FIPAS: FACTOR-INVARIANT AND POSITION-AWARE ACTIVATION STEERING FOR ROBUST VLA MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies can lose closed-loop reliability under visual changes that preserve the intended task. Post-hoc activation steering can adapt a frozen policy, but success–failure contrasts may entangle environmental variation with action error and task progress. We introduce FIPAS, which estimates low-rank environment-correlated subspaces from behavior-matched clean–perturbed activations, protects an estimated task-predictive subspace, and conditionally corrects abnormal residuals at calibrated layer–action-query sites. A gate based on current or past rollout signals attenuates intervention during sensitive manipulation stages. In a 50-task RoboTwin aggregate, FIPAS raises randomized success from 20.88% to 36.90% on the same clean500 policy, while clean success changes from 85.40% to 86.34%. In a 10-task factor-controlled ManiSkill3 aggregate, gains under isolated and composed shifts range from 5.83 to 9.40 percentage points, with a -0.21-point clean change. These reported aggregates support selective hidden-state correction as a complement to clean-data scaling under calibrated visual shifts.

1 INTRODUCTION

Vision-language-action (VLA) policies transfer semantic priors from large vision-language models to robot control and can generalize across objects and instructions (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2025). Related generalist robot-policy work studies scaling and transfer across tasks and robot embodiments (Brohan et al., 2023; Open X-Embodiment Collaboration, 2024; Ghosh et al., 2024). Yet semantic generalization does not guarantee robustness in closed-loop control. A policy may succeed in a familiar scene but fail when visual domain-randomization factors change. We focus on controlled visual distribution shifts that preserve the task instruction, the physical states of the robot and objects, the target geometry, and the intended action, while changing the background, object surface appearance, illumination, or sensor imaging statistics. Even when the desired behavior is unchanged, these observation shifts may lead the policy to produce different actions. This raises the central question of this paper: without updating policy weights, how can we reduce the effect of visual shifts on closed-loop actions while preserving performance in the original domain?

Increasing the number of clean demonstrations can strengthen a policy's competence under nominal conditions without necessarily eliminating its sensitivity to distribution shift. The original RoboTwin 2.0 study reports a substantial gap between clean and domain-randomized evaluation after training on clean demonstrations (Chen et al., 2025). Separately, RobustVLA's real-robot data-scaling experiment shows that the perturbed-condition success rate of its base π_0 policy increases from 37.5% with 25 demonstrations to 65.0% with 100 demonstrations (Guo et al., 2026). In our 50-task RoboTwin evaluation, increasing clean demonstrations per task from 50 to 500 raises clean success from 67.96% to 85.40% and randomized success from 6.62% to 20.88%. Nevertheless, the absolute clean-to-randomized gap remains large, changing from 61.34 to 64.52 percentage points. This motivates a complementary question: once a task-capable policy is frozen, can we correct environment-induced changes in its internal computation without retraining the model or removing task-relevant information?

Activation steering is appealing because it modifies intermediate representations without updating model weights. In language models, ActAdd derives steering directions from contrasting prompt pairs (Turner et al., 2023), while CAA averages activation differences between positive and negative

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.

