863	0.889	0.862	<u>0.893</u>	0.880	0.874	0.886	0.883	**0.889**

Table 13: Detector latency grows only mildly with window size W. Per-inference latency (ms), averaged over 10×30 trajectories.

		Window size W				
GPU	Model	10	20	30	40	50
H800	PI_0	2.64	3.14	3.65	4.54	5.22
	$PI_{0.5}$	2.69	3.21	3.72	4.61	5.30
RTX 4090	PI_0	4.08	4.27	4.47	4.79	5.06
	$PI_{0.5}$	4.02	4.22	4.42	4.74	5.01

Running time. We measure the detector's per-inference cost end-to-end, averaged over 10×30 trajectories (Table 13). The cost rises mildly with W and reaches at most 5.30 ms on H800 (5.06 ms on RTX 4090) even at $W=50$, a small fraction of a typical VLA inference step, so the detector adds negligible overhead and does not perturb the control loop.

Threshold calibration: α trades timeliness for accuracy. The significance level α sets the detector's tolerance to transient anomalies (Table 14). Because the threshold is the $(1-\alpha)$ quantile of successful-trajectory scores, and successful trajectories that recover from mid-execution perturbations still carry the label $f^{\xi}=0$, a smaller α raises the threshold and lets more transient excursions pass without an alarm. Lowering α from 0.10 to 0.01 thus raises the true-negative rate from 0.584 to 0.906, while the true-positive rate stays above 0.95. The cost is timeliness: the first alarm moves from 5.5% to 16.0% of the trajectory. Even at the strictest setting, the detector reaches 0.932 balanced accuracy after only 16% of execution, and practitioners can tune α to their risk profile.

Table 14: α controls tolerance for transient anomalies. TNR: fraction of successful trajectories not falsely flagged.

α	TPR	TNR	Bal. Acc	Det.%
0.010	.958	**.906**	**.932**	16.0
0.050	.995	.736	.865	9.6
0.100	.995	.584	.789	5.5

E More Explorations

Can the method work on non-VLA policies? Our formulation assumes only a closed loop over $(\mathbf{S}_t, \mathbf{A}_t, \mathbf{S}_{t+1}, \mathbf{A}_{t+1})$, not a vision-language backbone. We therefore test it on **BC-RNN** (Mandlekar et al., 2021), an LSTM behavioral-cloning policy, trained on two datasets that each provide a clean and a noisy demonstration pool: (a) the 20%-spoiled *Can* split of Layman V1.0 (Sojib & Begum, 2024), where the clean pool contains 120 expert demonstrations (277 steps; *Baseline* in the original release) and the noisy pool adds 30 lay-user demonstrations (503 steps; *Type-1*), yielding success rates (SR) of 0.768 and 0.683; (b) Robomimic Multi-Human (Mandlekar et al., 2021), where the clean pool contains *better* demonstrations (143 steps) and the noisy pool contains *worse* ones that deviate but still succeed (304 steps), yielding SR 0.847 and 0.577. We sample 900 rollouts per policy for (a) and 300 for (b). For in-domain evaluation we use a 70/30 split; for transfer we train on all source rollouts and test on all target rollouts. We report BA at 20% intervals of trajectory progress, mBA over $t \in \{0.1, \ldots, 1.0\}$, and trajectory-level AUC on $P^{\xi}_{\max}$, each averaged over three seeds (Table 16). Tri-Info reaches an AUC of 0.997 and 0.999 in domain, remains reliable on the noisy-data policies (0.987 and 0.899), and transfers from clean- to noisy-policy rollouts (0.972 and 0.943). **Tri-Info is thus not specific to VLAs and applies to other closed-loop policies.**

Does selecting actions with better information raise the success rate? We test Tri-Info as a score rather than as an alarm. $\pi_{0.5}$ draws its action chunk from a flow-matching ODE, so one observation admits many distinct chunks; at every replan point we therefore sample $K = 32$ candidates, score each of them, and execute the best one. Nothing is retrained and no candidate is rejected and resampled: the rule changes only which of the chunks the sampler has already produced gets executed. We compare rules *Tri-Info* executes the candidate with the lowest predicted failure probability. *No selection* draws a single sample, the default deployment. *Min-* and *max-entropy* pick the candidate with the lowest or highest entropy, which is information-theoretic but uses neither a learned detector nor the history.

Table 15: **Hyperparameter search.** (a) The grid swept for each method. (b) The configuration selected in each of the three domains on which the search is run. h: hidden size; E: training epochs; W: window length; α: significance level; $\mathcal{A} = \{0.001, 0.05, 0.10, 0.15\}$. Methods in the upper block read model embeddings and are swept separately on the raw embeddings (raw) and on the VAE-matched latent (VAE); methods in the lower block read executed or sampled actions and are swept once, so their selected configurations hold under both representations. FIPER-ACE fixes $W = 30$, stride 1, and the translational dimension set.

