IF AT FIRST YOU DON'T SUCCEED, BACKTRACK: LIGHTWEIGHT RECOVERY FOR FROZEN VLAS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models perform well on robotic manipulation, yet remain brittle once their own actions carry them into states their training data barely covers. Recovering from such failures usually requires recovery demonstrations, policy retraining, or a separate model at inference time. We introduce **Backtrack**, a lightweight failure detection and recovery mechanism for frozen VLAs that needs none of these. A small failure-prediction head, trained only on outcome-labeled rollouts of the frozen policy, reads the VLA's existing internal representations to estimate whether the current episode will fail. When the predicted failure probability exceeds a threshold, a controller pauses the policy, retraces the arm's recently achieved motion, and hands control back to the unchanged policy so it can retry from an earlier configuration. The head adds only 0.48–0.88% to the policy's parameter count. We evaluate Backtrack on LIBERO with three VLAs spanning two action-generation families: SmolVLA and π0.5, which use flow-matching action experts, and the autoregressive π0-FAST. On LIBERO-Long, the suite of long-horizon multi-step tasks and the one on which all three policies fail most often, Backtrack reduces each policy's failure rate by 19–48%, raising success by up to 7.4 percentage points; on the three shorter suites, where control success is already 85–99%, its effect is smaller. On MetaWorld, all twelve controller configurations tested with SmolVLA improve on the unassisted policy.

1 INTRODUCTION

Recent advances in Vision-Language Models (VLMs) have spurred the development of Vision-Language-Action (VLA) models (Black et al., 2025a; Pertsch et al., 2025; Shukor et al., 2025). These models typically use pretrained VLMs as a backbone, leveraging their visual and language representations to condition the generation of robot actions. Yet even large VLAs trained on diverse, multi-embodiment robotics datasets remain brittle outside their training distribution: modest changes to camera viewpoint, robot initialization, or scene composition can sharply reduce success rates (Fei et al., 2026). This challenge is particularly relevant for compact VLAs such as SmolVLA, whose VLM backbone benefits from broad vision-language pretraining, but whose robotics training consists of fewer than 30,000 episodes of community-contributed demonstrations collected primarily on a single robot embodiment (Shukor et al., 2025).

Out-of-distribution states are not only something a VLA encounters; they are also something it can create. Robot imitation datasets are built primarily from successful demonstrations and contain comparatively little data showing how to recover once something has gone wrong (Dai et al., 2025; Shukor et al., 2025). For instance, when a VLA drops an object, knocks something over, or otherwise makes a mistake, it can therefore place itself in a state poorly represented in its training data. Simply allowing more execution time leaves most failures unresolved: on LIBERO-Long, doubling the episode step budget increases success by only 2.3–3.7 percentage points across the three evaluated VLAs (Appendix A.3).

One remedy is to give the policy the data it is missing: collect demonstrations of failures and their recoveries, then train on them (Dai et al., 2025). But collecting recovery data is expensive, and the cost may need to be paid again for every new embodiment and task. Instead, we ask whether part of the distribution shift can be reversed directly. After a mistake, both the scene and the robot

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

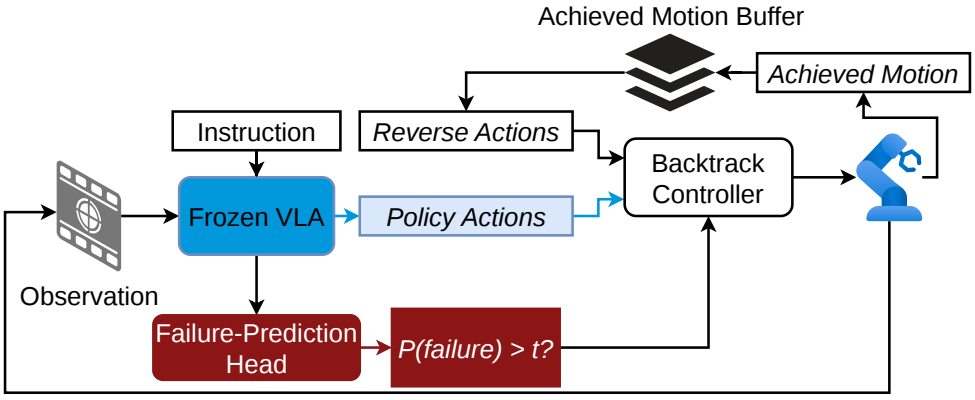


Figure 1: **Overview of Backtrack.** A small failure-prediction head reads features already computed by the frozen VLA and estimates the probability that the current episode will fail. When the predicted failure probability exceeds the threshold t, the Backtrack controller suppresses the policy's proposed actions and instead retraces recent achieved arm motion stored in a buffer. Otherwise, the policy's actions are executed normally. The resulting state is observed by the unchanged VLA and the control loop continues. Achieved motion is added to the buffer during normal policy execution; motion executed during backtracking is not. The diagram abstracts over differences in action generation between the evaluated VLAs.

may have changed: objects may have been displaced, while the arm has moved into an unfamiliar configuration. Changes to the scene are difficult to undo, since restoring a displaced object is itself a manipulation problem. The arm configuration, however, is under our control. This suggests a simple recovery strategy: retrace the arm's motion along the path that led to failure, in the hope of returning toward an earlier configuration from which the unchanged policy can successfully retry.

In this work, we introduce Backtrack, a lightweight failure detection and recovery mechanism for frozen VLAs (Figure 1). Similar to SAFE (Gu et al., 2025), a small failure prediction head reads features the policy has already computed and estimates the probability that the current episode will end in failure; when that probability exceeds a threshold, a controller retraces the arm's recent motion and returns control to the policy. The only supervision Backtrack needs is whether autonomous rollouts of the frozen policy succeed or fail: no teleoperated recovery demonstrations, no policy retraining, and no second model at inference. The head adds just 0.48–0.88% to the policy's parameters.

We evaluate Backtrack on three policies with two different action-generation mechanisms: SmolVLA (Shukor et al., 2025) and π0.5 (Black et al., 2025a) use flow-matching action experts, while π0-FAST (Pertsch et al., 2025) generates discrete action tokens autoregressively. On LIBERO-Long (Liu et al., 2023a), the suite on which all three policies fail most often, Backtrack reduces failure rates by 19% to 48%; averaged over all four LIBERO suites, the reduction is 20–26%. On MetaWorld (Yu et al., 2020), where we evaluate SmolVLA on a second benchmark with a different embodiment, all twelve controller configurations we test improve on the unassisted policy.