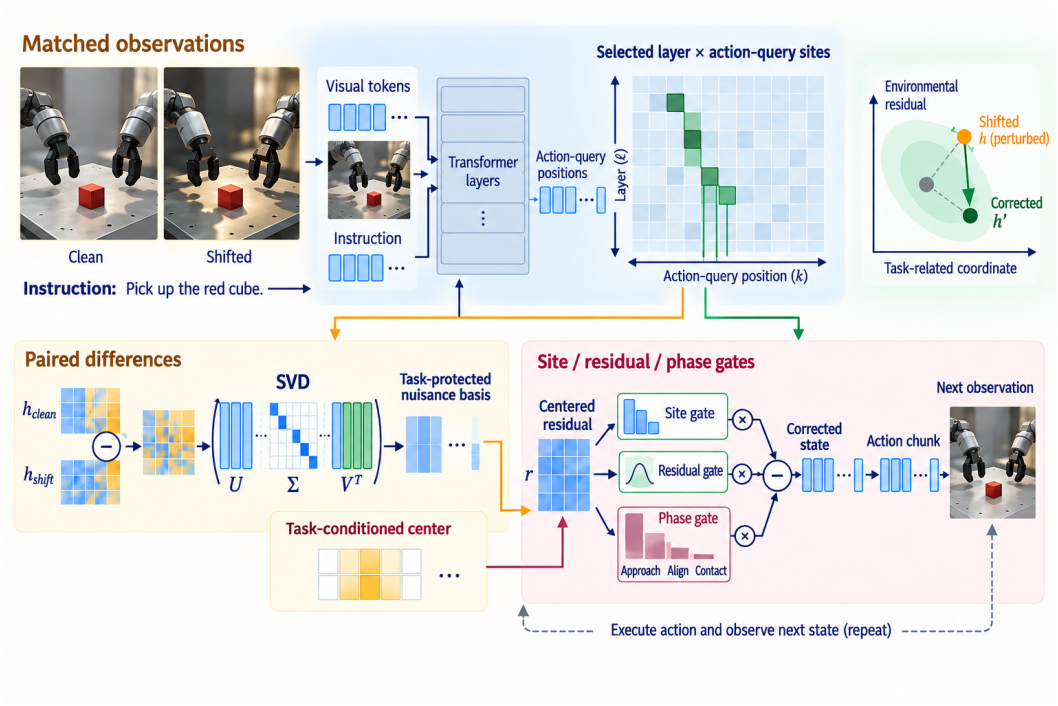

Figure 1: **Overview of FIPAS.** Behavior-matched clean and random observations preserve the instruction, robot state, and intended action while changing the visual environment. For a frozen VLA policy, candidate intervention sites are indexed by transformer layer and action-query position. Offline, paired activation differences define a low-rank environment-correlated subspace, task-predictive directions are protected, and clean rollouts calibrate task-conditioned reference centers and residual thresholds. Online, site, residual, and causal rollout-state gates modulate hidden-state correction before action-chunk decoding. The robot scenes are illustrative renders, and the representation-space inset is schematic rather than a measured embedding.

behavioral examples (Rimsky et al., 2024). CAST conditions whether steering is applied on the input context (Lee et al., 2025), whereas DSAS adapts intervention strength across inputs and layers (Ferrando de las Morenas et al., 2026). In robotics, COAST estimates subspaces associated with task outcomes from successful and failed rollouts (Miao et al., 2026). However, an outcome-based contrast does not isolate environmental variation: successful and failed trajectories may also differ in physical state, task progress, contact events, action errors, and recovery behavior. Thus, an activation direction associated with task outcome need not be the direction induced by a visual shift when the intended behavior is held fixed.

To address this gap, we propose FIPAS. For the same task and a matched physical state, we construct clean and randomized observation pairs and compare the hidden activations produced by a frozen VLA policy. FIPAS uses these differences to estimate a low-rank subspace associated with environmental variation, then removes directions that are linearly predictive of task information in clean data. During online execution, the method uses a task-conditioned clean representation as a reference and scales the correction according to how far the residual deviates from the clean distribution. The model layer and action-query position determine *where* in the hidden state to intervene; a gate based only on current or past observable rollout states determines *when* to modulate the intervention during closed-loop execution. An action-query position is not itself treated as a physical phase of robot execution. We therefore test whether such conditional representation correction improves task success under calibrated visual shifts and how it affects performance in clean scenes (Figure 1).

Our contributions are:

- **A controlled contrast for environmental steering.** We estimate environment-induced representation changes from behavior-matched clean–perturbed pairs rather than success–failure labels.
- **A selective, task-preserving intervention.** FIPAS protects task-predictive directions and combines task-conditioned centering, site selection, residual gating, and causal rollout-state gating in a frozen policy.
- **A two-benchmark robustness evaluation.** We report clean and shifted aggregates on 50 RoboTwin and 10 factor-controlled ManiSkill3 tasks, and distinguish these primary results from earlier subset diagnostics.

2 Related Work

Robustness and generalization of VLA policies. Visual robustness in VLA policies involves both measuring closed-loop performance under distribution shifts and reducing sensitivity to them. LIBERO-Plus evaluates policies under controlled perturbations (Fei et al., 2026), while RoboTwin 2.0 evaluates bimanual manipulation under clean and domain-randomized conditions (Chen et al., 2025). Some benchmark shifts also change object layout, initial robot state, or table height; their full evaluation results therefore cannot be interpreted as the effect of visual observations alone at an identical physical state. RobustVLA improves multimodal robustness through input consistency and worst-case output optimization (Guo et al., 2026), and STRONG-VLA separates perturbation robustness learning from realignment with clean task behavior (Xie et al., 2026). By comparison, BYOVLA keeps the policy fixed and edits task-irrelevant image regions at run time (Hancock et al., 2024). Cross-view action consistency renders different camera views from the same simulated state and uses training to encourage consistent actions (Huang et al., 2026). FIPAS also studies task-preserving observation changes, but estimates and corrects environment-correlated displacement inside a frozen policy's hidden states.

