THINKING MORE, MOVING LESS: STEALTHY AVAILABILITY ATTACKS ON REASONING VLA MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning vision-language-action (VLA) models are increasingly studied for physical AI, yet the robustness of their computational efficiency during execution remains largely unexplored. We investigate a novel behavior-preserving availability attack, in which a static visual patch increases both the number of policy calls and the inference cost per call while preserving task-directed behavior. We introduce SPADE, a two-stage framework that separates the discovery of executable slow behavior from its realization through pixels. First, a learnable embedding suffix elicits smaller task-directed movements and longer reasoning from a frozen VLA, and closed-loop validation retains only behaviors that increase both cost factors while satisfying task, motion, and action-validity constraints. Second, a visual patch distills these behaviors by matching reasoning-exit decisions and actions, with supervision continually refreshed on the patched policy's own prefixes and visited states. At deployment, only the fixed patch remains, leaving the model weights, instructions, decoding procedure, and controller unchanged. Across three reasoning VLAs and two robotic manipulation benchmarks, SPADE substantially increases cumulative inference time while largely preserving task success and trajectory geometry, and achieves greater cost amplification than direct patch optimization under the same optimization budget. These results reveal an availability vulnerability that conventional task-success evaluation can overlook.

1 INTRODUCTION

Reasoning vision-language-action (VLA) models are increasingly studied for embodied AI, which generate explicit intermediate reasoning, describing objects, plans, and movements, before predicting robot actions. This design can improve task generalisation and make policy decisions easier to inspect (Zawalski et al., 2024; Lee et al., 2025; Yin et al., 2025). However, it also places reasoning on the critical path of control: when a policy reasons before returning each action, every additional reasoning token delays the next action. As reasoning length varies with the input, the inference cost of such policies is inherently input-dependent and may therefore be susceptible to adversarial manipulation. This creates a failure mode that task success alone cannot capture: a robot may still complete the intended task along a plausible trajectory while consuming substantially more computation and execution time.

We ask whether a static visual patch can jointly amplify these two sources of inference cost. Smaller task-directed approach movements can increase the number of policy calls required before grasping, while longer reasoning can increase the latency of each call. Since cumulative inference time is determined by the product of call count and mean per-call latency, increasing both factors can compound the overall computational burden. We formulate this objective as *behaviour-preserving availability degradation*: increasing inference cost while preserving task completion, valid actions, and trajectory geometry within predefined bounds. By maintaining task-directed behaviour, the attack can remain difficult to detect through outcome monitoring alone, as the robot continues to make visible progress toward its intended goal.

Existing attacks on vision-language and VLA models pursue related but distinct objectives. TRAP redirects VLA reasoning and actions towards an adversary's goal, while POV disrupts perception and trajectories (Huang et al., 2026; Wang et al., 2026). Verbose Images and Hidden Tail increase vision-language generation cost (Gao et al., 2024; Zhang et al., 2025), and scene-text triggers can delay

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.

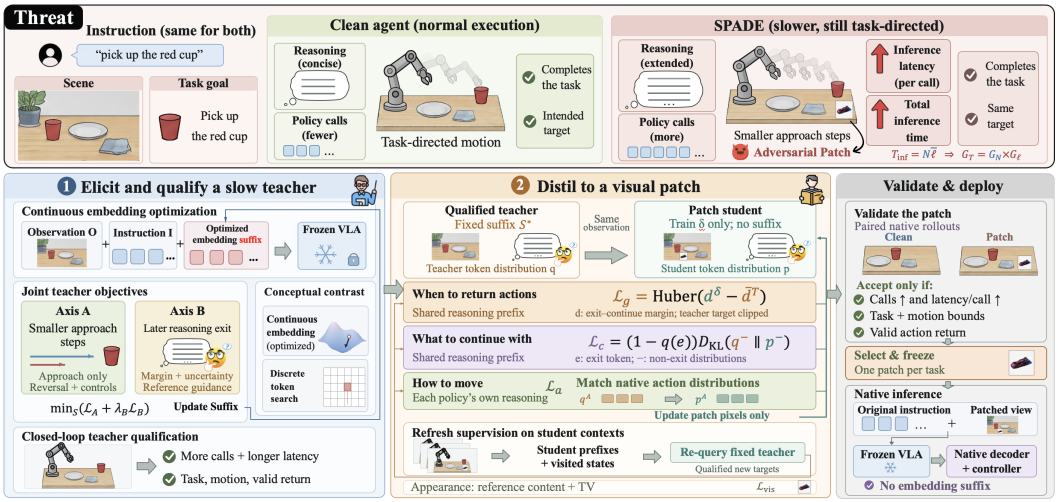


Figure 1: **Overview of the SPADE attack framework.** Offline suffix search elicits a qualified slow teacher; distillation yields a fixed patch using refreshed student contexts. Here N counts calls, $\bar{\ell}$ is mean call latency, and total inference time is $T_{\text{inf}} = N\bar{\ell}$; G_N, G_ℓ, G_T are paired attacked/clean ratios for calls, latency, and total inference time. S is the suffix, δ the patch, and q, p the teacher/student distributions; $\bar{d}^T$ denotes the clipped teacher exit margin. Objectives are defined in Section 3.

robotic decisions (Han et al., 2026b). Our setting combines call-count and per-call cost inflation with successful, task-directed execution through a fixed patch. Assessing the attack requires evaluating the full closed control loop, rather than measuring the cost of isolated forward passes.

The central challenge is to preserve coordination between reasoning and execution while increasing computational cost. Smaller movements can disrupt contact timing, whereas excessively delayed reasoning can prevent the policy from returning a valid action. Direct patch optimisation must therefore solve two coupled problems simultaneously: discovering an executable slow behaviour and inducing that behaviour through a small image region. We address this challenge with SPADE, a two-stage attack that decouples behaviour discovery from visual realisation (Figure 1). In the first stage, we optimise a learnable embedding suffix (Schwinn et al., 2024) that conditions a frozen VLA to produce smaller task-directed movements and longer reasoning. Because the suffix operates directly in embedding space, it provides a less constrained search space than pixel-level optimisation for discovering such behaviours. We then qualify candidate teachers through native closed-loop execution, retaining only those that increase both cost factors while satisfying task-completion, motion-preservation, and action-validity requirements.

In the second stage, we optimise a visual patch to reproduce the behaviour of a qualified teacher, including its reasoning-exit decisions, continuation behaviour, and actions. Prior work has transferred text-induced behaviours to images through output matching (Bailey et al., 2023) or embedding alignment (Luo et al., 2025). In robotic control, however, the patch can induce distributional drift in both generated text prefixes and visited states. To account for this, we continually refresh teacher supervision on contexts encountered by the patch-controlled policy. A benign-reference penalty reduces visual conspicuousness, and the resulting patches are validated under native closed-loop execution. At deployment, a single patch is fixed per task, while the embedding suffix is removed.

We evaluate SPADE on three reasoning VLAs under both standard and perturbed conditions. Across these settings, SPADE substantially increases inference cost while largely preserving task success and trajectory geometry. It also achieves stronger cost amplification than direct patch optimisation under the same total optimisation budget. Our main contributions are as follows:

- **A constrained availability objective.** We formulate behaviour-preserving availability degradation for reasoning VLAs, jointly increasing policy-call count and per-call inference cost while enforcing explicit task-completion, motion-preservation, and action-validity constraints.

- **A two-stage visual attack.** We propose SPADE, which discovers executable slow behaviours using suffix-conditioned teachers and then transfers them to a static visual patch. Student-context supervision is used to mitigate drift in both reasoning prefixes and visited states.
- **A closed-loop evaluation.** We evaluate SPADE on Standard LIBERO and LIBERO-Plus (Liu et al., 2023; Fei et al., 2026), with budget-matched comparisons, component and deployment analyses, and an independent-proxy extension using public target feedback.

2 Problem Formulation and Threat Model

We separate policy-call count from per-call latency to characterise behaviour-preserving slowdown. At call t, policy P_θ with frozen weights θ receives observation O_t (including sensory history) and instruction I, generates reasoning R_t of L_t tokens, and returns H control commands $A_t = (a_{t,1}, \ldots, a_{t,H})$. H is fixed across conditions; terminal chunks may end early. A rollout τ contains N calls, distinct from frames or control steps. With measured call latency ℓ_t and mean $\bar{\ell} = N^{-1} \sum_{t=1}^{N} \ell_t$, cumulative inference time is

$$T_{\text{inf}}(\tau) = \sum_{t=1}^{N} \ell_t = N\bar{\ell}. \tag{1}$$

SPADE targets larger N through smaller task-directed movements and larger $\bar{\ell}$ through longer reasoning. Joint success requires both gains over clean execution within task, motion, and action-validity bounds. Preservation concerns geometry and manipulation, not pauses or reasoning readability. The mechanism requires variable-length reasoning with a detectable exit.

The white-box setting permits offline model/gradient access and simulator poses and phase annotations (Huang et al., 2026). Deployment uses one static RGB patch $\delta \in [0,1]^{h_p \times w_p \times 3}$, of height h_p and width w_p pixels, in a designated region across a task's layouts. Instructions, weights, decoding, sensors, and controller remain unchanged. Digital attacks alter one external-view stream and its history; other inputs are retained. A physical patch is visible to every camera viewing its surface. The black-box extension uses an independent white-box proxy and public target observations, actions, timing, and outcomes on training/validation layouts; target internals and held-out feedback remain inaccessible to optimisation (Section 3.3).

3 Method

SPADE separates the search for slow, task-directed behaviour from its realisation through pixels (Figure 1). We first elicit and qualify a suffix-conditioned teacher (Section 3.1), then distil its behaviour into a patch using student-context supervision (Section 3.2). Appearance constraints and closed-loop selection complete the construction (Section 3.3); deployment uses only the fixed patch.

3.1 Eliciting a Slow, Task-Directed Teacher

The teacher stage seeks smaller movements and longer reasoning that still permit successful execution. With the VLA frozen, we optimise a suffix $S \in \mathbb{R}^{n \times d}$ of n embedding positions of width d, appended after instruction I and before reasoning (Schwinn et al., 2024). Dropping the call index t, the policy at observation O is $Q_S(\cdot \mid O, I) = P_\theta(\cdot \mid O, [I; S])$, where $[I; S]$ denotes embedding concatenation. One suffix is shared across the task's training layouts. Candidate generation alternates with joint movement (Axis A) and reasoning (Axis B) updates, within per-row ℓ_2 balls around the initial embeddings (Section B).

Axis A: reducing approach progress. Let A^c be the native action matrix under clean reasoning without suffix or patch, and $\widehat{A}^S$ the differentiable suffix-conditioned action estimate, each with H command rows. Keep A^c fixed while evaluating $\widehat{A}^S$ under recorded suffix-generated reasoning. For command h, denote their physical-unit translations by $\Delta x_h^c, \Delta \widehat{x}_h^S$ and define mean step lengths

$$s^S = \frac{1}{H} \sum_{h=1}^{H} \|\Delta \widehat{x}_h^S\|_2, \qquad s^c = \frac{1}{H} \sum_{h=1}^{H} \|\Delta x_h^c\|_2.$$

Summing individual lengths avoids cancellation. Set gate $\chi = 1$ only when the entire clean chunk precedes contact, contains no gripper transition, and its mean step length s^c exceeds a threshold; otherwise $\chi = 0$. Let $0 < r < 1$ be the target retained-length ratio.

Let g, x be the approach-target and end-effector positions from training snapshots, fixed for the chunk. Predeclared phases identify the target; clean-chunk replay determines χ (Section B.1). Using positive-part function relu and cosine similarity $\cos_\epsilon$ with norm floor $\epsilon > 0$, define reversal penalty $B^S = H^{-1} \sum_{h=1}^{H} \mathrm{relu}[-\cos_\epsilon(\Delta \widehat{x}_h^S, g - x)]$. Diagonal W_χ rescales controls by fixed clean-data statistics: W_1 selects non-translation controls when $\chi = 1$, and W_0 all controls otherwise. With nonnegative weights λ_b, λ_k and Frobenius norm $\|\cdot\|_F$, minimise

$$\mathcal{L}_A = \chi\big[(s^S/s^c - r)^2 + \lambda_b B^S\big] + \lambda_k H^{-1} \|(\widehat{A}^S - A^c) W_\chi\|_F^2. \tag{2}$$

Skip the bracket when $\chi = 0$. This reduces approach step length while penalising reversal and preserving other controls. Expected-bin estimates and codec relaxations are training surrogates, not executed actions. Adapter qualification and all rollouts use native decoding (Section B.3).

Axis B: extending reasoning while preserving return. We adapt Verbose Images' exit-delay and uncertainty objectives (Gao et al., 2024), using a greedy continuation margin and reference guidance to limit content drift. At O, generate $M_B = 4$ candidates from the same suffix-conditioned policy Q_S (one greedy, three stochastic), each followed by native actions. Let $\mathcal{J}_B$ index nonempty candidates with valid returns and acceptable Axis-A direction/preservation scores. Candidate j has reasoning tokens $R_j = (r_{j,1}, \ldots, r_{j,L_j})$, length $L_j > 0$, and exit token e. For the no-suffix clean reasoning length $K > 0$, assign normalised weights $\omega_j \propto \exp\{2\,\mathrm{clip}(\log(L_j/K), 0, \log 3)\}$ over $\mathcal{J}_B$, where clip truncates to the stated interval. Longer feasible candidates receive more weight, with the gain capped to limit domination by one trace.