2 Related Work

2.1 Vision-Language-Action Models

VLAs adapt a pretrained vision-language backbone to output robot actions, inheriting its scene and instruction understanding (Brohan et al., 2023; Kim et al., 2024). They differ in how actions are produced. One family attaches a separate action expert trained by flow matching or diffusion and cross-attending to the backbone, as in SmolVLA (Shukor et al., 2025) and π0.5 (Black et al., 2025a). The other discretizes actions into tokens that the backbone emits autoregressively, as in π0-FAST (Pertsch et al., 2025). A parallel line of work develops compact VLAs and backbones for more

Table 1: Parameter count of the failure prediction head, relative to the frozen policy it attaches to. All three heads read the `post` feature tap.

POLICY	PARAMS	HEAD PARAMS	HEAD / POLICY
SmolVLA	450.0M	3.97M	0.88%
π0-FAST	3450.0M	17.35M	0.50%
π0.5	3616.8M	17.35M	0.48%

efficient deployment (Shukor et al., 2025; Wang et al., 2025; Marafioti et al., 2025). Backtrack attaches to a frozen checkpoint and is agnostic to the action-generation architecture; we evaluate it on both families.

2.2 Failure Detection for VLAs

Several lines of work detect signs of failure during policy execution. One uses a separate vision-language model as a judge, either to classify success (Du et al., 2023) or to reason about what went wrong (Duan et al., 2025; Liu et al., 2023b). Another uses signals derived from the policy itself: Agia et al. (2025) monitor consistency and progress at runtime, while SAFE (Gu et al., 2025) trains a lightweight classifier on the policy's internal features and shows transfer across tasks and policies, including π0-FAST, but stops at detection. Backtrack builds on this latter approach: we use the prediction to trigger a lightweight recovery controller, measure what that intervention is worth across three policies, and test whether the effect transfers to a second benchmark.

2.3 Failure Recovery for VLAs

Detection alone does not change the outcome of an episode; something must be done once a failure is predicted. Retrying after a detected failure is not new. Bellman-Guided Retrials (Du et al., 2024) monitors progress with a value function trained on demonstrations and resamples a strategy from the base policy when progress stalls, biasing away from behaviors that have recently failed. Other work prompts a language model to reason its way to a corrective plan (Lin et al., 2025; Liu et al., 2023b). These approaches introduce additional learned components or demonstrations to decide how to recover, whereas they do not provide a task-agnostic mechanism for returning the robot to a prior configuration while leaving the base policy unchanged. Backtrack instead uses a task-agnostic recovery primitive that requires no recovery demonstrations or learned recovery policy: the controller retraces the arm's recent motion and then returns control to the unchanged VLA.

3 Method

3.1 Failure Prediction Head

We attach a *Failure Prediction Head* (FPH) to the frozen policy. The FPH is a multi-head attention pooling (MAP) layer over the policy's own token features, followed by a three-layer feed-forward network (FFN) that outputs a single scalar: the predicted probability that the current episode will end in failure.

Where the features come from. The FPH consumes activations the policy already computes during normal action generation, so it requires no additional forward pass through the backbone. We consider three feature taps: the prefix embeddings the VLM takes as *input* (`pre`), the VLM's *output* representation of that same prefix (`post`), and `post` concatenated with the action expert's own hidden states (`post_expert`). Unless stated otherwise, all reported results use `post`.

π0-FAST has no separate action expert: the VLM emits action tokens itself. Its `pre` tap is the prefix before the VLM, as for the other two policies, and its `post` tap is the output of the prefill, the VLM's representation of that prefix immediately before the first action token is emitted.

Table 2: Rollout data behind each failure prediction head. The Episodes column gives the number of rollouts of the frozen policy pooled across all four LIBERO suites; the failure rate is the fraction of those episodes that ended in failure. Sampled frames is the number drawn from that pool to train and validate the head.

POLICY	EPISODES	FAILED	SAMPLED FRAMES
SmolVLA	20,000	3,464 (17.3%)	100,000
π0-FAST	7,953	812 (10.2%)	100,000
π0.5	14,000	676 (4.8%)	100,000

Parameter overhead. The head is evaluated once per action chunk and reuses activations already present in the forward pass, so its only marginal cost is the MAP layer and the FFN. Its size depends on the width of the features it reads rather than on the size of the policy, so the relative overhead is largest on the smallest policy: it adds at most **0.88%** of the policy's parameters, on the 450M SmolVLA, and as little as 0.48% on the 3.6B π0.5 (Table 1).

Training. The head is trained on rollouts of the frozen policy, with each frame labeled by the outcome of the episode it belongs to. Rollouts are collected across all four LIBERO suites, and we sample with equal weight over the eight (suite, label) cells rather than from the natural distribution. This matters because failures are distributed very unevenly across suites: for π0-FAST, LIBERO-Long supplies 56% of all failure frames and LIBERO-Object only 7%, and the imbalance is larger still for SmolVLA. Sampling from the raw distribution would give one head per policy that is effectively specialized to the suite its policy fails on most, whereas we want a single head that serves all four. The amount of rollout data available differs across policies; Table 2 summarizes the rollout pools used for each head.

We train each head for 20,000 steps with Adam, batch size 32, learning rate 10^{-4} and weight decay 10^{-5}, with no additional positive-class weighting on top of the balanced sampler.

3.2 BACKTRACKING

Trigger. Once the predicted failure probability exceeds a threshold, the policy is paused and the controller replays previously achieved arm motion in reverse, recomputing the failure probability at the same interval at which action chunks would otherwise be generated. In the fixed variant, the controller retreats for a predetermined number of steps; in the adaptive variant, it continues until the failure probability falls below a second, lower threshold, subject to minimum and maximum retreat lengths. The intuition is that retracing the arm may undo part of the deviation that led to failure, returning it toward a state better represented in the policy's training data. Backtracking reverses the arm's motion without explicitly restoring the scene; displaced objects may remain displaced, and contact during reversal may move them further.

Parameters. The controller has seven parameters. The *threshold* t is the predicted failure probability that triggers a retreat. A *fixed* retreat reverses a set number of steps s; an *adaptive* retreat continues until the probability falls below the *restart threshold* a, bounded by $s_{\min}$ and $s_{\max}$ steps. The *cooldown* cd is the number of steps after a retreat during which the trigger is suppressed, so the controller does not immediately re-fire on the state it has just left. The *retreat cap* K limits how many retreats may be initiated per episode. The configurations selected for the main experiments (Appendix A.1) are: SmolVLA, adaptive with t=0.97, a=0.2, 50–150 steps; π0.5, fixed with t=0.99, s=50; and π0-FAST, adaptive with t=0.90, a=0.3, 25–100 steps. All three use cd=50 and K=3.

What is replayed. During normal operation we append every executed motion to a buffer. These are not the actions commanded by the policy. The policy emits deltas relative to the arm's current pose, but if the arm cannot move as commanded, for instance because the motion is blocked by an object or by the work surface, recording the command would cause the reversal to undo a motion that never happened. We therefore record the *achieved* delta, that is, the difference between the

arm's current and previous pose. The gripper is replayed with the arm: on LIBERO, where gripper commands are absolute, the recorded commands are re-issued in reverse order, so retreating past a grasp re-opens the gripper; on MetaWorld, where they are incremental, they are negated.