Activation steering and representation editing. Activation steering makes intermediate representations of a frozen model an inference-time intervention target. ActAdd and CAA derive steering directions from activation differences between contrasting inputs (Turner et al., 2023; Rimsky et al., 2024); Inference-Time Intervention further studies the selection of attention heads (Li et al., 2023). LEACE provides a linear objective for erasing decodable attributes while limiting changes to other information (Belrose et al., 2023). Whether and how strongly to intervene matters as well: CAST applies steering conditionally on the input (Lee et al., 2025), and DSAS modulates steering strength across inputs and layers (Ferrando de las Morenas et al., 2026). These studies motivate FIPAS's contrastive estimation, task-direction protection, and conditional gating. Yet the ability of a direction to change an output does not establish that it represents only environmental variation. In closed-loop control, model layer and action-query position must be distinguished from the robot's execution state; applying the same edit everywhere can disturb useful task information.

Steering internal representations of VLA models. Inference-time guidance of robot policies can act at different points in the control pipeline. V-GPS uses an additional learned value function to rerank candidate actions at selection time (Nakamoto et al., 2025). Mechanistic steering identifies VLA features associated with action attributes such as speed and direction and changes those attributes by editing activations (Häon et al., 2025). COAST builds conceptors from successful and failed trajectories to steer outcome-associated hidden representations in a frozen VLA (Miao et al., 2026). FIPAS examines a different contrast: how clean and shifted observations alter internal computation when the task, physical state, and intended behavior are matched. It estimates the environment-correlated displacement from paired activations, protects task-predictive directions, and adjusts the correction by hidden-state site and closed-loop execution state. Its intervention therefore targets the effect of task-preserving visual variation rather than action re-ranking, an individual action attribute, or a generic success-associated subspace.

Together, these lines of work show how to measure VLA sensitivity to visual changes, edit representations without updating policy weights, and guide robot behavior at inference time. They motivate three linked steps: estimate environment-correlated activation change under matched task and physical state, avoid disrupting task information, and determine the hidden states and rollout times

at which a correction should act. The next section describes FIPAS's paired contrast, task-direction protection, and conditional online intervention.

3 METHOD

Let a frozen VLA policy π_θ map an observation history $o_{\le t}$ and a language instruction ℓ_k to an action chunk. For task k, rollout step t, transformer layer l, and action-query position p, we denote the corresponding hidden state by $h_{k,t,l,p} \in \mathbb{R}^d$. FIPAS modifies selected hidden states at inference time while keeping all parameters θ frozen. Its objective is to suppress representation changes caused by task-preserving environmental variation without removing information required for the intended action.

3.1 TASK-PRESERVING ENVIRONMENTAL CONTRASTS

For each calibration state, we seek a clean observation x_i^0 and a perturbed counterpart x_i^f that differ only in environmental factor f. The target contrast preserves the task, instruction, robot and object state, target geometry, and desired action semantics. Re-rendering the same simulator snapshot is the preferred construction; identical seeds or normalized progress alone do not establish state equivalence. When exact snapshot matching is unavailable, progress bins and manipulation events yield an *approximate* match, whose state discrepancy and exclusion rate must be reported. For a history-conditioned policy, the observation histories, not just the current frames, must satisfy the matching rule.

The factor-induced activation displacement at site (l, p) is

$$\Delta^f_{i,t,l,p} = h^f_{i,t,l,p} - h^0_{i,t,l,p}. \tag{1}$$

Unlike success–failure steering, this contrast does not intentionally change the desired behavior or task outcome. It therefore provides a more direct estimate of representation variation induced by the environment. When multiple factors are separately controlled, the same contrast construction can be applied to each factor. The benchmark-specific contrasts used in this paper are specified in Section 4.

The matching rule is especially consequential for closed-loop data. Two rollouts can start from the same seed but take different actions after a visual change, so their later observations no longer isolate appearance. Pairing those later states would confound environmental displacement with differences in robot pose, object pose, and task progress. Snapshot re-rendering avoids this particular confound; approximate matching requires an explicit state distance criterion and an accounting of excluded pairs. Likewise, a change that alters graspability or occludes the target may cease to be task-preserving. Such cases should be evaluated as a distinct, harder shift instead of entering the calibration contrasts as if the intended action were unchanged. These conditions delimit what the nuisance basis can represent; they are not implied by the name of a benchmark condition.