At recorded prefix $h_{j,i} = (r_{j,1}, \ldots, r_{j,i-1})$ before position i, let $z_{j,i}^S(v)$ be the suffix logit of token v, and $\mathcal{C}_{j,i}$ the legal continuations excluding e; $h_{j,1}$ is empty. Define exit and leading-continuation logits $z_{j,i}^{S,e} = z_{j,i}^S(e)$ and $z_{j,i}^{S,o} = \max_{v \in \mathcal{C}_{j,i}} z_{j,i}^S(v)$. The recorded tokens are supplied as context when scoring these distributions; they are not regenerated during each gradient update.

Let $\mathcal{A}_j \subseteq \{1, \ldots, L_j + 1\}$ contain positions allowing both choices, including the observed exit when eligible. Sample body positions $\mathcal{I}_j \subseteq \{1, \ldots, L_j\}$ with $|\mathcal{C}_{j,i}| \geq 2$; set $m_j = |\mathcal{I}_j|$. Let $q_{j,i}^{S,\mathcal{C}}$ be the suffix softmax restricted to $\mathcal{C}_{j,i}$ and $U_{j,i}$ uniform on that set. With margin $m_{\mathrm{len}} = 1$ and Kullback–Leibler (KL) divergence D_{KL}, define exit-delay and uncertainty losses

$$\begin{aligned} \mathcal{L}_{\mathrm{len},j} &= \frac{1}{|\mathcal{A}_j|} \sum_{i \in \mathcal{A}_j} \mathrm{relu}\Big(z_{j,i}^{S,e} - z_{j,i}^{S,o} + m_{\mathrm{len}}\Big), \\ \mathcal{L}_{u,j} &= \frac{1}{m_j} \sum_{i \in \mathcal{I}_j} \frac{D_{\mathrm{KL}}(q_{j,i}^{S,\mathcal{C}} \| U_{j,i})}{\log |\mathcal{C}_{j,i}|}. \end{aligned} \tag{3}$$

The hinge favours continuation over exit by margin m_{len}; uncertainty broadens continuation choices. Define $\mathcal{L}_{\mathrm{ref},j}$ as the mean KL over $\mathcal{I}_j$ from the suffix distribution to the no-suffix model at the same prefixes, both restricted to $\mathcal{C}_{j,i}$. Entropy broadening and reference anchoring trade off on the same positions without directly matching clean stopping. With weights $\lambda_u = \lambda_{\mathrm{ref}} = 0.1$, the reasoning loss is

$$\mathcal{L}_B = \sum_{j \in \mathcal{J}_B} \omega_j [\mathcal{L}_{\mathrm{len},j} + \lambda_u \mathcal{L}_{u,j} + \lambda_{\mathrm{ref}} \mathcal{L}_{\mathrm{ref},j}].$$

These losses encourage later exit; native-rollout qualification checks valid return. Coherence is not guaranteed. Empty-case and multi-token-exit rules follow Sections B.1 and B.3.

Joint updates and qualification. Minimise the state-averaged $\mathcal{L}_A + \lambda_B \mathcal{L}_B$ with relative weight $\lambda_B > 0$, averaging Axis A over the same R_j and ω_j. Thus both axes optimise the same suffix on the same candidate contexts. Candidate tokens, positions, weights, and clean/reference outputs stay fixed within each update block; suffix logits and action estimates are recomputed after each update of S. Candidates are regenerated at block boundaries. Sampling and simulation are not differentiated. Native greedy rollouts qualify joint call-count/latency gains under task, motion, and validity bounds; cap exhaustion is failure. Refresh schedules and thresholds appear in Sections B and C.

3.2 Distilling the Teacher through a Visual Patch

We next transfer the qualified teacher's behaviour to pixels, so that deployment requires no suffix. Fix $S^\star$ and define teacher $Q^\star = Q_{S^\star}$ and student $P_\delta(\cdot \mid O, I) = P_\theta(\cdot \mid \mathcal{C}_\xi(O, \delta), I)$. The compositor $\mathcal{C}_\xi$ renders patch δ with geometric and photometric transforms $\xi \sim \mathcal{D}_\xi$ from a fixed distribution. Both policies share frozen weights; only pixels are updated. At the same recorded prefix h_i before position i, the teacher and student token distributions are $q_i(v) = Q^\star(v \mid O, I, h_i)$ and $p_i(v) = P_\delta(v \mid O, I, h_i)$ over native allowed vocabulary $\mathcal{V}_i$. Observations, prefix tokens, and teacher outputs remain fixed during gradient updates.

Boundary-aware reasoning imitation. To transfer Axis B, match when the teacher continues or returns actions. For teacher and student logits $z_i^T(v), z_i^\delta(v)$ under greedy decoding, superscript T denotes the teacher. At positions allowing exit e and alternatives, define exit margins

$$d_i^T = z_i^T(e) - \max_{v \in \mathcal{V}_i \setminus \{e\}} z_i^T(v), \qquad d_i^\delta = z_i^\delta(e) - \max_{v \in \mathcal{V}_i \setminus \{e\}} z_i^\delta(v). \tag{4}$$

Positive margins favour exit. With clipping bound $M_g = 3$ and unit-transition Huber loss, fit the boundary using $\mathcal{L}_{g,i} = \mathrm{Huber}(d_i^\delta - \mathrm{clip}(d_i^T, -M_g, M_g))$. Let q_i^-, p_i^- denote distributions conditioned on $v \neq e$, including any other native stop/control tokens excluded from Axis B's legal continuations. Match this non-exit distribution:

$$\mathcal{L}_{c,i} = (1 - q_i(e)) D_{\mathrm{KL}}(q_i^- \| p_i^-), \qquad q_i^-(v) = \frac{q_i(v)}{1 - q_i(e)}, \quad v \in \mathcal{V}_i \setminus \{e\}. \tag{5}$$

Here $p_i^-(v) = p_i(v)/(1 - p_i(e))$; both use restricted softmax for stability. This matches non-exit content and competing stops without changing native decoding; it cannot repair a degraded teacher continuation. Multi-token exits use sequence supervision (Section B.3).

Native action imitation at the same state. To transfer Axis A, teacher and student generate their own reasoning R^T, R^δ at the same observation O. For a categorical head with J native action slots, let q_j^A, p_j^A be their action distributions at slot j under these respective contexts. Match them with

$$\mathcal{L}_a^{\mathrm{own}}(O) = \frac{1}{J} \sum_{j=1}^{J} D_{\mathrm{KL}}(q_j^A \| p_j^A). \tag{6}$$

Native head conventions follow Section B.3. Define auxiliary loss $\mathcal{L}_a^{\mathrm{aux}}$ by replacing the student's R^δ with R^T. This gives the student a usable action target before its own reasoning stabilises. The training loss $\mathcal{L}_a$ uses this branch during warm-up, then averages it with $\mathcal{L}_a^{\mathrm{own}}$ for valid student returns; otherwise retain the auxiliary branch and record failure (Section B.2).

Learning on the student's contexts. Distillation changes the student's future text and visited states, so teacher-only contexts can become unrepresentative. Mix teacher and student prefixes (Agarwal et al., 2024), with sampling distribution $\mathcal{H}(O)$ over prefix–position pairs (h_i, i) balancing exit neighbourhoods and ordinary positions. Initialise replay buffer $\mathcal{B}$ with teacher demonstrations, then add unpatched observations of student-visited states (Ross et al., 2011). Admit targets only when paired clean/teacher continuations qualify within the remaining budget. Refresh schedules and replay proportions are in Section B; failures remain in the accounting. Teacher feedback can be unreliable on student prefixes (Xu et al., 2025), and local action matching does not imply more closed-loop calls. Final patch rollouts test the transferred behaviour.

3.3 Optimisation, Selection, and Deployment

To obtain a deployable patch, we combine behavioural imitation with an appearance constraint and select checkpoints through closed-loop validation. For benign reference image I_j^{ref} indexed by j, let $\mathcal{L}_{\mathrm{vis}}(\delta; j)$ denote the appearance loss. Positive weights $\lambda_a, \lambda_g, \lambda_c, \lambda_v$ balance action, boundary, content, and appearance across replay states and rendering changes:

$$\min_{\delta \in [0,1]^{h_p \times w_p \times 3}} \mathbb{E}_{O \sim \mathcal{B},\, \xi \sim \mathcal{D}_\xi} \left[\lambda_a \mathcal{L}_a + \mathbb{E}_{(h_i, i) \sim \mathcal{H}(O)} (\lambda_g \mathcal{L}_{g,i} + \lambda_c \mathcal{L}_{c,i}) \right] + \lambda_v \mathcal{L}_{\mathrm{vis}}(\delta; j). \tag{7}$$

Appearance and rendering. Let ϕ be a frozen perceptual encoder of output dimension d_ϕ and TV the total variation (mean adjacent-pixel absolute difference), weighted by $\lambda_{\mathrm{TV}} > 0$. Following Huang et al. (2026), define feature distance D_j and appearance loss:

$$\begin{aligned} D_j(\delta) &= \|\phi(\delta) - \phi(I_j^{\mathrm{ref}})\|_2^2 / d_\phi, \\ \mathcal{L}_{\mathrm{vis}}(\delta; j) &= D_j(\delta) + \lambda_{\mathrm{TV}} \mathrm{TV}(\delta). \end{aligned} \tag{8}$$

The reference and within-rollout digital geometry stay fixed. Physical rendering projects one stationary surface into each camera that sees it.

Selection and deployment. Each qualified teacher receives an equal-duration pilot patch run. Continue the best validation patch with its teacher fixed, ranking checkpoints on a fixed clean-success cohort with penalties for task, action, and motion failures (Section B). Search, pilots, reference selection, feedback, and validation share the same total budget as direct optimisation. Accept only joint call-count/latency gains satisfying the predeclared bounds; deploy the fixed patch without suffix or policy changes.

Proxy-assisted black-box extension. An independent frozen white-box proxy supplies teacher search, distillation, and all gradients under the same objectives. Public target rollouts on training layouts rank proposed patches by total inference amplification with failure/appearance penalties; improving candidates seed further proxy updates. Disjoint validation layouts select the patch before held-out testing. Dual-axis qualification still requires both cost gains and all preservation bounds. Target weights, logits, gradients, and hidden states remain inaccessible; the suffix never reaches the target. Proxy compute and target queries, including validation, are separately budgeted (Section D.1).

4 Experiments

We evaluate whether a fixed patch can amplify inference cost while preserving task-directed execution. After introducing the setup (Section 4.1), we establish the attack effect (Section 4.2), examine operating conditions and visual conspicuousness (Section 4.3) and cross-task transfer (Section 4.4), identify component contributions (Section 4.5), and test printed patches on a physical robot (Section 4.6).

4.1 Experimental Setup

We pair clean and attacked executions to separate added cost from task disruption. Targets are DeepThinkVLA, CoTinyVLA, and the MiniVLA-1B Full-ECoT reproduction (ECoT) (Yin et al., 2025; Lee et al., 2026; Chen et al., 2025), evaluated on Standard LIBERO's 40 Spatial/Object/Goal/Long tasks and all 10,030 LIBERO-Plus variants (Liu et al., 2023; Fei et al., 2026). Plus perturbs the same task family. Native interfaces and model-training exposure are detailed in Section B.3.

Comparators and budgets. POV with shared training-state access (POV-SA) and UPA-RFAS retain their disruption objectives (Wang et al., 2026; Lu et al., 2026); Direct patch optimises SPADE's objectives in pixels. Clean inputs and benign prints calibrate visual interference. White-box runs receive 12 GPU-hours per task/attack seed, including teacher search and validation. Black-box runs use a separate frozen ECoT-OpenVLA-7B proxy (Zawalski et al., 2024), with 12 proxy GPU-hours and 20,000 public target calls including calibration and selection (Sections D and D.1).

Paired evaluation. Selected patches occupy 14.1% of the external view and are frozen per task/seed before held-out testing, with shared placement and rendering. The same Standard LIBERO patch is evaluated on corresponding Plus variants without further optimisation. Disjoint splits, repetitions, and attack seeds are specified in Section C.

Metrics. Inference inflation G_T is the geometric-mean ratio of attacked to clean cumulative inference time on jointly completed pairs. Call-count and mean-latency ratios G_N, G_ℓ use the same pairs, giving $G_T = G_N G_\ell$. Wall-time increase ΔT_{wall} is the task-balanced added seconds from first motion to success on those pairs. Task success rate (TSR) measures completion over all trials.

Normalised dynamic time warping (nDTW; $[0, 1]$) measures ordered path similarity on the clean-success cohort, including attacked failures; higher scores indicate closer paths (Ilharco et al., 2019). Joint attack success (JAS) is the percentage of that cohort completing validly within motion bounds and passing both gain thresholds. Section C gives the full definitions and statistical procedures.