The return is approximate. Reversing recorded deltas does not restore the arm exactly. Although we record achieved rather than commanded motion, the reversed deltas are themselves executed approximately because of the embodiment's tracking dynamics. The arm therefore returns to a neighborhood of the earlier state rather than to the state itself.

4 Experimental Setup

4.1 Benchmark and protocol

We evaluate primarily on LIBERO (Liu et al., 2023a), using its four ten-task suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. To check that the method is not specific to one benchmark we additionally evaluate on MetaWorld (Yu et al., 2020), a separate manipulation benchmark with a different simulator, task set, and embodiment. The MetaWorld experiments use SmolVLA with the `pre` feature tap.

We build on the LeRobot library (Cadene et al., 2024), which provides the policy implementations, the LIBERO environment wrappers, and the evaluation harness. We implement the failure prediction head, the backtracking controller, and our evaluation extensions in a fork of LeRobot.

Protocol. Our main LIBERO evaluation uses five seeds and 50 episodes per task. We use one controller configuration per policy, selected on LIBERO-Long and transferred unchanged to the other three suites; the grid and selection procedure are given in Appendix A.1. Control and treatment use the same step budget, and retreat steps are charged against it. We report uplift, the treatment minus control success rate in percentage points, paired by seed, with 95% confidence intervals over the five per-seed differences.

Step budget. We evaluate at twice the per-suite step limit set by LeRobot's LIBERO wrappers, giving 560 steps for LIBERO-Spatial and LIBERO-Object, 600 for LIBERO-Goal, and 1040 for LIBERO-Long. The default limits are set just above the longest training demonstration in each suite, leaving as few as fifteen steps of slack on LIBERO-Long. This leaves little room for a policy to deviate and subsequently recover before reaching the episode limit. We double these limits because a retreat is charged against the same budget, and we apply the same budget to control and treatment throughout. The original benchmark does not specify per-suite episode limits, using a single 600-step cap for its own training-time evaluation (Liu et al., 2023a).

4.2 Policies

SmolVLA. SmolVLA (Shukor et al., 2025) is a compact VLA that truncates SmolVLM2 (Marafioti et al., 2025) and adds a flow-matching action expert, which cross-attends to the VLM's KV cache. In their LIBERO evaluation, Shukor et al. (2025) use 10 flow-matching steps and regenerate an action chunk after every executed action. We change two inference settings. First, we use only 2 flow-matching steps, since our ablation in Appendix A.2 shows this only slightly degrades performance on LIBERO and improves it on MetaWorld (Yu et al., 2020). Second, we execute 10 actions from each generated action chunk before regenerating, which substantially reduces the number of policy forward passes required during evaluation.

π0.5. π0.5 is the largest policy we evaluate, with 3.6B parameters (Black et al., 2025a). Like SmolVLA, it uses a VLM backbone with a flow-matching action expert, but at roughly eight times the parameter count.

π0-FAST. π0-FAST is the π0 VLA (Black et al., 2025b) with its flow-matching action expert replaced by autoregressive decoding of discrete action tokens (Pertsch et al., 2025). Unlike the other two policies we evaluate, it has no separate action expert: the VLM emits the action tokens itself.

Table 3: Recovery on the long-horizon LIBERO-Long suite, and aggregated over all four LIBERO suites. Five seeds (1000–5000), 50 episodes per task. Uplift is in percentage points and intervals are 95% confidence intervals over the five per-seed differences. Recovery rate is the uplift divided by the base failure rate, measuring the net relative reduction in failures.

SUITE	POLICY	CONTROL	UPLIFT	95% CI	RECOVERY
LIBERO-Long	SmolVLA	61.00	$+7.40$	$[+4.52, +10.28]$	19.0%
	π0.5	94.72	$+1.60$	$[+0.06, +3.14]$	30.3%
	π0-FAST	85.32	$+7.08$	$[+5.91, +8.25]$	48.2%
all four suites	SmolVLA	83.26	$+3.40$	$[+2.04, +4.76]$	20.3%
	π0.5	96.98	$+0.79$	$[+0.05, +1.53]$	26.2%
	π0-FAST	91.31	$+1.70$	$[+0.97, +2.43]$	19.6%

5 RESULTS

5.1 RECOVERY

Table 3 reports the main result. On LIBERO-Long, Backtrack reduces the failure rates of SmolVLA, π0.5 and π0-FAST by 19.0%, 30.3% and 48.2%, respectively. Every 95% confidence interval excludes zero. Aggregated over all four LIBERO suites, the three policies sit much closer together, at 20.3%, 26.2% and 19.6%, again with every interval excluding zero.

The spread across policies is much wider on LIBERO-Long, at 19–48%, than over the benchmark as a whole, at 19.6–26.2%, and the ordering is not preserved between the two: π0-FAST has the largest reduction on the long-horizon suite and the smallest reduction overall. Recovery rate thus depends on the evaluation suite; these three policies do not establish a relationship between recovery rate and base success rate.

Selection. Each policy's configuration was selected on LIBERO-Long using seeds 1000–3000, where Backtrack has the most room to improve performance, and transferred unchanged to the other three suites. Seeds 4000–5000 played no part in selection; on them, the selected configurations give $+7.20$ pp for SmolVLA (vs. $+7.40$ pooled), $+0.70$ pp for π0.5 (vs. $+1.60$) and $+7.40$ pp for π0-FAST (vs. $+7.08$). For π0.5 and π0-FAST, selection was run on a different feature tap from the one reported (`post_expert` and `pre`, respectively), and the configurations were carried over to `post` unchanged; subsequent `post`-head comparisons found no better configuration among those tested. Appendix A.1 gives the complete selection procedures, configuration grids, and post-selection checks.

5.2 WHERE RECOVERY PAYS

Table 4 breaks the aggregate down by suite. The three short suites sit closer to ceiling, between 84.6% and 99.2% control success, leaving less room for improvement. The long-horizon suite provides the largest uplift for all three policies.

A simple accounting identity illustrates the tradeoff. Let S and F be the baseline success and failure proportions. Recovering a fraction c of failures while disrupting a fraction b of successes changes the success rate by $cF - bS$, or $100(cF - bS)$ percentage points. For $c > 0$, this is positive when $b/c < F/S$. On π0.5's LIBERO-Object, S/F is 124: disrupting just 1% of baseline successes would outweigh recovering every baseline failure. On LIBERO-Long for SmolVLA, S/F is only 1.6, allowing a larger disruption rate for the same fraction of failures recovered.