3.2 TASK-CONDITIONED NUISANCE SUBSPACE

A global activation center can incorrectly treat differences between tasks as environmental variation. We therefore estimate a task-conditioned clean reference at every candidate site:

$$\widehat{\mu}_{k,l,p} = \rho_k \mu_{k,l,p} + (1 - \rho_k)\mu_{g,l,p}, \qquad \rho_k = \frac{n_k^{\text{eff}}}{n_k^{\text{eff}} + \lambda}, \tag{2}$$

where $\mu_{k,l,p}$ is the clean mean for task k, $\mu_{g,l,p}$ is the global clean mean, and n_k^{eff} counts independent calibration units according to a predeclared rule rather than correlated video frames. This shrinkage estimate approaches the task-specific center when sufficient calibration data are available and falls back toward the global center otherwise.

For factor f, we stack the matched differences in Eq. equation 1 into a matrix $D^f_{l,p}$ and compute a truncated singular value decomposition:

$$D^f_{l,p} = Q\Sigma V^\top, \qquad B^f_{N,l,p} = V_{[:,1:r]}. \tag{3}$$

The columns of $B^f_{N,l,p} \in \mathbb{R}^{d\times r}$ span a low-rank factor-correlated subspace. Because environmental and task information may still overlap, we construct a task-predictive basis $B_{T,l,p}$ from clean task centers and linear task-probe directions. We then remove task-predictive components from the factor basis after orthonormalizing $B_{T,l,p}$:

$$\widetilde{B}^f_{N,l,p} = \operatorname{orth}\left[\left(I - B_{T,l,p}B^\top_{T,l,p}\right) B^f_{N,l,p}\right]. \tag{4}$$

This operation reduces linear task leakage in the estimated nuisance subspace. It does not by itself guarantee behavioral preservation, which is why FIPAS additionally uses clean-calibrated position and state gates.

3.3 Position- and State-Aware Steering

Environmental information is not equally editable at every layer or action-query position. In the specified site-selection procedure, disjoint calibration rollouts are used to evaluate candidate sites on three criteria: factor decodability, task leakage, and clean action distortion. We retain a small set of sites with high factor sensitivity but low task leakage and low clean disruption. The resulting site weight is denoted by $g^f_{\mathrm{pos}}(l,p) \in [0,1]$ and is zero outside the selected sites.

At an active site, the current representation is centered using the task-conditioned clean reference:

$$z_{k,t,l,p} = h_{k,t,l,p} - \widehat{\mu}_{k,l,p}. \tag{5}$$

Its component inside the task-orthogonalized nuisance subspace is

$$n^f_{k,t,l,p} = \widetilde{B}^f_{N,l,p}\left(\widetilde{B}^f_{N,l,p}\right)^\top z_{k,t,l,p}. \tag{6}$$

We calibrate $q^f_{k,l,p}$ as a quantile of the projected residual norm $\|n^f_{k,t,l,p}\|_2$ on disjoint clean calibration data, and define a soft residual gate with $\tau > 0$:

$$g^f_{\mathrm{res}}(k,t,l,p) = \sigma\left(\frac{\left\|n^f_{k,t,l,p}\right\|_2 - q^f_{k,l,p}}{\tau}\right). \tag{7}$$

Steering strengthens as the residual exceeds its clean threshold. The sigmoid is nonzero below the threshold and equals $1/2$ at it, so the quantile, temperature, and clean false-activation rate must be reported for the actual run.

FIPAS further uses a rollout-state gate $g_{\mathrm{phase}}(s_{\le t}) \in [0,1]$, where $s_{\le t}$ may contain current or past gripper state, end-effector motion, and a deployable progress or contact estimate. Ground-truth simulator contact, final rollout length, future events, and success labels are excluded from the main-policy gate. The gate permits stronger correction during coarse approach and attenuates the intervention during fine alignment and contact-sensitive manipulation. The action-query index p represents a position inside the predicted action chunk and is not used as a proxy for physical rollout phase.

These gates answer different questions. The site weight selects where an edit is allowed, the residual gate responds to departure from the clean reference at that site, and the rollout-state gate limits when the resulting edit is appropriate for control. A low factor-decoding error alone cannot select a safe site: a feature may predict the shift while also carrying action-critical information. For the same reason, calibration, gate tuning, and final evaluation need disjoint trajectories or explicitly disclosed reuse. All state inputs to the online gate must be observable at the time of the action; retrospectively assigning phases from completed trajectories would leak future information into the intervention.