4.2 Slowdown with Preserved Execution

We first examine whether increased inference cost coexists with successful manipulation and preserved motion. Both runs in Figure 2 succeed, while calls rise from nine to sixteen and mean latency from 1.7 to 3.4 s: approximately 15 versus 54 s of inference (3.6×). Mean path deviation is 17 mm and the path-length ratio 0.89; the additional cost thus accompanies successful execution along a nearby, shorter path. Aggregate motion diagnostics appear in Table 10.

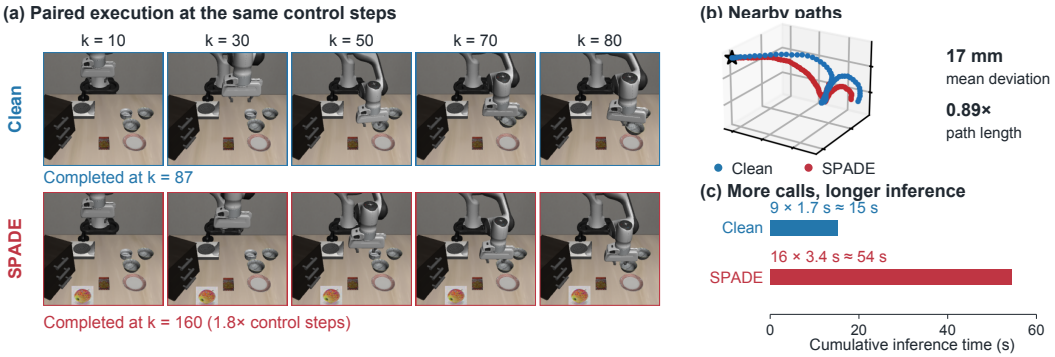


Figure 2: **Paired execution, geometry, and cost.** One clean/patch pair with the same task and initial state. (a) Control steps k, distinct from policy calls. (b) End-effector paths. (c) Calls × rounded mean latency; approximate inference totals, distinct from motion-to-success time.

Table 1: **Main results on Standard LIBERO.** Fixed task-specific patches on held-out layouts; indented row: black-box access. G_T: inference ratio; ΔT_{wall}: added motion-to-success seconds; TSR: %; nDTW: $[0, 1]$. †Adaptations retaining the source objectives (Section D).

	DeepThinkVLA				CoTinyVLA				ECoT			
Method	G_T	ΔT_{wall}	TSR	nDTW	G_T	ΔT_{wall}	TSR	nDTW	G_T	ΔT_{wall}	TSR	nDTW
Clean	1.00×	0.0 s	94%	1.00	1.00×	0.0 s	95%	1.00	1.00×	0.0 s	86%	1.00
Benign print	1.03×	+0.3 s	93%	0.81	1.03×	+0.3 s	94%	0.83	1.03×	+0.4 s	85%	0.79
POV-SA†	2.22×	+15.7 s	37%	0.14	2.13×	+14.2 s	36%	0.13	1.98×	+12.4 s	38%	0.18
UPA-RFAS†	1.81×	+9.7 s	31%	0.19	1.85×	+9.4 s	34%	0.21	1.64×	+7.8 s	33%	0.16
Direct patch	3.01×	+19.1 s	87%	0.44	2.53×	+14.9 s	88%	0.47	2.10×	+12.1 s	79%	0.40
SPADE	**3.98×**	**+27.1 s**	**89%**	**0.60**	**3.48×**	**+21.7 s**	**90%**	**0.63**	**3.01×**	**+19.6 s**	**80%**	**0.56**
↳ *Black-box*	3.10×	+19.3 s	90%	0.57	2.75×	+15.4 s	91%	0.59	2.40×	+13.8 s	81%	0.53

Across all three policies, SPADE raises inference cost to 3.01–3.98× while keeping TSR within six percentage points of Clean (Table 1). It exceeds Direct patch in cost amplification, TSR, and nDTW under matched budgets that include teacher search. This supports separating behaviour discovery from pixel optimisation. POV-SA and UPA-RFAS also raise cost, but their low TSR and nDTW reflect their disruptive objectives. SPADE preserves paths better than these attacks and Direct patch, although nDTW remains below benign prints. Completed-pair costs complement all-trial TSR; Table 13 provides intervals and paired coverage.

The proxy-assisted variant retains 2.40–3.10× cost at 81–91% TSR without target gradients, using only proxy gradients and public target feedback under the separate compute and query budgets (Section D.1). Separately, the frozen-patch Plus evaluation tests scene variation within the same benchmark family, without refitting (Table 6).

4.3 Operating Conditions and Visual Conspicuousness

We next examine how patch area, scene changes, and inference budgets affect slowdown and task preservation (Figure 3), alongside the visual effect of appearance regularisation. These controlled DeepThinkVLA–Standard LIBERO studies complement the full Plus evaluation.

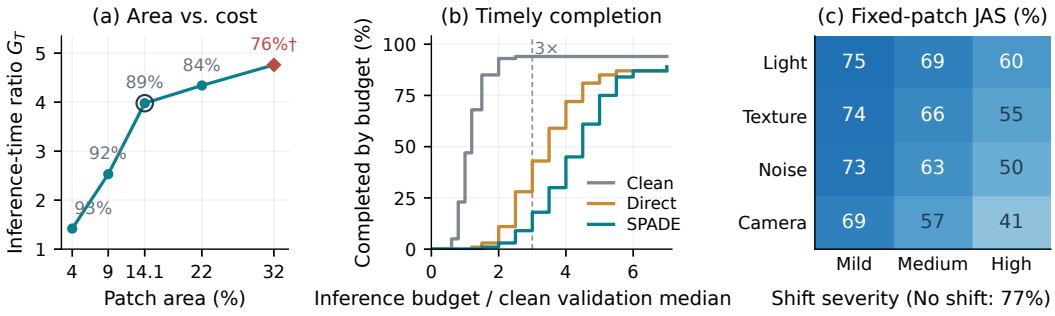


Figure 3: **Operating conditions on DeepThinkVLA–Standard LIBERO (white-box).** (a) Area versus cost; labels: TSR, ring: 14.1%, †: tolerance exceeded. (b) Completion by clean-normalised inference budget. (c) JAS on the shifted-clean-success cohort; no shift: 77%. See Section D.2 and Table 12 for protocols and TSR.

At equal optimisation budgets, 14.1% area gives $G_T = 3.98$ and 89% TSR. Increasing area to 32% raises cost to $4.76\times$ but lowers TSR to 76%, beyond the 10-point tolerance relative to 94% clean success. Larger cost alone therefore does not justify a larger patch. At $3\times$ the clean inference reference, completion rates are 94% (Clean), 43% (Direct), and 18% (SPADE). As more runs finish at larger budgets, the gap narrows: eventual TSR alone misses much of the delay. The frozen patch also retains 69–75% JAS under mild scene shifts versus 77% nominally, but larger camera yaw lowers it from 69% to 41%. This indicates bounded robustness rather than view invariance; paired TSR in Table 12 separates joint-attack loss from task failure.

We assess whether the patch becomes less conspicuous. Appearance regularisation lowers patch-crop Learned Perceptual Image Patch Similarity (LPIPS) distance to the reference from 0.48 to 0.14 (Zhang et al., 2018), and scene-level flagging (blinded suspicious-region judgements) from 78% to 21% (Tables 11 and 13). Compared with the unregularised patch, G_T decreases from 4.43 to 3.98, while TSR changes from 87% to 89%. Appearance refinement trades some cost amplification for lower conspicuousness while retaining substantial slowdown. These measures support improved reference similarity and reduced visual salience in the tested settings. The 12% benign-print flag rate remains lower, so camouflage is incomplete; Figure 5 shows a separate refined patch.

4.4 Transfer to Unseen Tasks

We test whether a patch remains effective beyond its fitted task. Eight predeclared tasks form 56 ordered source–destination pairs within each policy, with no destination feedback used for fitting or selection. Equally weighting these pairs, Table 2 shows 1.74–$2.46\times$ cost across access settings with 89–91% TSR. Thus, slowdown can survive a task change without destination-specific optimisation. This establishes transfer within the evaluated task set, rather than a universal patch; detailed splits and all three policies appear in Section D and Table 9.

Table 2: **Cross-task transfer.** Standard LIBERO; access during source-task fitting. G_T: cost ratio; TSR: %.

	DeepThinkVLA		CoTinyVLA	
Access	G_T	TSR	G_T	TSR
White-box	$2.46\times$	89%	$2.07\times$	90%
Black-box	$2.04\times$	90%	$1.74\times$	91%

4.5 Component Ablations

We isolate how the teacher objectives and distillation terms contribute to joint slowdown under matched area, budgets, and splits (Table 3 and Section D).

Axis A mainly increases calls ($1.57\times$), whereas Axis B mainly increases call latency ($2.70\times$). Joint optimisation retains most of each single-axis effect, yielding $3.98\times$ cost and 89% TSR. JAS rises

from 4% and 6% to 77%, consistent with its requirement that both factors increase in a valid completed rollout. The factor-wise results show how the two objectives complement each other.

The distillation controls further distinguish what must transfer to pixels. Removing boundary matching lowers latency inflation from 2.60× to 1.32×, while call inflation stays similar. Without action matching, calls fall to 1.17× and TSR to 64% despite longer inference. These patterns support distinct roles in transferring delayed return and executable motion. Longer reasoning can persist even as task execution degrades, which explains why per-call inflation must be assessed together with calls and TSR. Sequence imitation changes both supervision and prefix feedback; its 2.72× cost is a configuration comparison, not an isolated test of one loss (Section D).

Table 3: **Component ablations.** White-box DeepThinkVLA–Standard LIBERO patches. Ratios: ×; TSR/JAS: %. Full metrics: Table 8.

Variant	G_N	G_ℓ	TSR	JAS
Full SPADE	1.53	2.60	89	77
Axis A only	1.57	1.04	90	4
Axis B only	1.04	2.70	89	6
Sequence imitation	1.36	2.00	85	49
No boundary loss	1.50	1.32	90	26
No action loss	1.17	2.73	64	19

4.6 Physical Deployment

We test slowdown after printing and partial occlusion (Figure 4). DeepThinkVLA performs "pick up the orange" with the simulation-optimised patch, without physical reoptimisation. Each input/visibility condition uses ten held-out layouts with five paired repetitions each (50 trials). Clean, benign-print and patched runs share matched layouts, with input order randomised within blocks.

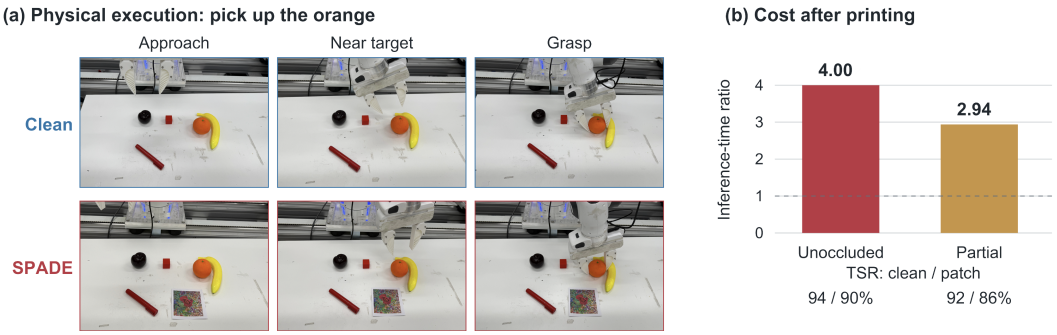


Figure 4: **Physical deployment.** (a) Clean and refined-patch execution of the same instruction; columns align phases, not times or initial poses. (b) Inference cost and clean/patch TSR by visibility (Tables 13 and 14); dashed line: unit cost. Apparatus and further phases: Figures 6 and 7.

Unoccluded deployment yields 1.60× calls and 2.50× mean latency, hence 4.00× inference cost, with 90% TSR (clean: 94%). Partial occlusion gives 2.94× cost and 86% TSR (clean: 92%). Separately measured motion-to-success time increases by 38 s and 25 s, respectively. Benign prints incur only 1.03× and 1.02× cost (Section D.2). The contrast with benign prints supports an effect beyond ordinary visual interference. Printing preserves joint slowdown in this setup, while partial occlusion attenuates it without eliminating the cost increase. The photographs document completed manipulation phases; logged timing and trajectories provide the temporal evidence. These results establish physical feasibility for the tested task and hardware, with broader deployment left open.

5 Conclusion

In this work, we introduced SPADE, a two-stage framework for behaviour-preserving availability attacks on reasoning VLAs. SPADE transfers slow, task-directed behaviours from suffix-conditioned teachers to a static visual patch, jointly increasing policy-call count and per-call latency. We provide sufficient experiments on three reasoning VLAs across Standard LIBERO and LIBERO-Plus show that SPADE achieves greater inference-cost amplification than direct patch optimisation under matched budgets, while largely preserving task success and trajectory geometry. These results demonstrate that successful task completion can mask substantial availability degradation, underscoring the importance of evaluating computational cost and execution behaviour throughout the control loop rather than relying on task outcomes alone.