This motivates selecting configurations on LIBERO-Long, where baseline failure rates are highest. However, b and c may also vary across suites, and the configurations are transferred unchanged. These results therefore do not isolate the effects of baseline failure rate, task characteristics, and configuration selection.

Table 4: Per-suite control success rate and uplift, five seeds, 50 episodes per task.

	SPATIAL		OBJECT		GOAL		LONG	
POLICY	ctrl	upl	ctrl	upl	ctrl	upl	ctrl	upl
SmolVLA	84.64	+5.12	94.72	−0.20	92.68	+1.28	61.00	+7.40
π0.5	98.24	+0.16	99.20	+0.16	95.76	+1.24	94.72	+1.60
π0-FAST	90.92	−0.36	96.80	−0.20	92.20	+0.28	85.32	+7.08

5.3 DETECTION

Across 10,000 Backtrack episodes per policy, spanning all four suites and five seeds, success conditioned on no trigger is 99.71% for SmolVLA, 99.99% for π0.5 and 99.94% for π0-FAST: 29 failures occur without a trigger in 30,000 episodes. These comprise 23 SmolVLA failures on LIBERO-Long, one π0.5 failure on LIBERO-Goal, and five π0-FAST failures on LIBERO-Object. Among triggered episodes, the mean first-trigger time ranges from 11.0% to 22.2% of the budget across the twelve policy-by-suite cells, leaving about 810–870 steps on LIBERO-Long. These results show that few failures go unflagged; they do not by themselves establish how often the intervention was necessary.

Offline discrimination. For π0.5, we also score the head offline on 2,772 held-out rollouts of the frozen policy (135 failures), using a sample of frames from each rollout. It separates failing from successful episodes with an episode-level AUROC of 0.977 (frame-level 0.908). At the deployed threshold $t = 0.99$, it flags 80% of failing episodes at a 2.0% false-alarm rate; lowering the threshold to $t = 0.9$ raises detection to 96% but triples the false-alarm rate to 6.1%. The `pre` and `post_expert` taps reach similar AUROCs (0.980 and 0.987), consistent with the small closed-loop differences between taps (Section 5.5).

5.4 A SECOND BENCHMARK

The results above are all on LIBERO with the `post` feature tap. As a second-benchmark test, we evaluate Backtrack on MetaWorld with SmolVLA and the `pre` feature tap, using a head trained on rollouts from MetaWorld's medium-difficulty tasks and evaluated on all 50 tasks.

SmolVLA's control success rate on MetaWorld is 72.32%. We swept twelve controller configurations, five seeds each. *Every one of them improves on control*, with observed improvements ranging from +2.44 to +5.20 percentage points; eleven of the twelve seed-paired 95% confidence intervals exclude zero (Appendix Table 14). The largest observed improvement comes from a fixed 150-step retreat with a 100-step cooldown at $t = 0.9$, reaching 77.52% success and a recovery rate of 18.8%.

5.5 ABLATIONS

Which features to read. On both flow-matching policies, we find no resolvable benefit from adding the action expert's hidden states to the VLM representation (Appendix A.5). On π0.5, where we evaluate this comparison under the full evaluation protocol, `post` achieves 96.32% versus 96.60% for `post_expert`, a paired difference of −0.28 pp with a 95% interval of $[-0.97, +0.41]$. Since `post` is available before the action chunk is generated and can therefore support parallel failure prediction and action generation, we use it for the main LIBERO evaluations. On π0-FAST, which has no separate action expert, the `post`-minus-`pre` difference is likewise unresolved (+1.40 pp, 95% interval $[-3.0, +5.8]$).

Threshold and retreat rule. On π0.5, Table 5 sweeps the trigger threshold under both fixed and adaptive retreat. The headline configuration, a fixed 50-step retreat with a 50-step cooldown at $t = 0.99$, has the highest observed success rate at 96.32%, followed by 96.24% at $t = 0.97$.

Fixed and adaptive retreat behave differently depending on the trigger threshold. At the lower thresholds, fixed retreat is clearly better: adaptive minus fixed is −2.76 pp at $t = 0.90$ and −2.00 pp at $t = 0.95$, with both intervals excluding zero. At $t = 0.99$ and $t = 0.995$, however, the difference

Table 5: Threshold and retreat rule on $\pi 0.5$, LIBERO-Long, `post` head, five seeds, 50 episodes per task, paired by seed. Control success is 94.72%, a failure rate of 5.28%. The first two blocks sweep the trigger threshold under each retreat rule; the third varies retreat depth at t=0.99. The adaptive rule retreats between 50 and 150 steps, stopping once the predicted failure probability falls below 0.2.

CONFIGURATION	SR	UPLIFT	95% CI	RECOVERY
Threshold, fixed retreat (s=50, cd=50)				
t=0.90	95.56	+0.84	$[+0.11, +1.57]$	15.9%
t=0.95	95.00	+0.28	$[-1.24, +1.80]$	5.3%
t=0.97	96.24	+1.52	$[+0.02, +3.02]$	28.8%
t=0.99 (headline)	96.32	+1.60	$[+0.06, +3.14]$	30.3%
t=0.995	95.56	+0.84	$[-0.36, +2.04]$	15.9%
Threshold, adaptive retreat (50–150 steps, cd=50)				
t=0.90	92.80	−1.92	$[-3.14, -0.70]$	−36.4%
t=0.95	93.00	−1.72	$[-3.92, +0.48]$	−32.6%
t=0.99	95.60	+0.88	$[-0.81, +2.57]$	16.7%
t=0.995	95.32	+0.60	$[-2.06, +3.26]$	11.4%
Retreat depth, fixed (t=0.99)				
s=50, cd=50 (headline)	96.32	+1.60	$[+0.06, +3.14]$	30.3%
s=100, cd=100	95.44	+0.72	$[-0.59, +2.03]$	13.6%

is no longer resolvable. Notably, adaptive retreat at $t = 0.90$ is the only configuration in the sweep that performs significantly worse than control, at -1.92 pp with $[-3.14, -0.70]$, while fixed retreat at the same threshold significantly improves over control, at $+0.84$ pp with $[+0.11, +1.57]$.

One possible explanation is that the two rules differ in how much of the episode budget they consume when the detector fires frequently. At a low threshold, an adaptive retreat of up to 150 steps can spend substantially more of the budget per trigger than a fixed 50-step retreat. Fewer triggers at higher thresholds could reduce this cost. The observed narrowing of the gap is consistent with this explanation, but does not isolate it from the effects of intervention timing or state.

Controller sensitivity across policies. Table 6 repeats the controller ablations on SmolVLA, where the failure rate is substantially larger than on $\pi 0.5$ (39.00% versus 5.28%). Every configuration tested improves over control. The threshold sweep peaks at the reported $t = 0.97$, but the response is relatively broad: thresholds from 0.90 to 0.99 yield uplifts between $+4.60$ and $+7.40$ pp. Of the alternative thresholds, only $t = 0.995$ resolvably underperforms the headline configuration, while still yielding a $+4.00$ pp uplift over control.