(a) Search grids

Method	Grid
Representation-dependent (swept once per representation)	
SAFE	backbone $\times E \times \alpha$: {mlp, lstm} $\times \{50, 100\} \times \mathcal{A}$
Fail-Detect	$h \times \alpha$: $\{32, 64, 256\} \times (\mathcal{A} \cup \{0.20\})$
FIPER	$\alpha\times$ combine $\times\, w_{\text{act}}$: $(\mathcal{A} \cup \{0.25\}) \times$ {and, or} $\times \{10, 15, 20, 45\}$
GRU-Variance / kNN / CosDis / Gradient	$h \times E \times W \times \alpha$: $\{32, 64, 256\} \times \{50, 100\} \times \{20, 30\} \times \mathcal{A}$
Representation-independent (swept once)	
STAC	score mode $\times\, \alpha$: {cumulative, max} $\times\, \mathcal{A}$
GRU-STAC	$h \times E \times W \times \alpha$: $\{32, 64, 256\} \times \{50, 100\} \times \{20, 30\} \times \mathcal{A}$
FIPER-ACE	aggregation $\times\, h \times E$: 4 settings $\times \{32, 64, 256\} \times \{50, 100\}$
Tri-Info	$h \times E \times \alpha$: $\{32, 64, 256\} \times \{50, 100\} \times \mathcal{A}$

(b) Selected configurations

			Domain		
Method	**Format**	**Repr.**	PI_0-LIBERO-10	$\text{PI}_{0.5}$-LIBERO-10	$\text{PI}_{0.5}$-ALOHA-real-drawer
SAFE	(backbone, E, α)	raw	lstm, 100, .05	mlp, 50, .15	mlp, 50, .10
		VAE	lstm, 100, .10	mlp, 100, .15	lstm, 100, .05
Fail-Detect	(h, α)	raw	256, .15	256, .20	256, .15
		VAE	256, .15	256, .15	256, .10
FIPER	$(\alpha, \text{combine}, w_{\text{act}})$	raw	.05, or, 15	.05, or, 15	.05, or, 45
		VAE	.25, or, 45	.05, or, 10	.10, or, 10
GRU-Variance	(h, E, W, α)	raw	256, 100, 20, .05	256, 100, 20, .05	256, 50, 20, .15
		VAE	256, 100, 20, .05	32, 50, 20, .15	64, 50, 20, .05
GRU-kNN	(h, E, W, α)	raw	256, 100, 20, .05	256, 100, 30, .05	256, 50, 30, .15
		VAE	256, 100, 20, .05	64, 100, 20, .05	64, 50, 30, .15
GRU-CosDis	(h, E, W, α)	raw	64, 100, 30, .05	256, 100, 20, .05	64, 100, 30, .05
		VAE	256, 50, 30, .15	64, 50, 20, .05	256, 100, 30, .05
GRU-Gradient	(h, E, W, α)	raw	32, 50, 20, .15	64, 50, 20, .05	256, 50, 20, .15
		VAE	64, 100, 30, .05	32, 50, 20, .15	256, 100, 20, .05
Tri-Info	(h, E, α)	both	32, 100, .10	256, 100, .10	32, 50, .05
STAC	(mode, α)	both	cumulative, .10	max, .10	cumulative, .10
GRU-STAC	(h, E, W, α)	both	32, 100, 20, .05	256, 100, 20, .15	32, 100, 20, .05
FIPER-ACE	(aggregation, h, E)	both	window-30, 256, 50	raw, 256, 100	raw, 32, 50

Scoring a candidate would normally require the VAE latents used in the main results, so we re-fit the detector on the raw action and proprioceptive state the controller already holds: a candidate is scored by building the window its execution would produce, re-estimating the three metrics on that window, and advancing the GRU one step without committing, which adds no policy forward pass on top of the $K = 32$ sample. We evaluate a $\pi_{0.5}$ policy fine-tuned on relabelled *Can* demonstrations in robosuite PICKPLACECAN, with a 400-step horizon and a replan every 5 steps. All rules run on the same 248 initial states, so every comparison is paired and is tested with an exact McNemar test against C0, a pre-registered set of 200 single-sample rollouts of the same checkpoint (SR 0.670). Tri-Info selection attains the highest success rate of any rule, 0.710 against 0.670 for C0 (Table 17).

Table 16: Tri-Info on the non-VLA BC-RNN policy under clean and noisy demonstrations. BA at each trajectory progress, mBA, and trajectory-level AUC; mean over three seeds.

BC-RNN policy	20%	40%	60%	80%	100%	mBA	AUC
(a) Layman V1.0							
Clean→Clean	0.593	0.774	0.870	0.928	0.923	0.794	0.997
Noise→Noise	0.584	0.665	0.779	0.869	0.888	0.739	0.987
Clean→Noise	0.605	0.705	0.784	0.841	0.840	0.738	0.972
(b) Robomimic Multi-Human							
Clean→Clean	0.688	0.920	0.932	0.956	0.956	0.867	0.999
Noise→Noise	0.558	0.636	0.663	0.717	0.746	0.656	0.899
Clean→Noise	0.708	0.763	0.741	0.741	0.702	0.710	0.943

Table 17: **Tri-Info as an inference-time action selector.** At each replan point $K = 32$ candidate action chunks are sampled from $\pi_{0.5}$ and one is executed. SR is over all 248 initial states; Δ, the win counts (rule/C0 among the discordant pairs) and the exact McNemar p are over the 200 initial states shared with the pre-registered single-sample reference C0.

Selection rule	Criterion	SR	Δ vs. C0	McNemar p
C0 (pre-registered)	no selection, K=1	0.670	—	—
Tri-Info	$\arg\min_k \hat{p}^{(k)}$(failure)	**0.710**	+0.060 (35/23)	0.148
No selection	K=1	0.653	−0.015 (29/32)	0.80
Min-entropy	$\arg\min_k \hat{H}(a^{(k)})$	0.621	−0.050 (23/33)	0.23
Max-entropy	$\arg\max_k \hat{H}(a^{(k)})$	0.585	−0.080 (24/40)	0.060