AI USE STATEMENT

We used generative AI tools to assist in debugging parts of the experiment code and to improve the language and typesetting of the manuscript. They did not generate any data or experimental results. AI-assisted code was reviewed and tested 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.

ETHICS STATEMENT

This work exposes an availability vulnerability of reasoning VLA policies and is therefore dual-use; our aim is to show that task-success evaluation alone can miss this failure mode and to motivate cost-aware evaluation and defences. All simulation experiments use public benchmarks and publicly released checkpoints, and no deployed or third-party system was attacked. Physical experiments were conducted on an authorised laboratory robot under operator supervision with an independent safety stop. The blinded visual-conspicuousness study (Section D.2) involves only perceptual judgements of rendered scene images, and no personal information was collected.

REPRODUCIBILITY STATEMENT

The threat model and success criteria are specified in Section 2 and the method in Section 3. Algorithm 1 gives the end-to-end procedure and Table 4 the reference optimisation configuration, compute budget and hardware. Section B describes both optimisation stages and the native policy interfaces; Section C defines all metrics, data splits and the bootstrap procedure; Section D specifies the baseline, ablation, black-box, transfer, scene-shift, human-rating and physical protocols; and Section E provides supporting result tables, including 95% bootstrap intervals for the main comparisons (Table 13). All benchmarks are public, and all evaluated policies and the black-box proxy are built on publicly released checkpoints. Code and configuration files will be released.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pp. 284–293. PMLR, 2018.

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. Image hijacks: Adversarial images can control generative models at runtime. *arXiv preprint arXiv:2309.00236*, 2023.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on robot learning*, pp. 66–75. PMLR, 2020.

William Chen, Suneel Belkhale, Suvir Mirchandani, Oier Mees, Danny Driess, Karl Pertsch, and Sergey Levine. Training strategies for efficient embodied reasoning. *arXiv preprint arXiv:2505.08243*, 2025.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4793–4801. IEEE, 2019.

Jianshuo Dong, Ziyuan Zhang, Qingjie Zhang, Tianwei Zhang, Hao Wang, Hewu Li, Qi Li, Chao Zhang, Ke Xu, and Han Qiu. An engorgio prompt makes large language model babble on. In *International Conference on Learning Representations*, volume 2025, pp. 67280–67307, 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

Kuofeng Gao, Yang Bai, Jindong Gu, Shu-Tao Xia, Philip Torr, Zhifeng Li, and Wei Liu. Inducing high energy-latency of large vision-language models with verbose images. In *International Conference on Learning Representations*, volume 2024, pp. 17156–17182, 2024.

Jinyi Han, Ying Huang, Ying Liao, Haiquan Zhao, Zishang Jiang, Xinyi Wang, Xikun Lu, Guanghao Zhou, Sihang Jiang, Jiaqing Liang, et al. Your models have thought enough: Training large reasoning models to stop overthinking. In *International Conference on Learning Representations*, volume 2026, pp. 141341–141365, 2026a.

Qiang Han, Jie Wu, and Bo Chen. Overthink-triggered slowdown attacks on lvlm-based robotic systems. *arXiv preprint arXiv:2607.01518*, 2026b.

Zhengxian Huang, Wenjun Zhu, Haoxuan Qiu, Xiaoyu Ji, and Wenyuan Xu. Trap: Hijacking vla cot-reasoning via adversarial patches. *arXiv preprint arXiv:2603.23117*, 2026.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635*, 2020.

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.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

Minhyeok Lee, Chiyoung Kim, Chanhoe Gu, Seongrok Kim, Sanghyuk Roy Choi, Donghwan Hwang, Donghun Ryu, and Seokhyun Kim. Cotinyvla: Chain-of-thought distillation for a sub-billion-parameter vision-language-action model. *arXiv preprint arXiv:2607.25487*, 2026.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024a.

Zheng Li, Xiang Li, Xinyi Fu, Xin Zhang, Weiqiang Wang, Shuo Chen, and Jian Yang. Promptkd: Unsupervised prompt distillation for vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26607–26616. IEEE, 2024b.

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.

Hui Lu, Yi Yu, Yiming Yang, Chenyu Yi, Qixin Zhang, Bingquan Shen, Alex C Kot, and Xudong Jiang. When robots obey the patch: Universal transferable patch attacks on vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22867–22878, 2026.

Zhi Luo, Zenghui Yuan, Wenqi Wei, Daizong Liu, and Pan Zhou. An image is worth ten thousand words: Verbose-text induction attacks on vlms. *arXiv preprint arXiv:2511.16163*, 2025.

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Leo Schwinn, David Dobre, Sophie Xhonneux, Gauthier Gidel, and Stephan Günnemann. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *Advances in Neural Information Processing Systems*, 37:9086–9116, 2024.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot-Desenonges, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shahriari, Sarah Perrin, Abram Friesen, Geoffrey Cideron, et al. Bond: Aligning llms with best-of-n distillation. In *International Conference on Learning Representations*, volume 2025, pp. 59040–59060, 2025.

Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. Sponge examples: Energy-latency attacks on neural networks. In *2021 IEEE European symposium on security and privacy (EuroS&P)*, pp. 212–231. IEEE, 2021.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Chawin Sitawarin, Norman Mu, David Wagner, and Alexandre Araujo. Pal: Proxy-guided black-box attack on large language models. *arXiv preprint arXiv:2402.09674*, 2024.

Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on robot learning*, pp. 1723–1736. PMLR, 2023.

Xiaofei Wang, Mingliang Han, Tianyu Hao, Yi Yang, Yun-Bo Zhao, and Keke Tang. Partially observable adversarial patch attacks on vision-language-action models in robotics. *IEEE Robotics and Automation Letters*, 2026.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. Consistency of a recurrent language model with respect to incomplete decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5553–5568, 2020.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, volume 2025, pp. 64616–64646, 2025.

Cheng Yin, Yankai Lin, Wang Xu, Sikyuen Tam, Xiangrui Zeng, Zhiyuan Liu, and Zhouping Yin. Deepthinkvla: Enhancing reasoning capability of vision-language-action models. *arXiv preprint arXiv:2511.15669*, 2025.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Rui Zhang, Zihan Wang, Tianli Yang, Hongwei Li, Wenbo Jiang, Qingchuan Zhao, Yang Liu, and Guowen Xu. Hidden tail: Adversarial image causing stealthy resource consumption in vision-language models. *arXiv preprint arXiv:2508.18805*, 2025.

A Related Work

We relate the joint slowdown objective to visual attacks, controlled generation, and distillation, distinguishing shared mechanisms from differences in deployment.

Reasoning policies and their inference interfaces. ECoT places grounded task and motion descriptions before action prediction (Zawalski et al., 2024). DeepThinkVLA separates causal text generation from parallel action decoding (Yin et al., 2025). CoTinyVLA combines an episode-level Plan with chunk-level Think and continuous actions (Lee et al., 2026), while MolmoAct uses spatial traces to organise action reasoning (Lee et al., 2025). These interfaces are not interchangeable: a fixed number of trace points can expose action steering without exposing an expandable reasoning length. OpenVLA and its OFT recipe provide useful reasoning-free controls (Kim et al., 2024; 2025). Training Strategies for Efficient Embodied Reasoning separates learning from reasoning supervision from generating reasoning at test time (Chen et al., 2025). Its Full-ECoT recipe underlies our MiniVLA reproduction; its reasoning-free ECoT-Lite variants illustrate the boundary of the length mechanism. Eligibility is checked at the checkpoint level (Section B.3).

Visual attacks on robot policies. TRAP uses intermediate-reasoning and action supervision to produce a targeted change in robot behaviour through a visual patch (Huang et al., 2026). SPADE shares its offline white-box access and fixed-patch deployment, but asks whether the original task can remain recognisable while more computation is spent executing it. POV studies a different, partially observable attacker and combines perception disruption with trajectory-curvature optimisation (Wang et al., 2026). Comparisons must therefore identify any change in attacker access. Our joint motion and outcome reporting distinguishes extra computation from disrupted execution.

Energy and latency attacks. Sponge Examples established energy-latency attack objectives for neural networks (Shumailov et al., 2021). Verbose Images extends this concern to multimodal generation through termination, uncertainty, and diversity objectives (Gao et al., 2024). Hidden Tail preserves user-facing content while producing an expensive hidden token stream (Zhang et al., 2025). Its return condition differs from ours: a VLA needs usable actions after the additional reasoning. Engorgio combines termination suppression with model-based self-mentoring without specifying a fixed long response (Dong et al., 2025). VTIA first searches verbosity-inducing text embeddings and then aligns perturbed visual embeddings to them (Luo et al., 2025). Thus neither open-ended continuation nor two-stage input transfer is unique to SPADE. Overthink-triggered robotic slowdown already demonstrates visually delivered text triggers for expensive robotic decision making (Han et al., 2026b). Accordingly, our proposed contribution concerns a constrained patch, the joint closed-loop factors, and their behavioural preservation, rather than the first instance of visual slowdown.

Controlled generation and outcome-filtered self-training. GDC expresses controlled generation through constraints and minimal distributional change (Khalifa et al., 2020). STaR and ReST$^{\mathrm{EM}}$ learn from model-generated reasoning selected by outcome feedback (Zelikman et al., 2022; Singh et al., 2023); BOND amortises candidate selection into a policy (Sessa et al., 2025). JET conditions a preference for shorter reasoning on correct outcomes (Han et al., 2026a). These works motivate our candidate-refitting control. The primary teacher instead combines an exit margin, uncertainty, and conditional reference guidance, changing only input suffixes. Model-training results do not establish feasibility in this setting. s1 and L1 provide length-control precedents (Muennighoff et al., 2025; Aggarwal & Welleck, 2025), although deployment here permits neither forced decoding nor parameter updates. Greedy return requires verification because positive exit probability is insufficient (Welleck et al., 2020).

Distillation and state aggregation. PromptKD learns visual embedding prompts and a feature projector while freezing CLIP backbones (Li et al., 2024b); SPADE instead changes only a pixel patch. Learning by Cheating, perceptive locomotion, and RialTo distil policies with training-only state access into sensory policies (Chen et al., 2020; Miki et al., 2022; Torne et al., 2024). Our teacher's distinct capability is suffix control, not an additional state sensor. GKD trains on student-generated prefixes; SKD identifies unreliable teacher feedback on these contexts (Agarwal et al., 2024; Xu et al., 2025). DAgger motivates collecting learned-policy states (Ross et al., 2011), and

evidence that stronger teachers can be harder to distil motivates pilot selection (Cho & Hariharan, 2019). These precedents do not supply guarantees for input-channel transfer. Expectation over transformation addresses a declared family of rendering changes (Athalye et al., 2018).

Benchmarking context. LIBERO offers a controlled task and initial-state interface for manipulation (Liu et al., 2023). LIBERO-Plus probes benign shifts within that family (Fei et al., 2026), while BridgeData V2 provides recorded real-robot observations (Walke et al., 2023). SimplerEnv targets simulation evaluation matched to particular real robot policies (Li et al., 2024a). Their different embodiments, controllers, and data distributions motivate the scope distinctions in our experimental matrix.

B Implementation and Native Interfaces

This section specifies the choices needed to reproduce the two training stages. Table 4 contains the reference configuration, and Algorithm 1 summarises the white-box construction for one task and attack seed. Model weights, native decoding, and controller settings remain fixed throughout each comparison. All search, feedback and validation operations share the stated optimisation budget; held-out testing is separate. Let M be the median norm of ordinary policy token embeddings, ρ the suffix-row radius, and η_S the suffix step size. Candidate weighting uses reasoning-length gain cap g_B and exponent scale η (Section B.1).

Table 4: **Core optimisation configuration.** Search, rollouts, feedback and validation are included in the same total allowance as Direct patch.

Component	Setting
Compute	12 GPU-hours per task and seed; A100 80 GB; bfloat16
Allocation	20% teacher search, 10% patch pilots, 70% continuation
Suffix	$n = 32$, radius $\rho = 0.2M$, step $\eta_S = 0.005M/\sqrt{d}$
Teacher search	$r \in \{0.7, 0.85\}$, two initialisation seeds each
Axis A	$\lambda_b = \lambda_k = 1$; $s^c > 1$ mm; $\epsilon = 10^{-6}$ m
Axis B	$\lambda_B = 0.25$, $m_{\text{len}} = 1$, $\lambda_u = \lambda_{\text{ref}} = 0.1$
Candidates	$M_B = 4$; one greedy, three temperature-one samples
Candidate weights	$g_B = 3$, $\eta = 2$; at most 16 body positions
Native cap	DeepThinkVLA reference: 1,024 tokens; other native caps fixed per checkpoint
Patch	96×96 RGB; projected sign step $4/255$
Student weights	$(\lambda_a, \lambda_g, \lambda_c, \lambda_v) = (1, 1, 0.1, 0.1)$
Appearance	VGG-19 relu3_3 features; $\lambda_{\text{TV}} = 0.05$
Boundary	$M_g = 3$; unit-transition Huber loss
Minibatch / warm-up	Four states; at most 50 pilot updates on teacher contexts
Refresh / replay	Text every ten updates; states every 100; half original states
Student text	Teacher/student traces sampled equally; eight boundary and eight ordinary positions
Validation	Every 50 updates and at stage transitions

Proxy-assisted extension. For the black-box setting, perform teacher search and pixel-gradient updates on the independent white-box proxy. Training-layout target rollouts supply only the public feedback used to accept patch proposals; disjoint target validation layouts select the final patch. No target internals or target gradients enter either stage, and the suffix never reaches the target. Proposal scheduling and query accounting follow Section D.1.