On SmolVLA at $t = 0.97$, adaptive retreat has a 2.24 pp higher mean success rate than fixed 50-step retreat, but the 95% interval $[-0.21, +4.69]$ includes zero. Fixed retreat performs better on $\pi 0.5$ at lower thresholds; at higher thresholds, its difference from adaptive retreat is unresolved. No single retreat rule performs best across the tested policies and thresholds.

Number of retreats. On $\pi 0.5$, varying the retreat cap from one to five yields no resolved difference in success (Appendix A.7). The mean rates span 0.60 pp, but the comparison uses only three seeds. It does not establish equivalence between caps or how much recovery is attributable to the first attempt.

6 Discussion and Limitations

6.1 What limits recovery?

Retreating returns the arm toward an earlier configuration after a predicted failure. A retry may differ because the arm returns only approximately, the scene has changed, or the policy samples different actions. Our experiments do not isolate these sources of variation. The detector supplies

Table 6: Threshold and retreat rule on SmolVLA, LIBERO-Long, `post` head, five seeds, 50 episodes per task, paired by seed. Control success is 61.00%, a failure rate of 39.00%. The first block sweeps the trigger threshold under SmolVLA's reported adaptive rule, which retreats between 50 and 150 steps, stopping once the predicted failure probability falls below 0.2; the second holds the threshold at the reported t=0.97 and substitutes a fixed retreat. Fire rate is the proportion of episodes in which the controller triggers at least once. Unlike for π0.5 (Table 5), we did not sweep the threshold under the fixed rule.

CONFIGURATION	SR	UPLIFT	95% CI	RECOVERY	FIRE
Threshold, adaptive retreat (50–150 steps, cd=50)					
t=0.90	65.76	+4.76	[+1.19, +8.33]	12.2%	49.8%
t=0.95	66.44	+5.44	[+2.78, +8.10]	13.9%	47.5%
t=0.97 (headline)	68.40	+7.40	[+4.52, +10.28]	19.0%	42.6%
t=0.99	65.60	+4.60	[+2.43, +6.77]	11.8%	42.5%
t=0.995	65.00	+4.00	[+0.69, +7.31]	10.3%	39.2%
Retreat rule and depth, fixed (t=0.97)					
s=50, cd=50	66.16	+5.16	[+4.29, +6.03]	13.2%	43.3%
s=100, cd=100	67.04	+6.04	[+3.90, +8.18]	15.5%	44.0%

the trigger, while backtracking determines where the policy resumes; success depends on whether the unchanged policy can complete the task from that resulting state.

Backtrack deliberately uses a simple recovery mechanism: it requires no recovery demonstrations or retraining of the VLA, and does not learn a recovery policy or generate a new strategy. That this intervention reduces failure rates by 19–48% on LIBERO-Long suggests how much can be gained from return-and-retry alone, while the remaining failures leave headroom for recovery mechanisms that can actively change the policy's strategy after failure.

6.2 Limitations

Simulation only. Every result here is in simulation. We do not validate on a physical robot, and reverse execution on hardware requires validation of contact forces, tracking error, and safety near people or fragile objects. MetaWorld adds a second benchmark, but it is also simulated and was run only with SmolVLA.

Reversibility. Backtrack restores neither object poses nor the full environment state, and executing the return trajectory can itself change the scene. It cannot recover failures that have made the task physically impossible, and it may not help when recovery requires a new manipulation strategy.

7 Conclusion

We introduced Backtrack, a lightweight failure detection and recovery mechanism for frozen VLAs. Backtrack trains only a small failure prediction head on rollouts of the unchanged policy and requires no recovery demonstrations or VLA retraining. At inference time, it uses the policy's own representations to detect likely failure and retraces recent arm motion before returning control to the policy. The head adds less than 1% to the policy's parameter count, even on the 450M-parameter SmolVLA, and requires no separate model at inference.

On LIBERO-Long, the suite on which all three VLAs fail most often, Backtrack reduces failure rates by 19–48%; on the three shorter suites, where control success is already 85–99%, its effect is smaller. The method also improves performance on a second benchmark: on MetaWorld, every controller configuration tested with SmolVLA, the smallest policy we evaluate, improves on the unassisted policy, as does every SmolVLA configuration tested on LIBERO-Long. These results suggest that a meaningful fraction of VLA failures can be recovered with a simple return-and-retry mechanism, without collecting recovery demonstrations or modifying the underlying policy, and that the benefit is largest where the base policy fails most.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods, providing feedback on research methodology and experiments, and interpreting experimental results. We have not used generative AI tools for proposing or refining hypotheses, and the following required disclosure tasks are not applicable to this work: generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proofs, assisting in writing proofs, translation, cleaning or reformatting datasets, and supporting qualitative or thematic data analysis. Additionally, we used generative AI tools for software development, literature review, and drafting, editing, and restructuring the paper. We have reviewed all AI-assisted work. AI-generated code was reviewed and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The paper specifies the failure-prediction head and recovery controller, and Section 4 describes the policies, evaluation protocol, and step budgets. Appendix A.1 reports the hyperparameter sweeps and configuration-selection procedure, with additional ablations and implementation details provided throughout the appendix. Code will be released publicly.

REFERENCES

Christopher Agia, Rohan Sinha, Jingyun Yang, Ziang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking Failure Modes of Generative Policies: Runtime Monitoring of Consistency and Progress. In *Proceedings of The 8th Conference on Robot Learning*, pp. 689–723. PMLR, January 2025. URL https://proceedings.mlr.press/v270/agia25a.html.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert 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 Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, September 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025b. doi: 10.15607/RSS.2025.XXI.010.

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

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Khalil Meftah, Maxime Ellerbach, Jess Moss, and Thomas Wolf. LeRobot: State-of-the-art Machine Learning for Real-World Robotics in Pytorch, 2024. URL https://github.com/huggingface/lerobot.

Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. RACER: Rich Language-Guided Failure Recovery Policies for Imitation Learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15657–15664, 2025. doi: 10.1109/ICRA55743.2025.11127799.

Maximilian Du, Alexander Khazatsky, Tobias Gerstenberg, and Chelsea Finn. To Err is Robotic: Rapid Value-Based Trial-and-Error during Deployment, June 2024. URL http://arxiv.org/abs/2406.15917. arXiv:2406.15917 [cs.RO] version: 1.

Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-Language Models as Success Detectors. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*, pp. 120–136. PMLR, November 2023. URL https://proceedings.mlr.press/v232/du23b.html.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A Vision-Language-Model for Detecting and Reasoning Over Failures in Robotic Manipulation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 45493–45517, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/70a06501001e1820fd1eb9ee821302d2-Paper-Conference.pdf.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A Progressive Robustness Benchmark for Visual-Language-Action Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, June 2026.

Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. SAFE: Multitask Failure Detection for Vision-Language-Action Models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 40041–40076. Curran Associates, Inc., 2025. doi: 10.52202/085713-1337. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/392d0d05e2f514063e6ce6f8b370834c-Paper-Conference.pdf.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model, September 2024. URL http://arxiv.org/abs/2406.09246. arXiv:2406.09246 [cs].

Zijun Lin, Jiafei Duan, Haoquan Fang, Dieter Fox, Ranjay Krishna, Cheston Tan, and Bihan Wen. FailSafe: Reasoning and Recovery from Failures in Vision-Language-Action Models, October 2025. URL http://arxiv.org/abs/2510.01642. arXiv:2510.01642 [cs.RO].

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023a. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3468–3484. PMLR, November 2023b. URL https://proceedings.mlr.press/v229/liu23g.html.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. SmolVLM: Redefining small and efficient multimodal models, April 2025. URL http://arxiv.org/abs/2504.05299. arXiv:2504.05299 [cs.AI].

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models, January 2025. URL http://arxiv.org/abs/2501.09747. arXiv:2501.09747 [cs.RO].

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics, June 2025. URL http://arxiv.org/abs/2506.01844. arXiv:2506.01844 [cs].

Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, Siteng Huang, Yifan Tang, Wenhui Wang, Ru Zhang, Jianyi Liu, and Donglin Wang. VLA-Adapter: An Effective Paradigm for Tiny-Scale Vision-Language-Action Model, September 2025. URL http://arxiv.org/abs/2509.09372. arXiv:2509.09372 [cs.RO].

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, November 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

A Appendix

A.1 Hyperparameter sweeps and configuration selection

Each policy is evaluated with a single controller configuration, selected on LIBERO-Long and transferred unchanged to the other three suites. The paragraphs below describe the sweep used to select each reported configuration; for π0-FAST and π0.5, selection was performed using a different feature tap from the one used for final evaluation.

SmolVLA. Three configurations, differing only in the trigger threshold $t \in \{0.90, 0.97, 0.99\}$, with the retreat rule held fixed (adaptive, restart threshold $a = 0.2$, minimum 50 steps, maximum 150, cooldown 50). Three seeds at 10 episodes per task. The configuration with $t = 0.97$ had the highest mean success rate in this sweep. At that sample size the per-arm intervals span roughly ± 20 pp, so this comparison cannot separate the three candidates and the choice should be read as weakly determined rather than tuned. We therefore re-evaluate the threshold at the full protocol in Section 5.5: five seeds and 50 episodes per task, over five settings spanning 0.90–0.995. There $t = 0.97$ remains the highest of the five, the four alternatives cluster 2–3.4 pp below it, and only $t = 0.995$ is resolvably worse. The retreat parameters ($a = 0.2$, minimum 50, maximum 150, cooldown 50) were held fixed throughout and never tuned.

π0-FAST. Six configurations on LIBERO-Long, five seeds at 10 episodes per task: adaptive retreats at $t \in \{0.80, 0.85, 0.90\}$ with $a = 0.2$, minimum 50, maximum 150 and cooldown 50; an adaptive retreat at $t = 0.90$ with $a = 0.3$, minimum 25, maximum 100 and cooldown 50; and fixed retreats of 50 steps (cooldown 50) and 100 steps (cooldown 100), both at $t = 0.90$. The adaptive $t = 0.90$, $a = 0.3$ configuration had the highest mean success rate on selection seeds 1000–3000; seeds 4000–5000 were held out.

This selection was run on the `pre` head, and the configuration was carried over unchanged to the `post` head used for final evaluation. The two taps were compared separately (Section 5.5), but

Table 7: Threshold and retreat shape on π0-FAST, LIBERO-Long, `post` head, five seeds, 50 episodes per task, paired by seed. Control success is 85.32%, a failure rate of 14.68%. The first block varies the threshold under the reported retreat shape; the second varies it under the retreat shape used for the SmolVLA and π0.5 threshold sweeps (Tables 5 and 6). a is the restart threshold. Fire rate is the proportion of episodes in which the controller triggers at least once.

CONFIGURATION	SR	UPLIFT	95% CI	RECOVERY	FIRE
Threshold, reported shape (adaptive, a=0.3, 25–100 steps, cd=50)					
t=0.90 (headline)	92.40	+7.08	[+5.91, +8.25]	48.2%	33.7%
t=0.95	87.48	+2.16	[+0.45, +3.87]	14.7%	31.4%
t=0.99	90.84	+5.52	[+3.42, +7.62]	37.6%	20.8%
t=0.995	89.44	+4.12	[+2.38, +5.86]	28.1%	19.6%
Threshold, SmolVLA/π0.5 shape (adaptive, a=0.2, 50–150 steps, cd=50)					
t=0.90	86.08	+0.76	[−0.96, +2.48]	5.2%	38.0%
t=0.95	88.32	+3.00	[+1.02, +4.98]	20.4%	29.8%
t=0.99	89.72	+4.40	[+2.55, +6.25]	30.0%	21.7%
t=0.995	88.92	+3.60	[+0.52, +6.68]	24.5%	20.1%

that comparison does not establish that controller rankings are preserved across representations; the `post`-head checks below address this directly.

After selection, we ran two threshold sweeps on the `post` head at the full protocol, five seeds and 50 episodes per task, both paired by seed against the headline control (Table 7). The first holds the reported retreat shape fixed (adaptive, $a = 0.3$, minimum 25, maximum 100, cooldown 50) and varies the threshold over $t \in \{0.95, 0.99, 0.995\}$, with the reported $t = 0.90$ arm as the fourth point. All three alternatives improve over control, but each is resolvably worse than the reported configuration when paired by seed: -4.92 pp $[-6.48, -3.36]$ at $t = 0.95$, -1.56 pp $[-3.00, -0.12]$ at $t = 0.99$, and -2.96 pp $[-4.82, -1.10]$ at $t = 0.995$. The second sweep uses the retreat shape of the SmolVLA and π0.5 threshold sweeps (adaptive, $a = 0.2$, minimum 50, maximum 150, cooldown 50) over $t \in \{0.90, 0.95, 0.99, 0.995\}$. Its best point, $+4.40$ pp at $t = 0.99$, is below the reported configuration. At the shared $t = 0.90$, the two shapes differ by 6.32 pp. Threshold rankings also depend on retreat shape: $t = 0.90$ is the best threshold under the reported shape and the worst under the other. The reported configuration is the best of the eight `post`-head configurations evaluated, although only these two retreat shapes were tested on `post`. These sweeps were run after the configuration had been selected and on the same seeds as the headline arm, so we use them only as a check, not to select or re-rank configurations.