3.4 Final Intervention

Let $\star$ denote the basis, reference, and gate configuration selected on calibration/validation data and held fixed during evaluation. It can be a single estimated factor basis or a factor-blind union of calibrated bases; the latter does not require a test-time factor label. The update at a selected hidden state is

$$\begin{aligned} h'_{k,t,l,p} = h_{k,t,l,p} - \beta_\star\, g^\star_{\mathrm{pos}}(l,p)\, g^\star_{\mathrm{res}}(k,t,l,p) \\ \cdot\, g_{\mathrm{phase}}(s_{\le t})\, n^\star_{k,t,l,p}, \end{aligned} \tag{8}$$

where $\beta_\star$ controls the maximum correction strength. The intervention scales the signed projected residual toward the task-conditioned clean reference. The soft gate may also edit below its threshold, and the task-level clean mean may conflate legitimate within-task state variation with the visual shift; neither the projection nor the gate guarantees clean action preservation.

For the factor-blind union, $B_{\star,l,p}$ is obtained by concatenating the calibrated factor bases and re-orthogonalizing them; its threshold and strength are calibrated for this union rather than inherited from individual factors. A factor selector reading the test-time shift label uses extra information and is an oracle control. The union can increase effective rank, so a matched-rank comparison is necessary to attribute any benefit to factor separation. Candidate sites, ranks, thresholds, strengths, and the online rollout-state rule must be fixed on calibration/validation data before evaluation. A rank-r projection costs $O(dr)$ operations per edited site, where r is the realized union rank when applicable; total latency also depends on the number of edited sites and gate computation. The per-site edit obeys $\|h'_{k,t,l,p} - h_{k,t,l,p}\|_2 \leq \beta_\star \|n^\star_{k,t,l,p}\|_2$ when $\beta_\star \geq 0$ and all gates lie in $[0, 1]$. This local bound does not bound action differences or closed-loop trajectory error; clean preservation requires a same-checkpoint closed-loop evaluation.

4 Experimental Setup

Benchmarks and comparisons. RoboTwin 2.0 supplies 50 bimanual tasks under `demo_clean` and `demo_randomized` (Chen et al., 2025). We compare policies trained with 50 or 500 clean demonstrations per task to describe data scaling; the FIPAS comparison is reported against the same frozen clean500 policy. The randomized RoboTwin condition is used as one aggregate contrast and does not isolate visual factors. Using ManiSkill3, we define a separate 10-task, five-condition study: clean, background, object, sensor, and their composition (Tao et al., 2025). For factor-controlled calibration, separate background, object, and sensor contrasts can be fitted and combined into one factor-blind union bank. The intended ManiSkill3 test rule uses that same bank across test conditions;

Calibration and evaluation. The prescribed protocol fits clean references, environmental contrasts, site weights, and thresholds on calibration/validation rollouts disjoint from final evaluation. A matched contrast ideally re-renders the same physical state and observation history with only the visual factor changed; repeated seeds alone are insufficient. Within each policy comparison, initial state, perturbation seed, checkpoint, and evaluation budget should match. The authors report 100 trials per task and condition (5,000 per condition) for each of the three RoboTwin arms. The supplied archive reproduces their aggregate totals, The ManiSkill3 source gives condition means without task-level denominators or a run-linked deployment manifest.

Implementation record. Reproducing the final intervention requires the backbone and checkpoint hash, exact activation hook and action-query positions, calibration-pair and history-matching rules, independent-unit definition for n_k^{eff}, task-basis fitting split, site score and weights, selected rank and λ, residual quantile and τ, strength $\beta_\star$, and the online inputs and timing rule for g_{phase}. These values must be linked to the reported run configuration and fixed without using final test outcomes. The supplied success tables do not encode these settings; the rank-four basis in the separate clean50 diagnostic bank is not assumed to be the basis used for the clean500 main result. We therefore distinguish the defined intervention from the execution details still requiring a manifest. The supplied planning configuration specifies StarVLA QwenOFT with a Qwen3-VL-4B-Instruct backbone and DiT-B action head, an action-head-input intervention, a rank-four nuisance basis, task-center shrinkage $\lambda = 8$, a clean residual quantile of 0.95, runtime strength $\alpha = 0.05$ (the coefficient $\beta_\star$ in our notation), a residual gate slope of 12, and a selected query fraction of 0.30. It also proposes 10 independent calibration episodes and five disjoint validation episodes per task. In particular, the configuration says that the current runtime lacks the separate rollout-state gate defined in Section 3; its implementation and run identity must be resolved before attributing the main aggregate to the full method.