B.1 Teacher Elicitation

Teacher optimisation alternates differentiable suffix updates with native rollout checks. The following settings define the initialisation, action reference, and candidate refresh.

Let S_0 be the embeddings of *Proceed carefully and complete the requested task.*, cycled to n positions. Each search starts from independent Gaussian noise of coordinate standard deviation $0.01M/\sqrt{d}$ around S_0, projected into the per-row radius-ρ balls. At update k, projected sign descent uses

$$S_{k+1} = \Pi_{\rho, S_0}\left(S_k - \eta_S \operatorname{sign} \nabla_{S_k}[\mathcal{L}_A + \lambda_B \mathcal{L}_B]\right),$$

Algorithm 1 SPADE: teacher elicitation, patch distillation and selection

Input: Frozen policy P_θ, instruction I, training layouts with successful clean traces $\mathcal{D}_0$, clean-success validation cohort $\mathcal{V}_0$, fixed reference I^{ref}, total budget C; qualified native adapters (Section B.3).
Output: Patch $\delta^\star$ and qualification flag ok, or a failed search.

Allocate C to teacher search C_S, pilots C_P and continuation C_R (Table 4); initialise qualified-teacher bank $\mathcal{T} \leftarrow \varnothing$.
Stage I: teacher elicitation and qualification
for each suffix initialisation and step-ratio setting within its share of C_S **do**
  Initialise suffix S and training-state replay from $\mathcal{D}_0$ (Section B.1).
  while the run's search allowance remains **do**
    Refresh states, clean references and weighted candidate reasoning as in Section B.1.
    Update only S by projected sign descent on $\mathcal{L}_A + \lambda_B \mathcal{L}_B$; detach contexts and share candidate weights across both axes.
    At scheduled native validations, retain qualified suffix teachers in $\mathcal{T}$ (Section B.2).
  end while
end for
Keep the best four teachers (or all if fewer); **if** $\mathcal{T} = \varnothing$ **then return** failed search.
Stage II: patch pilots, continuation and selection
for each $S^\star \in \mathcal{T}$ **do**
  Freeze $Q^\star = Q_{S^\star}$; initialise patch δ from I^{ref} and replay $\mathcal{B}$ from its demonstrations.
  Save FitPatch$(Q^\star, \delta, \mathcal{B}, C_P/|\mathcal{T}|)$.
end for
Restore the best pilot's $(Q^\star, \delta, \mathcal{B})$ by validation score and basic feasibility; add unused search time to C_R (Section B.2).
Resume FitPatch$(Q^\star, \delta, \mathcal{B}, C_R)$, retaining its pilot and continuation checkpoints.
ok $\leftarrow$ whether a retained checkpoint passes final qualification on $\mathcal{V}_0$; $\delta^\star \leftarrow$ the best qualifying checkpoint, or best-scoring failure otherwise (Sections B.2 and C).
return $(\delta^\star, \mathrm{ok})$; deploy only a qualified patch, without the suffix.
Patch fitting; all operations count towards allowance b
procedure FitPatch$(Q^\star, \delta, \mathcal{B}, b)$
  while allowance b remains **do**
    Sample replay states and rendering transforms; refresh teacher/student traces on schedule.
    Query the fixed teacher on clean observations and the patched student with *no suffix*.
    Match shared-prefix reasoning and native actions using the warm-up/own-context schedule in Section 3.2.
    Detach contexts and targets; update only δ by sign descent on Equation (7) and clip to $[0, 1]$.
    Refresh qualified student-state replay and validate/save checkpoints on schedule; include end-of-allocation validation (Section B.2).
  end while
  return patch/replay state and validation records, including failures.
end procedure

where Π_{ρ,S_0} projects each row around its initial token embedding. Dropout is disabled; KL and log probabilities are accumulated in float32.

Action reference and phase gate. At each snapshot, A^c is generated without suffix or patch and converted to Cartesian increments in metres. The clean chunk is replayed in an independent simulator copy. The gate $\chi = 1$ requires the entire chunk to precede task-relevant contact, contain no gripper transition, and satisfy $s^c > 1\,\mathrm{mm}$. Contacts are checked at physics substeps, excluding fixed base contacts. Task metadata and predeclared phase annotations specify the approach site g; unavailable targets disable the approach term. The observation, A^c, g, x, and χ remain fixed within the update block.

For vectors a, b in metres, $\cos_\epsilon(a, b) = a^\top b / [\max(\|a\|_2, \epsilon) \max(\|b\|_2, \epsilon)]$. ReLU is applied separately to each command's reversal score, so later forward motion cannot cancel an earlier reversal. Each coordinate scale is the larger of its clean-training standard deviation and 1% of its controller range. W_0 contains their inverses; W_1 sets translation entries to zero and retains rotation/gripper entries. These statistics are fixed before attack optimisation.

Candidates, support and empty cases. The four candidate draws share one suffix and observation. Sampling uses temperature one, with no extra top-k or top-p truncation beyond native masks.

The complete native exit sequence and required action markers must be generated; missing exits, invalid actions, and cap exhaustion are failures. The token cap includes transition tokens and autoregressive action codes; parallel action slots do not consume this budget.

For native candidate actions A_j, define

$$c_j^{\mathrm{keep}} = \chi\lambda_b B_j^{\mathrm{nat}} + \lambda_k H^{-1}\|(A_j - A^c)W_\chi\|_F^2,$$

where B_j^{nat} is the per-command reversal score on decoded translations. The acceptance threshold is the larger of 0.25 and the 95th percentile of repeated clean calibration scores. Nonempty candidates with valid return and acceptable c_j^{keep} form $\mathcal{J}_B$; duplicate draws are retained. Their weights are

$$u_j = \mathrm{clip}(\log(L_j/K), 0, \log g_B), \qquad \omega_j = \frac{\exp(\eta u_j)}{\sum_{k\in\mathcal{J}_B}\exp(\eta u_k)}.$$

Reduced step length is an optimisation target, not an additional candidate filter.

The margin is scored at eligible prefixes through the observed exit, not just where its hinge is active. The continuation set excludes the reasoning exit, premature whole-response stops, reserved action codes, and invalid control IDs. These are scoring supports, not changes to free decoding. Body positions are sampled uniformly without replacement, with $m_j = \min(16, |\mathcal{E}_j|)$, where $\mathcal{E}_j$ contains positions with at least two legal continuations and a recorded token in that set.

At the same prefix, let $q_{j,i}^{S,\mathcal{C}}$ and $p_{j,i}^{0,\mathcal{C}}$ be restricted softmax distributions with and without the suffix. The reference term is

$$\mathcal{L}_{\mathrm{ref},j} = \frac{1}{m_j}\sum_{i\in\mathcal{I}_j} D_{\mathrm{KL}}(q_{j,i}^{S,\mathcal{C}}\|p_{j,i}^{0,\mathcal{C}}).$$

Conditioning on continuation avoids restoring the clean stopping probability. Uncertainty and reference terms can compete when the reference is sharp; neither certifies coherent text. Both are zero when $m_j = 0$; an empty margin support also contributes zero before division.

For positive coefficients, the two regularisers have an interpretable compromise. At a fixed prefix, their unconstrained distribution-space minimiser satisfies $q(v) \propto [p_{j,i}^{0,\mathcal{C}}(v)]^{\beta_{j,i}}$, where $\beta_{j,i} = \lambda_{\mathrm{ref}}/(\lambda_{\mathrm{ref}} + \lambda_u/\log|\mathcal{C}_{j,i}|)$. It is a flattened reference distribution, rather than cancellation of two equal weights. This characterises the isolated regulariser; a shared suffix and the other objectives need not realise that optimum, nor does it guarantee coherent text.

The state loss is $\sum_{j\in\mathcal{J}_B}\omega_j\mathcal{L}_A(S;O,R_j) + \lambda_B\mathcal{L}_B$. If $K = 0$ or $\mathcal{J}_B$ is empty, Axis B is omitted and Axis A averages uniformly over valid action contexts; no valid contexts contribute zero. Candidate tokens, supports, weights and references are detached for at most ten updates, with logits recomputed every update. New observations trigger an immediate refresh. Every 100 updates, suffix rollouts on four training layouts refresh states while half the samples remain from successful clean trajectories. New clean references require a successful same-state clean continuation within the remaining episode budget.

B.2 Patch Imitation and Selection

Patch fitting transfers a qualified teacher while adapting supervision to the student's contexts. We specify the training schedule, rendering, and validation-based selection below.

Teacher checkpoints must pass the native task, motion, call-count and latency criteria of Section C, plus a geometric-mean reasoning-length gain of at least 1.10. At most four qualified teachers receive equal-duration patch pilots. Failure to obtain a qualified teacher is an unsuccessful search. Patch pixels start from the same benign reference with RGB noise of standard deviation 0.02, clipped to $[0, 1]$.

During warm-up, action imitation uses teacher reasoning. Afterwards, $\mathcal{L}_a = \frac{1}{2}(\mathcal{L}_a^{\mathrm{aux}} + \mathcal{L}_a^{\mathrm{own}})$ for valid student returns; invalid returns retain only the auxiliary term and remain failures. The replay buffer starts with qualified teacher demonstrations. Every 100 pixel updates, four student training rollouts supply up to eight uniformly spaced snapshots each. Paired clean/teacher continuations at those states must both succeed within the remaining budget and satisfy teacher motion bounds.

Accepted snapshots enter a first-in, first-out bank of up to 2,048 states, sampled equally with the original bank (up to 128 states per layout). The original bank is used alone while the new bank is empty.

Teacher and student text sources are equally likely. Each call supplies eight positions from the exit and its four predecessors and eight from other positions, with replacement if needed. A trace without exit uses its final five positions for the boundary group; empty groups transfer their weight, and empty traces retain teacher-context supervision. Losses average within groups, then calls. Text is refreshed every ten updates. Each student trace retains its rendering transform; changing that transform requires a new trace. The teacher receives the corresponding unpatched observation and shared prefix.

Rendering and appearance. The fixed 96×96 support occupies a 256×256 external view. Training uses rotation in $[-5^\circ, 5^\circ]$, scale in $[0.95, 1.05]$, translation within ± 3 pixels, and RGB gain in $[0.9, 1.1]$, followed by clipping. Geometry stays fixed within a digital rollout; physical rendering projects one stationary surface into all cameras that see it. Native preprocessing and required wrist views remain unchanged. The visual reference is fixed and resized to the patch dimensions. Features use ImageNet-pretrained VGG-19 relu3_3; TV is the mean absolute RGB difference over horizontally and vertically adjacent pixel pairs.

Selection and accounting. On a fixed clean-success validation cohort $\mathcal{V}_0$, trial v receives $q_v = \log(T_v^\delta / T_v^c)$ for completed, valid, motion-qualified execution and $q_v = -\log 2$ otherwise, where T_v^c, T_v^δ are cumulative inference times. Checkpoints maximise $|\mathcal{V}_0|^{-1} \sum_{v \in \mathcal{V}_0} q_v$ subject to cohort TSR, validity and aggregated motion bounds (Section C). Final patches also require both cost-factor gains; pilot selection uses the basic feasibility bounds while transfer develops. The selected teacher is frozen for the remaining patch optimisation. Failed searches and discarded candidates count towards the budget. Search, pilots, state collection, calibration and validation are charged; final held-out evaluation is separate. Loss-weight selection uses validation only, with factors $\{0.3, 1, 3\}$ around the reference values and the same accounting for Direct patch. Unused teacher-search time passes to continuation. Teacher checkpoints use the same validation score with the suffix in place of the patch.

B.3 Native Model Interfaces

The shared objectives are instantiated through each policy's native reasoning and action interfaces; these adapters do not change deployment decoding.

Each target keeps its native tokenizer, parser, controller units, cameras, execution horizon and configured generation cap. Standard LIBERO and LIBERO-Plus share the four-suite task family; eligibility is checked on the common manifest, not inferred from a model's published score. ECoT denotes the MiniVLA-1B reproduction of the Full-ECoT recipe (Zawalski et al., 2024; Chen et al., 2025). Its public LIBERO-90 artifact is architectural provenance, not the score used for our common 40-task and Plus evaluation. The exact weights, normalisers and any pre-attack adaptation identify each evaluated checkpoint.

Training exposure and visual input. DeepThinkVLA's released checkpoints are trained on Standard LIBERO; CoTinyVLA's release is trained on the four LIBERO-Plus suites (Yin et al., 2025; Lee et al., 2026). Thus frozen-patch transfer to Plus is not uniformly zero-shot policy generalisation. The patch modifies only the designated external camera stream, including its history frames before native preprocessing; wrist views and proprioception remain unchanged. The same scene support is projected consistently after camera changes.