π0.5. Four configurations on LIBERO-Long, five seeds at 50 episodes per task: fixed retreats of 50 steps (cooldown 50) and 100 steps (cooldown 100) at $t = 0.99$, a fixed 50-step retreat at $t = 0.995$, and an adaptive retreat at $t = 0.99$ spanning 25 to 100 steps with a cooldown of 50 and a restart threshold of 0.3. The fixed 50-step, 50-cooldown configuration at $t = 0.99$ had the highest mean success rate on selection seeds 1000–3000; held-out seeds 4000–5000 give $+0.70$ pp against a pooled $+1.60$ pp. This grid varied the retreat rule and its depth and compared the threshold only against 0.995; the full threshold sweep is reported in Section 5.5. The sweep read the `post_expert` tap. All four configurations were re-run on `post`, where the same configuration again wins on seeds 1000–3000 ($+2.20$ pp against $+1.07$ to $+1.33$ pp).

Held-out reporting. All three configurations were selected using seeds 1000–3000 only, so Section 5 also reports their uplift on seeds 4000–5000, which played no part in selection. All three continue to improve over control on these seeds. SmolVLA's uplift is essentially unchanged, at $+7.20$ pp against a pooled $+7.40$; π0-FAST's is $+7.40$ pp against a pooled $+7.08$; and π0.5's remains positive at $+0.70$ pp against a pooled $+1.60$, a reduction consistent with the narrow margins separating its candidate configurations. These estimates evaluate configurations fixed using seeds 1000–3000; they do not quantify variation in the selection procedure across different splits.

Table 8: Success rate against the number of flow-matching steps, SmolVLA. MetaWorld uses seeds 1000–5000 with a 500-step episode budget and 500 episodes per seed, 10 on each of the 50 tasks, with per-seed standard deviations. LIBERO is a single seed, 100 episodes per suite over the four suites, and carries no error bars.

FLOW-MATCHING STEPS	METAWORLD	LIBERO
1	64.88 ± 1.40	75.25
2 (ours)	71.92 ± 1.29	82.50
3	72.04 ± 0.57	85.00
5	71.60 ± 0.93	85.00
7	70.48 ± 0.97	85.75
10 (default)	68.88 ± 0.73	83.75

Table 9: Control success rate against step budget, LIBERO-Long, five seeds, 500 episodes per seed. 520 steps is the wrapper default and 1040 is the budget used throughout the paper. Computed offline from the per-episode step counts of the control arms, so no additional rollouts were run; the 1040 column reproduces the control success rates in Table 3.

POLICY	520	624	728	832	936	1040	Δ
SmolVLA	57.28	58.64	59.40	60.20	60.80	61.00	$+3.72$
π0-FAST	83.00	83.76	84.12	84.24	85.12	85.32	$+2.32$
π0.5	91.88	93.36	93.88	94.44	94.64	94.72	$+2.84$

A.2 Flow-matching steps

SmolVLA is evaluated by Shukor et al. (2025) with ten flow-matching steps; we use two throughout. On MetaWorld, five seeds at 500 episodes each, two steps is *better* than ten (71.92 against 68.88, per-seed standard deviation around 1), and the response falls monotonically above three steps. On LIBERO, a single-seed check gives 82.50% success at two steps versus 83.75% at ten. That check has no error bars, at 100 episodes per suite on one seed, and the LIBERO response is not monotone, with seven steps highest at 85.75, so we treat it as indicative and rest the choice on the MetaWorld sweep.

A.3 Control success against step budget

Backtracking spends steps, so we evaluate at twice the wrapper step limit and charge retreat steps against the same budget for control and treatment alike. That raises an obvious question: how much of the uplift could be had by simply allowing more time? Table 9 answers it by re-scoring the control arms at intermediate budgets, which needs no additional rollouts, since an episode that succeeded at step n also succeeds under any budget of at least n.

Doubling the budget from the wrapper default increases success by 2.3–3.7 percentage points. Most failures at the default budget persist at the doubled budget. Backtrack's reported uplift is measured against this longer-budget control, so it is additional to these gains from time alone.

A.4 Per-suite composition of the head training data

Table 2 in Section 3 gives the size of each rollout pool. Table 10 breaks those pools down by suite. All three cover all four suites, but the mix differs substantially between policies: SmolVLA's pool is weighted towards LIBERO-Spatial and π0.5's towards LIBERO-Long, reflecting when each set of rollouts was collected rather than any deliberate design. This is the imbalance the balanced sampler is there to remove: without it, each head would be fitted mostly to the suite its own pool happens to over-represent, and the differences between policies would be confounded with the differences between collection runs.

Table 10: Rollout episodes per suite in each policy's failure-head training pool. All counts are episodes of the frozen policy, pooled across seeds. π0-FAST's LIBERO-Long count is 47 episodes short of a round 2,000 because one collection run ended early.

POLICY	SPATIAL	OBJECT	GOAL	LONG	TOTAL
SmolVLA	8,000	4,000	4,000	4,000	20,000
π0-FAST	2,000	2,000	2,000	1,953	7,953
π0.5	3,000	3,000	3,000	5,000	14,000

Table 11: Feature taps evaluated per policy, at each policy's headline controller configuration on LIBERO-Long, five seeds unless noted. Rows at 10 episodes per task are screening runs, comparable only with the other 10-episode rows of the same policy; rows marked (headline) are the full-protocol arms reported in Table 3. Uplift is in percentage points, paired by seed against each arm's own control; intervals are 95% confidence intervals over the per-seed differences. Controls differ between runs, so compare uplifts rather than success rates across rows.

POLICY	TAP	EPS/TASK	CONTROL	UPLIFT	95% CI
SmolVLA	`pre`	10	62.20	$+3.00$	$[-3.33, +9.33]$
	`post` (3 seeds)	10	62.67	$+8.67$	$[-0.72, +18.06]$
	`post_expert`	10	60.40	$+8.20$	$[+0.28, +16.12]$
	`post` (headline)	50	61.00	$+7.40$	$[+4.52, +10.28]$
π0-FAST	`pre`	10	85.20	$+5.80$	$[+0.08, +11.52]$
	`post`	10	85.20	$+7.20$	$[+4.37, +10.03]$
	`post_expert`	not applicable, no action expert			
	`post` (headline)	50	85.32	$+7.08$	$[+5.91, +8.25]$
π0.5	`pre`	not run			
	`post` (headline)	50	94.72	$+1.60$	$[+0.06, +3.14]$
	`post_expert`	50	94.72	$+1.88$	$[+0.16, +3.60]$

A.5 Feature taps evaluated

Table 11 lists every feature tap we evaluated in closed loop, for each policy, at that policy's headline controller configuration on LIBERO-Long. For SmolVLA and π0-FAST, the tap comparison comes from screening runs at 10 episodes per task, the only protocol at which every tap was run for those policies; the headline `post` arm at the full 50 episodes per task used for the main LIBERO recovery results is listed alongside. On π0.5 the taps were compared directly at 50 episodes per task. Each arm is paired by seed against the control it was run with. π0-FAST has no action expert, so `post_expert` does not exist for it, and `pre` was never run on π0.5.