Scalar and continuous action heads. DeepThinkVLA uses causal reasoning and parallel scalar action slots; its LIBERO configuration has $H = 10$, seven coordinates and 2,048 bins (Yin et al., 2025). At slot j with K_j native bins, logits z_j^S, centres $c_{j,k}$ and inverse normaliser U_j define the physical-coordinate estimate $\widehat{a}_j^S$:

$$\widehat{a}_j^S = U_j \left(\sum_{k=1}^{K_j} \mathrm{softmax}(z_j^S)_k c_{j,k} \right).$$

The checkpoint supplies the token-to-bin map. Controller-to-Cartesian conversion precedes Axis A; expected-bin values are surrogates, while rollouts use native argmax. Stage II retains native masks and zero parallel action-input embeddings.

CoTinyVLA uses eight external and eight wrist frames, eight-dimensional proprioception, and an $8\times$ 7 continuous action chunk (Lee et al., 2026). Axis A uses its continuous prediction; Stage II replaces categorical action KL with $H^{-1}\|(A^{\delta} - A^{T})W_0\|_F^2$, where A^{δ}, A^{T} are student/teacher actions. Each rollout maintains its own native Plan cache. Axis B targets the recurrent Think span ending in `<|cot_think_end|>`; cached Plan text is not regenerated per call or counted as newly generated reasoning. Measured call latency includes Plan generation when it actually occurs, preprocessing and action-head work. Native caps and cache behaviour stay identical between clean and attacked runs.

Codecs and native exits. ECoT's MiniVLA implementation decodes a ten-step chunk from vector-quantised action codes. Stage I passes probability-weighted codebook embeddings through the frozen continuous decoder and inverse normaliser; this is not generally the expected decoded action. Qualification requires reconstruction, gradient flow and agreement with native action changes. Autoregressive imitation uses detached native action prefixes and normalises by the teacher's complete code sequence. Failed interface qualification excludes the corresponding cell from dual-axis evaluation.

Every return verifier checks the complete native transition and action markers. For an exit $\mathbf{e} = (e_1, \ldots, e_q)$ of length q, the single-token teacher margin uses e_1 only if it uniquely commits the parser to exit. Otherwise that formulation requires a qualified boundary adapter. Student boundary supervision uses

$$\mathcal{L}_{\text{exit-seq}} = -\frac{1}{q}\sum_{k=1}^{q} \log P_{\delta}(e_k \mid O, I, h, e_{<k}),$$

where h precedes the delimiter and $e_{<k} = (e_1, \ldots, e_{k-1})$. Complete-sequence supervision is not a single-token greedy-margin guarantee. A negative local margin, small conditional KL, or successful teacher-forced action does not establish a valid free rollout. A missing exit or cap fallback fails SPADE's valid-return criterion even if the native wrapper emits an action. We retain that fallback and record the actual task outcome for TSR; qualification and JAS still count the return as invalid.

C Evaluation and Metrics

Paired evaluation separates completed-task cost from failure and trajectory preservation. Standard LIBERO uses 40 Spatial/Object/Goal/Long tasks, with 25/10/10 disjoint training/validation/test layouts per task. Three attack seeds and five paired repetitions measure optimisation and rollout variability; repetitions of one layout are not independent layouts. All declared tasks remain in TSR. Task manifests, checkpoint/code hashes, native caps, hardware and raw trial records identify the evaluated configuration. For the main comparisons in Table 13, bootstrap uncertainty uses 2,000 hierarchical resamples of tasks, layouts, repetitions and attack seeds, rather than tokens. Clean/attack pairing is retained within each resample.

LIBERO-Plus evaluates all 10,030 official variants in seven perturbation families. Each variant uses the frozen patch of its parent Standard LIBERO task and a matched unpatched run, retaining the published instruction and initial state. No Plus rollout enters patch fitting, scoring or selection. The official per-variant trial protocol is applied to each source attack seed. Metrics use the declared variant manifest, retaining parent-task identities and clean/patch pairing. Model-training exposure to Plus is recorded separately (Section B.3).

Cost, completion and failure. The exact identity in Equation (1) motivates two coupled cost factors. An affine approximation, $T_{\text{inf}}(\tau) \approx N(c_0 + c_d\bar{L})$, separates fixed prefill/action-head cost c_0 from per-token decoding cost c_d, where $\bar{L} = N^{-1}\sum_{t=1}^{N} L_t$ is mean reasoning length. All reported ratios use measured latency rather than this approximation. End-to-end time is measured separately because inference and movement can overlap.

For positive ratios x_v on the nonempty set $\mathcal{P}$ of completed clean/patch pairs, define $\mathrm{GM}_{v\in\mathcal{P}}\, x_v = \exp(|\mathcal{P}|^{-1}\sum_{v\in\mathcal{P}} \log x_v)$. For trial v, N_v^m, $\bar{\ell}_v^m$ and $T_v^m = T_{\text{inf}}(\tau_v^m)$ are call count, mean call latency

and cumulative inference time for rollout τ_v^m in condition $m \in \{c, \delta\}$. The cost factors use the same pairs:

$$G_N = \mathrm{GM}_{v\in\mathcal{P}} \frac{N_v^\delta}{N_v^c}, \qquad G_\ell = \mathrm{GM}_{v\in\mathcal{P}} \frac{\bar{\ell}_v^\delta}{\bar{\ell}_v^c}, \qquad G_T = \mathrm{GM}_{v\in\mathcal{P}} \frac{T_v^\delta}{T_v^c} = G_N G_\ell.$$

Reasoning-token gain is a separate diagnostic where raw tokens are available. Empty-pair ratios are undefined; completed-pair coverage accompanies all-trial TSR. Native image processing and action generation belong to call latency. Device and host timing use declared synchronisation and preprocessing boundaries; public API latency can include transport and scheduling. Exit failures, wrong manipulation, step-cap exhaustion and process errors are logged separately. A native fallback does not satisfy return validity; actual task completion determines TSR. Accrued cost on unsuccessful tasks is separate from completed-task ratios. Deadline curves count success before each threshold over all trials; inference-only and robot wall-clock deadlines differ. Deadline claims require measured timing and an explicit threshold; a step cap censors execution and does not establish a deadline miss.

Normalised inference budgets. For each task, fix the median cumulative inference time of successful clean validation runs as the reference before testing. At each multiple of this reference, compute the fraction of all trials that have completed successfully, then average task fractions equally. Failures and capped runs remain in the denominator. The final plateau is TSR under the shared rollout cap. An unavailable reference makes that task undefined and is reported with coverage; test or attack outcomes never supply the reference. Curves use individual completion times, not aggregate G_T.

Motion qualification. Geometry is measured on executed end-effector positions in a common physical frame, separately for approach, transport and placement, anchored by contact/grasp/lift/place events. For position x_k and displacement $\Delta x_k = x_{k+1} - x_k$, project x_k onto the clean phase polyline (earliest arc position breaks ties); g_k is the point 20 mm farther along it, clipped to its endpoint. For phase indices $\mathcal{K}$, define path length P and reverse-distance fraction B, with $\mathbf{1}$ an indicator:

$$P = \sum_{k\in\mathcal{K}} \|\Delta x_k\|_2, \qquad B = \frac{\sum_{k\in\mathcal{K}} \|\Delta x_k\|_2 \mathbf{1}[\Delta x_k^\top (g_k - x_k) < -\epsilon_b]}{\max(P, \epsilon_P)},$$

with $\epsilon_b = 10^{-8}\,\mathrm{m}^2$ and $\epsilon_P = 10^{-6}\,\mathrm{m}$. This evaluation direction is distinct from Axis A's fixed training target. Path-ratio tolerance is initially $\pm 15\%$ and reverse-motion increase at most five percentage points.

Phase polylines are resampled every 5 mm. Directed attacked-to-clean deviation has initial mean/maximum bounds of 20/50 mm. Curvature is turning angle divided by mean adjacent segment length, excluding segments below 1 mm; maximum and 95th-percentile curvature are bounded by the larger of 1.5 times the clean value and the corresponding 95th percentile of repeated clean validation runs. Motion below 1 mm over 20 control steps is stagnation; its additional fraction is bounded by five percentage points, excluding native settling and task-required holds. Ordered manipulation events must match the task, and empty phases are undefined. These tolerances are frozen before test. Motion summaries use paired completed paths. Mean path deviation (MPD) averages arithmetically and positive ratios geometrically; clean curvature below $1\,\mathrm{m}^{-1}$ uses absolute difference and coverage rather than an unstable curvature ratio. Validation applies the motion bounds to these aggregate summaries; JAS and the behavioural-violation rate (BVR, defined below) also count per-trial violations.

Joint success. Let $\mathcal{C}_0$ be the fixed clean-success test cohort. Thresholds τ_N, τ_ℓ are the larger of 1.10 and the respective 95th-percentile clean-repeat ratio. Define

$$J_v = \mathbf{1}[\text{completed, valid, motion-qualified}]\mathbf{1}[N_v^\delta/N_v^c \geq \tau_N]\mathbf{1}[\bar{\ell}_v^\delta/\bar{\ell}_v^c \geq \tau_\ell],$$
$$\mathrm{JAS} = \frac{100}{|\mathcal{C}_0|} \sum_{v\in\mathcal{C}_0} J_v.$$

The primary operating point permits at most ten percentage points of TSR loss, with absolute TSR at least 50% and missing-action rate at most 5%. BVR is the percentage of $\mathcal{C}_0$ failing completion,

validity or motion qualification. Five clean validation repetitions per layout calibrate variability; checkpoint selection uses one fixed validation seed per layout. JAS diagnoses the joint mechanism, not the native success of attacks with different goals.

Spatial trajectory similarity. Following reference-length nDTW (Ilharco et al., 2019), clean and patched polylines are resampled at 5 mm arc spacing, retaining endpoints and reversals and removing only consecutive zero-length segments. No independent geometric alignment is fitted. For sequences $\mathbf{C}_v = (\mathbf{c}_1, \ldots, \mathbf{c}_{m_v})$ and $\mathbf{P}_v = (\mathbf{p}_1, \ldots, \mathbf{p}_{n_v})$, with m_v, n_v points in metres, define warping distance D_v and per-trial similarity s_v:

$$D_v = \min_{W} \sum_{(i,j)\in W} \|\mathbf{c}_i - \mathbf{p}_j\|_2, \qquad s_v = \exp[-D_v/(m_v d_{\text{geo}})], \qquad \text{nDTW} = |\mathcal{C}_0|^{-1} \sum_{v\in\mathcal{C}_0} s_v.$$

The matching path W runs from $(1,1)$ to (m_v, n_v) using steps $(1,0), (0,1), (1,1)$. The shared scale is $d_{\text{geo}} = 0.020\,\text{m}$, with sensitivity at 0.010 and 0.040 m. Failed or capped attacks retain their full observed paths against the complete clean reference; missing usable output scores zero and is counted separately. Instrumentation failures follow a common rerun policy. Clean self-comparison is one by identity. nDTW measures spatial preservation, not perceptual naturalness, timing invisibility, or task success. Repeated-clean variation and motion diagnostics contextualise it.

Paired wall-time change. For condition $m \in \{c, \delta\}$, $T^m_{\text{wall},v}$ is elapsed time from first detected arm motion to native task success, using a common onset rule. It includes movement, inference and waits with any overlap, but excludes reset/loading and waiting before motion. If $\mathcal{P}_u$ contains successful pairs for task u and $\mathcal{U}_+$ the tasks with nonempty $\mathcal{P}_u$, then

$$\Delta T_{\text{wall}} = \frac{1}{|\mathcal{U}_+|} \sum_{u\in\mathcal{U}_+} \frac{1}{|\mathcal{P}_u|} \sum_{v\in\mathcal{P}_u} (T^{\delta}_{\text{wall},v} - T^{c}_{\text{wall},v}).$$

This is task-balanced added seconds, not G_T times a clean duration. Table 13 reports paired intervals and completed-pair counts for the main comparisons. Eligible pairs can differ across methods; the SPADE/Direct G_T ratio in that table uses the shared completed cohort. Failures remain in TSR and deadline curves.

D Essential Experimental Protocols

The following settings define the comparisons retained in the main paper and Section E. They separate changes in attack access from changes in objective, task or visual conditions.

Baselines and component controls. Our state-access adaptation, POV-SA, retains POV's localisation, semantic and trajectory objectives but uses the common training-state access, rather than only POV's initial observed prefix (Wang et al., 2026). Universal Patch Attack via Robust Feature, Attention, and Semantics (UPA-RFAS) retains its feature, attention, semantic and robust optimisation modules, using the frozen target as its white-box surrogate and fitting one patch per task (Lu et al., 2026); this is not its original universal cross-model setting. Both retain their native objectives. Direct patch applies the Stage-I losses directly to pixels with the same adapters, refresh, appearance objective and full budget, without teacher search. Patch support, initial-state splits, native caps, hardware and validation allowances are shared. Source attacks use their native objective for selection; SPADE/Direct use the declared slowdown feasibility rule.

Axis A only sets $\lambda_B = 0$ and omits teacher latency and reasoning-length gain requirements; Axis B only removes $\mathcal{L}_A$ and the teacher call-count gain requirement. Both retain task, motion and action-validity bounds, budget and validation data. Distillation variants share the full teacher within each task and attack seed. Removing boundary or action imitation sets λ_g or λ_a to zero. Sequence imitation fits complete teacher reasoning and native actions by cross-entropy at the same accepted states, retaining appearance and search costs but omitting explicit boundary matching and student-prefix feedback. It compares configurations, not one loss.

Patch checkpoints use SPADE's failure-penalised validation score and common task/validity bounds. If none qualifies, evaluate the highest-scoring checkpoint and record failure; final joint-

gain acceptance does not filter ablation rows. Failed teacher searches remain counted. Test metrics follow Section C, including the unchanged two-axis JAS definition.

Content and feedback-data ablations. The final two rows of Table 8 isolate two further parts of the transfer design using the same qualified, frozen teacher as the full method. No content loss sets $\lambda_c = 0$, retaining action, boundary and appearance objectives. Fixed feedback data restricts reasoning imitation to the initial teacher state/prefix bank: it disables student-prefix sampling and student-visited-state augmentation. Action-context recomputation, loss weights, rendering, budget and validation remain unchanged. This tests the combined feedback configuration, rather than attributing its effect separately to text and state refresh. Full SPADE in Table 3 is the reference.

Minimal Axis-B controls. Table 5 varies the two body regularisers while retaining Axis A, the exit margin, initialisations, candidate weighting, native cap, refresh and total budget. Every variant uses the same patch-imitation procedure. Qualification is the fraction of task–seed searches yielding a qualified teacher within budget (40 tasks, three seeds). Teacher diagnostics use five fixed clean-validation observations per task and seed, with the selected teacher or the highest-scoring checkpoint if qualification fails. Return is the fraction of these native greedy calls that produce valid actions before the cap; G_L^S is the geometric-mean teacher/clean reasoning-length ratio on jointly valid calls with nonempty clean reasoning. Patch metrics use held-out layouts under Section C; failed searches remain in the accounting.

Table 5: **Axis-B regularisation and teacher-to-patch transfer.** White-box DeepThinkVLA–Standard LIBERO. Only λ_u and λ_{ref} vary; Axis A and the exit-margin term remain active. Teacher diagnostics use validation data; final patches use held-out layouts. Length/cost ratios: $\times$; return, qualification, TSR and JAS: %; nDTW: $[0, 1]$.

(a) Teacher: length, valid return and qualification

Teacher objective	λ_u	λ_{ref}	G_L^S	Return	Qualification
Full Axis B	0.1	0.1	4.60	98.7	96.7
Without uncertainty	0	0.1	3.30	99.2	95.0
Without reference	0.1	0	6.85	94.8	88.3
Margin only	0	0	5.75	96.3	91.7

(b) Final patch: cost and task preservation

Teacher objective	G_N	G_ℓ	G_T	TSR	nDTW	JAS
Full Axis B	1.53	2.60	3.98	89	0.60	77
Without uncertainty	1.54	2.16	3.33	90	0.61	63
Without reference	1.49	1.78	2.65	79	0.54	43
Margin only	1.51	1.93	2.91	83	0.57	51

Without reference guidance, teacher length rises to $6.85\times$, yet patch per-call inflation falls to $1.78\times$, compared with $2.60\times$ for the full objective. Removing uncertainty produces shorter reasoning but slightly higher TSR. These contrasts separate teacher length from usable transfer. They do not identify text degradation as the sole cause: the controls also change candidate distributions and optimisation dynamics. Isolating that cause requires additional text-quality diagnostics and teacher-length-matched comparisons.

Candidate-refitting control. For nonempty $\mathcal{J}_B$, let $Y_j = (y_{j,1}, \ldots, y_{j,n_j})$ contain candidate reasoning and its complete native exit delimiter, with token count n_j and prefix $y_{j,<i} = (y_{j,1}, \ldots, y_{j,i-1})$. The outcome-filtered sequence-fitting control is

$$\ell_j^{\text{seq}} = -\sum_{i=1}^{n_j} \log Q_S(y_{j,i} \mid O, I, y_{j,<i}),$$

$$\mathcal{L}_B^{\text{refit}} = \frac{\sum_{j\in\mathcal{J}_B} \omega_j \ell_j^{\text{seq}}}{\sum_{j\in\mathcal{J}_B} \omega_j n_j} + \lambda_{\text{ref}} \sum_{j\in\mathcal{J}_B} \omega_j \mathcal{L}_{\text{ref},j}.$$

It fits both reasoning tokens and observed exits, replacing the teacher's margin and uncertainty terms. Axis A, candidate weights, qualification and patch imitation are unchanged; the comparison uses the same total budget and reference weight.

Cross-task transfer. The first two task IDs in each fixed Standard LIBERO suite manifest give eight tasks and 56 ordered distinct source–destination pairs. Long denotes LIBERO-10. Each source patch is trained independently, then frozen with its reference, rendering and model for all seven destinations. Each destination uses five held-out layouts and three paired repeats per attack seed. Neither destination feedback nor best-source selection is allowed. The patch remains on a homologous stationary support; unsupported scenes are declared before observing outcomes and remain in coverage counts. Means weight task pairs equally: success rates arithmetically and cost ratios geometrically on common completed pairs. Reused clean runs are identified rather than counted as independent evidence. For each pair, nDTW and JAS use the destination clean-success cohort, retaining attacked failures; pair-level scores are then averaged equally. Undefined clean support is reported with coverage. Table 9 distinguishes total-cost transfer from preservation of both cost factors and destination-path geometry.

D.1 Proxy-Assisted Black-Box Setting

This setting separates gradient computation on the proxy from candidate assessment through public target feedback. We use disjoint optimisation, validation, and test layouts.

The frozen ECoT-OpenVLA-7B proxy uses the public Open X-Embodiment checkpoint and is adapted with public Standard LIBERO training demonstrations before attack fitting (Zawalski et al., 2024). It is distinct from all target checkpoints, including the ECoT/MiniVLA-1B target; the shared ECoT recipe does not make the 7B proxy the same checkpoint as the MiniVLA target. The target reveals only public observations, decoded actions, call completion, timing and task outcomes. External poses support motion checks; reasoning counts are used only if public. Target weights, gradients, logits and hidden states are inaccessible. White-box target artefacts cannot initialise black-box runs.

Each run has 12 proxy GPU-hours and 20,000 target policy calls. All proxy search and patch updates, clean calibration, failed returns, retries and selection are charged; held-out evaluation is separate. For a fixed nonempty set $\mathcal{U}_0$ of up to five clean-successful training layouts selected by manifest order, let q_v use the completed/valid/motion-qualified log cost ratio and $-\log 2$ failure penalty defined above. The public-feedback score is

$$F(\delta) = |\mathcal{U}_0|^{-1} \sum_{v \in \mathcal{U}_0} q_v - \lambda_v \mathcal{L}_{\text{vis}}(\delta; j),$$

with fixed appearance reference j. All candidates use the same layouts and seeds. Eight independently seeded blocks each make ten proxy updates; proxy scoring selects up to four candidates for complete target rollouts. A candidate replaces the incumbent only if F improves and then seeds the next round. No derivative passes through target feedback, and the target never receives the suffix. This proposal/assessment separation is motivated by PAL (Sitawarin et al., 2024), without using its token-probability API loss.

Ten disjoint validation layouts select at budget fractions $\{0.25, 0.5, 0.75, 1\}$ using 20% of target calls, enforcing TSR, validity and motion bounds. Incomplete assessments cannot select a patch but consume queries. Optimisation stops at either cap, and failed seeds remain in accounting. Proposal ranking uses total cost without requiring both axes to improve at every update. Final dual-axis qualification additionally requires both calibrated gains and preservation bounds; JAS measures this event on test trials. Public latency is not relabelled as pure device time, and hidden token counts are not inferred from it.

D.2 Scene Conditions, Appearance and Physical Evaluation

These studies vary rendering, appearance, and deployment conditions to examine where the nominal attack remains effective and how its visual footprint changes.

Area and budget comparisons. Each area receives an independent patch fit with the nominal settings and total budget; placement follows the same rule. The default area is fixed before testing.

At the nominal area, completion curves follow Section C; the reporting budget is three times the clean reference, shared across methods.

Scene shifts. Freeze the nominal patch and vary one family at a time. Illumination factors are $1 \pm \{0.1, 0.2, 0.3\}$, camera yaw is $\pm\{3, 6, 10\}^\circ$, and one, two or three non-target surface materials change using held-out assets. Gaussian RGB noise has standard deviations $\{0.01, 0.03, 0.05\}$ on $[0, 1]$ pixels, followed by clipping. Positive/negative lighting and yaw cases receive equal trial counts; texture and noise realisations follow fixed seed manifests. Clean and patched runs share each realisation, with the same patch surface reprojected after camera changes. These are selected LIBERO-Plus-inspired controlled conditions (Fei et al., 2026), distinct from the full official evaluation in Table 6; that evaluation also covers object layout, language and robot-initial-state variation.

For each displayed condition ζ, fix its shifted-clean-success cohort $\mathcal{C}_0^\zeta$ before inspecting attack outcomes. Apply the JAS definition using this cohort and its paired shifted clean costs and paths. Thresholds and motion tolerances remain those calibrated under nominal validation conditions. Thus JAS is conditional on clean success within each shift; Table 12 reports clean and patched TSR over all trials to expose changes in eligible coverage. Signed cases and seeded realisations are pooled with equal trial allocation, without selecting the best case.

Visual conspicuousness. Patch-crop LPIPS uses AlexNet features against the same benign reference, independently of the VGG training loss (Zhang et al., 2018). The human-evaluation protocol contains 40 scenes (one per LIBERO task), three conditions and five ratings per image: 30 blinded raters each see 20 images, with no repeated base scene per rater. Exposure is two seconds, order is randomised, and the binary judgement asks whether a visually suspicious or manipulated region is present. Flag rates average by scene, with benign prints as the false-positive reference; intervals in Table 13 resample scenes and raters, with contrasts paired by scene. This evaluates appearance rather than inferred malicious intent. Cost and TSR accompany visual scores. Figure 5 shows a separate DeepThinkVLA–Standard LIBERO case: refining an initial patch towards a fixed rose with $\mathcal{L}_{\text{vis}}$ gives $G_T \approx 3.98$ and TSR $\approx 89\%$ on closed-loop re-evaluation. Area is 14.1%; images share a display scale without cosmetic postprocessing. Rose structure and residual texture remain visible. This example is distinct from the aggregate results in Table 11.

Physical protocol. DeepThinkVLA's hardware-adapted single-view checkpoint and action normaliser are frozen before attack fitting. A Franka Panda with parallel gripper and front/side Intel RealSense D415 cameras follows Huang et al. (2026); only the front view drives the policy. An A100 80 GB runs bfloat16, batch one and greedy reasoning (1,024-token cap), with ten-command chunks at 20 Hz and five inference-only warm-up calls before timing.

The matte print is 10×10 cm excluding borders. Camera–robot calibration is fixed, without learned colour correction or physical patch reoptimisation. Occlusion covers $15\% \pm 5\%$ of the front-view patch projection, leaving the orange visible. Ten held-out layouts and five paired repetitions yield 50 trials per input/visibility condition, 300 across three inputs and two visibility conditions. Matched blocks randomise input order. Benign prints use the unoptimised rose with matching size, paper and placement; their G_T/TSR are 1.03/92% unoccluded and 1.02/90% occluded. The raw patch is a photographic comparison only.

Success requires a 5 cm lift held for two seconds; exceeding 300 commands or 180 s counts as failure. Call timing spans preprocessing to action return with device synchronisation; wall timing starts at first motion. Visibility-matched clean runs define ratios. The 95% percentile intervals in Table 13 use 2,000 bootstrap resamples of layouts, then paired repetitions within layouts, preserving condition pairs. TSR includes all trials; cost ratios use completed pairs, with coverage in Table 14.

E Supporting Results

These results complement the main tables with the second benchmark, complete cost-factor and ablation results, and spatial and visual preservation diagnostics.

Table 6: **Frozen-patch results on LIBERO-Plus.** All 10,030 variants; white-box patches optimised on Standard LIBERO, with no Plus optimisation. G_T: inference-time ratio; TSR: %; nDTW: $[0, 1]$. †Common-protocol adaptations of the source attacks.

	DeepThinkVLA			CoTinyVLA			ECoT		
Method	G_T	TSR	nDTW	G_T	TSR	nDTW	G_T	TSR	nDTW
Clean	1.00	72	1.00	1.00	76	1.00	1.00	64	1.00
Benign print	1.03	71	0.80	1.03	75	0.82	1.03	63	0.77
POV-SA†	2.03	18	0.12	1.94	20	0.11	1.78	19	0.16
UPA-RFAS†	1.69	13	0.17	1.73	19	0.18	1.53	15	0.14
Direct patch	2.21	65	0.41	1.95	69	0.44	1.59	57	0.38
SPADE	**3.12**	**66**	**0.57**	**2.87**	**70**	**0.59**	**2.48**	**58**	**0.52**

Table 7: **Cost-factor comparison with direct optimisation.** Direct patch and white-box SPADE share the total optimisation budget; black-box SPADE uses the proxy-assisted extension. Plus rows reuse the Standard LIBERO patches. Ratios use common completed pairs within each row; TSR includes all trials. Unrounded factors satisfy $G_T = G_N G_\ell$.