A.6 Executable reversal and privileged restoration

Backtrack returns the arm by physically replaying its recorded motion in reverse. MetaWorld also allows a privileged alternative: restoring the arm directly to its earlier state without executing the return trajectory. At the same operating point, instantaneous restoration achieves $+5.72$ pp compared with $+2.80$ pp for executable reversal (Table 12); the seed-paired difference is $+2.92$ pp with a 95% interval of $[+1.44, +4.40]$.

This comparison does not isolate the source of the gap. Instantaneous restoration both returns the arm more precisely and avoids the time spent executing the retreat, during which the scene may continue to evolve. It therefore represents a privileged simulator intervention rather than a variant of Backtrack that could be executed directly on a physical system.

Table 12: Executable reversal against privileged instantaneous restoration, SmolVLA on Meta-World, seeds 6000–10000, a 500-step episode budget, and 500 episodes per seed. Both recovery mechanisms use the same operating point (t=0.8, 100-step retreat, 100-step cooldown) and are paired by seed against a shared control. All three arms come from the same evaluation sweep; uplifts are relative to its control.

RECOVERY	SR	UPLIFT	95% CI
control	70.56		
executable reversal	73.36	$+2.80$	$[+1.87, +3.73]$
instantaneous restoration	76.28	$+5.72$	$[+3.37, +8.07]$

Table 13: Retreat cap on π0.5, LIBERO-Long, `post` head, fixed 50-step retreat, 50-step cooldown, $t = 0.99$. Per-seed success rates at 50 episodes per task, and the seed-paired uplift against the shared control. Intervals are 95% confidence intervals over the three per-seed differences, using $t_{0.975,2} = 4.303$; they are wide because $n = 3$, and the comparison of interest is between caps rather than against zero. Control is the main evaluation's control restricted to seeds 1000–3000, giving 94.07% success and a $+2.20$ pp uplift for the headline configuration, compared with 94.72% and $+1.60$ pp over all five seeds (Table 3). The independently evaluated cap-3 arm matches the headline arm's per-seed success rates on these three seeds.

	SUCCESS RATE BY SEED						
CAP	1000	2000	3000	MEAN	SD	UPLIFT	95% CI
control	93.20	95.00	94.00	94.07	0.90		
1	96.80	95.60	95.40	95.93	0.76	$+1.87$	$[-1.99, +5.73]$
2	96.60	95.20	95.40	95.73	0.76	$+1.67$	$[-2.35, +5.68]$
3	96.20	97.00	95.60	96.27	0.70	$+2.20$	$[+0.41, +3.99]$
4	96.80	96.00	95.80	96.20	0.53	$+2.13$	$[-1.17, +5.44]$
5	95.40	96.00	95.60	95.67	0.31	$+1.60$	$[+0.11, +3.09]$

A.7 RETREAT-CAP ABLATION

Maximum retreats per episode limits how many times the controller may initiate a retreat during an episode. Once an episode has used its budget the trigger is disabled for the remainder of that episode and the policy runs unassisted to the step limit. The cap does not change the length of an individual retreat, only how many times the controller may fire.

We vary the cap over 1, 2, 3, 4 and 5 on π0.5 with the `post` head, on LIBERO-Long, holding the rest of the headline configuration fixed: a fixed 50-step retreat, a 50-step cooldown, and a trigger threshold of $t = 0.99$. Each arm is three seeds (1000, 2000, 3000) at 50 episodes per task over the suite's ten tasks, so 500 episodes per seed and 1,500 per cap, evaluated at twice the wrapper step limit (1040 steps). Retreat steps are charged against that budget, as everywhere else in the paper.

Table 13 gives the result. The five caps span 0.60 percentage points, from 95.67 at a cap of 5 to 96.27 at a cap of 3, against a per-seed standard deviation of 0.31 to 0.76 within each arm. Comparing caps directly and pairing by seed, a cap of 1 against a cap of 3 is -0.33 pp with an interval of $[-2.83, +2.17]$, and a cap of 3 against a cap of 5 is $+0.60$ pp with an interval of $[-0.71, +1.91]$. Neither comparison resolves a difference. With only three seeds, these intervals still allow differences of several percentage points; they do not establish that one retreat is sufficient or that the caps are equivalent.

A.8 METAWORLD CONTROLLER SWEEP

Table 14 reports all twelve controller configurations evaluated in Section 5.4. Each configuration is compared with the control from this sweep. The flow-step ablation (Table 8) and restoration

Table 14: Controller sweep on MetaWorld, SmolVLA, `pre` head. All arms use seeds 1000–5000, a 1000-step episode budget including retreat steps, and 500 episodes per seed (10 per task across all 50 tasks). Control success is 72.32%. SR is success rate (%); uplifts and intervals are in percentage points relative to this sweep's control. Intervals are 95% confidence intervals over the five per-seed differences. Here t is the trigger threshold, a the adaptive restart threshold, s the fixed retreat length or adaptive minimum–maximum, and cd the cooldown; lengths and cooldowns are in steps.

t	a	s	cd	**SR**	**UPLIFT**	**95% CI**
Control				72.32		
Fixed retreat						
0.8	–	100	50	74.76	$+2.44$	$[-0.65, +5.53]$
0.8	–	100	100	76.28	$+3.96$	$[+1.62, +6.30]$
0.9	–	100	50	75.44	$+3.12$	$[+1.14, +5.10]$
0.9	–	100	100	75.96	$+3.64$	$[+1.57, +5.71]$
0.9	–	150	100	77.52	$+5.20$	$[+2.28, +8.12]$
0.9	–	200	100	77.20	$+4.88$	$[+1.61, +8.15]$
Adaptive retreat						
0.8	0.2	50–150	50	75.32	$+3.00$	$[+0.37, +5.63]$
0.9	0.1	50–150	50	75.28	$+2.96$	$[+1.34, +4.58]$
0.9	0.2	50–100	50	75.92	$+3.60$	$[+0.57, +6.63]$
0.9	0.2	50–150	50	75.96	$+3.64$	$[+1.09, +6.19]$
0.9	0.2	50–200	50	76.44	$+4.12$	$[+1.38, +6.86]$
0.9	0.3	50–100	100	76.36	$+4.04$	$[+1.74, +6.34]$

comparison (Table 12) are separate evaluations with 500-step budgets; this sweep uses 1000 steps for both control and treatment.