Benchmark	Target	Method / access	G_N	G_ℓ	G_T	TSR	nDTW
Standard LIBERO	DeepThinkVLA	Direct (White-box)	1.61	1.87	3.01×	87%	0.44
		SPADE (White-box)	1.53	2.60	3.98×	89%	0.60
		SPADE (Black-box)	1.39	2.23	3.10×	90%	0.57
Standard LIBERO	CoTinyVLA	Direct (White-box)	1.56	1.62	2.53×	88%	0.47
		SPADE (White-box)	1.48	2.35	3.48×	90%	0.63
		SPADE (Black-box)	1.35	2.04	2.75×	91%	0.59
Standard LIBERO	ECoT	Direct (White-box)	1.48	1.42	2.10×	79%	0.40
		SPADE (White-box)	1.40	2.15	3.01×	80%	0.56
		SPADE (Black-box)	1.29	1.86	2.40×	81%	0.53
LIBERO-Plus	DeepThinkVLA	Direct (White-box)	1.50	1.47	2.21×	65%	0.41
		SPADE (White-box)	1.42	2.20	3.12×	66%	0.57
		SPADE (Black-box)	1.33	1.98	2.63×	68%	0.54
LIBERO-Plus	CoTinyVLA	Direct (White-box)	1.48	1.32	1.95×	69%	0.44
		SPADE (White-box)	1.40	2.05	2.87×	70%	0.59
		SPADE (Black-box)	1.30	1.83	2.38×	72%	0.56
LIBERO-Plus	ECoT	Direct (White-box)	1.42	1.12	1.59×	57%	0.38
		SPADE (White-box)	1.34	1.85	2.48×	58%	0.52
		SPADE (Black-box)	1.25	1.69	2.11×	60%	0.50

Table 8: **Complete component ablations.** White-box DeepThinkVLA–Standard LIBERO; final patches. Axis-only variants change the teacher; distillation variants share it. Cost factors use common completed pairs within each row; TSR uses all trials and JAS the clean-success cohort. Ratios: ×; TSR/JAS: %. Variant definitions follow Section D.

Variant	G_N	G_ℓ	G_T	TSR	JAS
Full SPADE	1.53	2.60	3.98	89	77
Axis A only	1.57	1.04	1.63	90	4
Axis B only	1.04	2.70	2.81	89	6
Sequence imitation	1.36	2.00	2.72	85	49
No boundary loss	1.50	1.32	1.98	90	26
No action loss	1.17	2.73	3.19	64	19
No content loss	1.46	2.75	4.01	73	35
Fixed feedback data	1.35	1.89	2.55	84	44

Table 9: **Cross-task transfer factors and preservation.** Same target policy, distinct source and destination tasks. Access: source-task fitting. Cost factors use common completed pairs; nDTW retains failed attacks in the destination clean-success cohort. TSR/JAS: %; nDTW: $[0, 1]$.

Target	Access	G_N	G_ℓ	G_T	TSR	nDTW	JAS
DeepThinkVLA	White-box	1.28	1.92	2.46×	89%	0.56	41.8%
	Black-box	1.22	1.67	2.04×	90%	0.53	31.9%
CoTinyVLA	White-box	1.23	1.68	2.07×	90%	0.59	40.0%
	Black-box	1.17	1.49	1.74×	91%	0.55	31.0%
ECoT	White-box	1.25	1.80	2.26×	81%	0.51	36.0%
	Black-box	1.20	1.58	1.89×	82%	0.48	28.0%

Table 10: **Motion diagnostics on DeepThinkVLA–Standard LIBERO.** Path: approach path-length ratio; MPD: mean deviation (mm); C_{95}: 95th-percentile curvature ratio. Reverse/stagnation: added percentage points; BVR: %, including failed trials.

Method	Path	MPD	C_{95}	Reverse	Stagnation	BVR
Benign print	1.01	4.8	1.03	0.3	0.4	3.1
POV-SA	1.48	61.3	2.91	17.2	10.4	77.2
UPA-RFAS	1.31	48.5	2.36	12.8	14.2	81.5
Direct patch	1.09	18.7	1.23	3.0	3.3	21.8
SPADE	1.04	12.6	1.11	1.3	1.1	12.4

Table 11: **Appearance and retained efficacy.** DeepThinkVLA–Standard LIBERO; LPIPS is patch-crop distance to the benign reference. Flag is the blinded suspicious-region rating percentage; TSR and JAS are percentages.

Condition	LPIPS	Flag	G_T	TSR	JAS
Benign print	0.00	12	1.03	93	1
No appearance loss	0.48	78	4.43	87	70
Content + TV	0.14	21	3.98	89	77

(a) Reference (b) Initial patch (c) After $\mathcal{L}_{\text{vis}}$



Figure 5: **Reference-guided patch appearance.** DeepThinkVLA–Standard LIBERO example. (a) Fixed benign reference; (b) initial attack patch; (c) patch after optimisation with $\mathcal{L}_{\text{vis}}$. Patch area is 14.1% of the external view. Images are displayed at equal scale without postprocessing.

Table 12: **Paired success under scene shifts.** DeepThinkVLA–Standard LIBERO. Each cell gives clean/patch TSR (%); corresponding JAS is shown in Figure 3c.

Shift	Mild	Medium	High
Light	93/88	91/85	87/80
Texture	93/88	90/84	86/78
Noise	92/87	89/83	83/75
Camera	92/86	87/80	79/70

Table 13: **95% bootstrap intervals for the main comparisons.** Point estimates with bracketed intervals from 2,000 resamples. Standard LIBERO uses task/layout/repetition/attack-seed resampling (Section C); physical and flag-rate protocols follow Section D.2. G_T and ΔT_{wall} use completed clean/patch pairs; nDTW uses the clean-success cohort, retaining attacked failures; TSR uses all trials. Pairs: completed/total, except ratio rows, which give the shared completed count.

Method	Pairs	G_T (×)	ΔT_{wall} (s)	TSR (%)	nDTW
DeepThinkVLA, Standard LIBERO (clean TSR 94%)					
Direct patch	5,036/6,000	3.01 [2.78, 3.29]	19.1 [16.9, 21.6]	87 [83, 90]	0.44 [0.41, 0.48]
SPADE	5,243/6,000	3.98 [3.66, 4.42]	27.1 [24.1, 30.9]	89 [85, 91]	0.60 [0.56, 0.63]
SPADE (black-box)	5,296/6,000	3.10 [2.84, 3.51]	19.3 [16.9, 22.3]	90 [86, 93]	0.57 [0.53, 0.60]
SPADE / Direct	4,733	1.32 [1.23, 1.43]	—	—	—
SPADE drop (pp)	—	—	—	5 [3, 8]	—
CoTinyVLA, Standard LIBERO (clean TSR 95%)					
Direct patch	5,122/6,000	2.53 [2.32, 2.76]	14.9 [13.1, 17.4]	88 [84, 91]	0.47 [0.42, 0.51]
SPADE	5,327/6,000	3.48 [3.23, 3.85]	21.7 [19.0, 24.8]	90 [87, 92]	0.63 [0.60, 0.67]
SPADE (black-box)	5,383/6,000	2.75 [2.48, 3.10]	15.4 [13.3, 17.5]	91 [87, 93]	0.59 [0.56, 0.64]
SPADE / Direct	4,877	1.38 [1.30, 1.49]	—	—	—
SPADE drop (pp)	—	—	—	5 [2, 8]	—
ECoT, Standard LIBERO (clean TSR 86%)					
Direct patch	4,530/6,000	2.10 [1.92, 2.30]	12.1 [10.8, 13.7]	79 [75, 82]	0.40 [0.36, 0.43]
SPADE	4,636/6,000	3.01 [2.74, 3.31]	19.6 [17.2, 22.8]	80 [76, 84]	0.56 [0.52, 0.59]
SPADE (black-box)	4,705/6,000	2.40 [2.20, 2.73]	13.8 [12.4, 15.9]	81 [77, 84]	0.53 [0.49, 0.57]
SPADE / Direct	4,338	1.43 [1.34, 1.55]	—	—	—
SPADE drop (pp)	—	—	—	6 [3, 9]	—
Physical deployment, DeepThinkVLA (50 trials per condition)					
Unoccluded	44/50	4.00 [3.57, 4.80]	38.0 [32.8, 45.4]	90 [78, 96]	—
Partly occluded	42/50	2.94 [2.58, 3.52]	25.0 [21.7, 30.9]	86 [74, 94]	—

SPADE / Direct reports the geometric-mean cost ratio on trials where clean, SPADE and Direct all complete. TSR drop is clean minus white-box SPADE in percentage points (tolerance: 10). Physical clean TSR is 94% unoccluded and 92% partly occluded. Dashes denote metrics not reported for that comparison.

Blinded flag rate, DeepThinkVLA (40 scenes, 30 raters) Condition	Flag (%)
No appearance loss	78 [70, 84]
Content + TV	21 [15, 28]
Benign print	12 [9, 18]
Reduction: no appearance loss − Content + TV (pp)	57 [50, 66]
Residual: Content + TV − benign print (pp)	9 [4, 16]

Flag-rate contrasts are paired by scene; pp denotes percentage points.

F PHYSICAL DEPLOYMENT EXAMPLES

Physical outcomes and deployment views complement the simulation study.

Table 14: **Physical availability with the refined print.** DeepThinkVLA; 50 trials per condition. Ratios: $\times$; ΔT_{wall}: s; TSR: clean/patch (%); pairs: completed/total.

Visibility	G_N	G_ℓ	G_T	ΔT_{wall}	TSR	Pairs
Unoccluded	1.60	2.50	4.00	38.0	94/90	44/50
Partly occluded	1.40	2.10	2.94	25.0	92/86	42/50

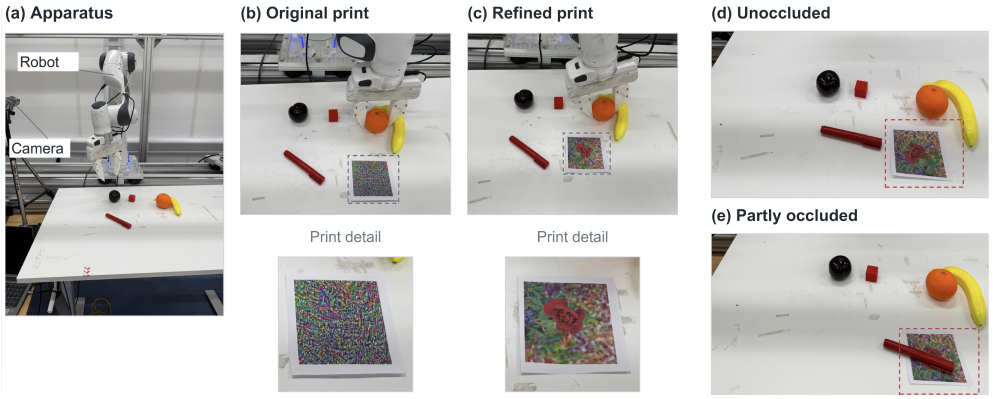


Figure 6: **Apparatus, print appearance, and visibility.** (a) Robot and camera. (b,c) Original/refined prints with details cropped from the corresponding photographs. (d,e) Unoccluded and partly occluded placement; dashed boxes locate the print.

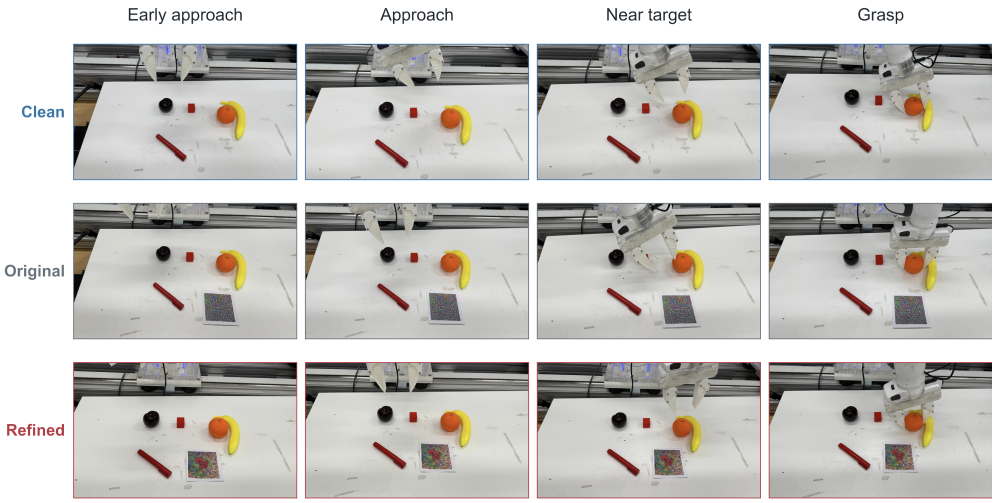


Figure 7: **Physical execution under three visual conditions.** "Pick up the orange" with clean input, original patch and refined patch. Columns group motion phases, not matched times, calls or initial poses.