Measures and evidence boundary. We use an equal-task mean of binary task success rates; under the reported 100 trials per task and condition, the pooled and equal-task means coincide arithmetically. We report the clean-minus-shift gap, within-checkpoint FIPAS-minus-frozen difference, and reduction in that gap. The task-level RoboTwin allocation does not establish measured task-level changes

Table 1: Aggregate success rates (%) on 50 RoboTwin tasks. Clean50 and clean500 denote the number of clean demonstrations per task. Gap and gap reduction are computed from displayed means (percentage points). RoboTwin means use author-reported aggregate counts (50 tasks, 100 trials per task and condition).

Training	Method	Clean	Random	Gap	Gap red.
Clean50	Frozen policy	67.96	6.62	61.34	–
Clean500	Frozen policy	85.40	20.88	64.52	–
Clean500	**FIPAS**	**86.34**	**36.90**	**49.44**	**15.08**

or contain episode and repeated-fit records. A paired task-cluster interval could be computed after obtaining verified task pairs, whereas an episode-level paired analysis requires episode linkage. Neither a general-task improvement claim nor clean non-inferiority follows from the aggregate means: the latter requires a predeclared margin and an appropriate one-sided paired interval. Matched-budget strong baselines and main-run component ablations are not in the supplied results. The confirmatory analysis is specified before taskwise case selection: it contrasts paired versus mismatched calibration, protected versus unprotected nuisance bases, selective versus global sites, and each online gate on the same clean500 checkpoint, reporting both clean and shifted success. We select representative improvement, near-zero, and degradation cases only from verified task-level records under a fixed rule. Definitions, candidate search grids, and the case-selection rule are detailed in Appendices C–E.

5 Results

5.1 FIPAS Narrows the Robustness Gap Left by Clean-Data Scaling

Table 1 shows the current author-reported aggregate summary on 50 RoboTwin 2.0 tasks (Chen et al., 2025). Increasing the number of clean demonstrations from 50 to 500 per task substantially improves nominal performance: clean success rate increases from 67.96% to 85.40%. However, randomized success rate increases only from 6.62% to 20.88%. Consequently, the clean-to-randomized gap remains large, changing from 61.34 to 64.52 percentage points. Across these two reported checkpoints, the absolute gap remains above 61 points. This is a descriptive scaling comparison; the small difference between their gaps is not a statistically established deterioration.

Applying FIPAS to the frozen clean500 policy increases randomized success rate from 20.88% to 36.90%, corresponding to a gain of 16.02 points from the displayed values. At the same time, clean success rate changes from 85.40% to 86.34%. The resulting displayed-value robustness gap is 49.44 points, yielding a displayed gap reduction of 15.08 points. Task-level variation, episode pairing, and seed variation still require run-linked records.

The clean50–clean500 comparison separates nominal data scaling from the within-checkpoint steering comparison. The stronger clean500 policy still has a large absolute robustness gap; FIPAS is evaluated against this same frozen checkpoint rather than against the weaker clean50 policy.

5.2 Reported ManiSkill3 Differences Under Isolated and Composed Shifts

Our RoboTwin evaluation uses an aggregate randomized condition without isolated background, object, and sensor contrasts. We therefore use ManiSkill3 (Tao et al., 2025) to evaluate factor-controlled variation. Table 2 reports positive aggregate differences in all four shifted conditions and a small aggregate clean change. Task-level heterogeneity and uncertainty remain to be audited, as does whether the reported runs used a factor-blind union or a factor-label-aware selector.

Background-randomized success rate improves from 47.80% to 53.63%, a gain of 5.83 points. Object-randomized success rate increases from 43.40% to 50.94%, and sensor-randomized success rate increases from 44.60% to 51.73%, corresponding to gains of 7.54 and 7.13 points, respectively.

The reported all-randomized mean differs by 9.40 points (29.50% versus 38.90%), the largest displayed condition difference. Pooled matrices and single-factor controls remain necessary.

Table 2: Factor-controlled aggregate success rates on 10 ManiSkill3 tasks. The table compares the frozen policy and FIPAS under clean scenes, isolated background, object, or sensor randomization, and composed randomized conditions. The Change column reports the difference in percentage points. In the historical preset, the clean change is –0.21 points and the all-randomized summary changes from 29.50% to 38.90% (+9.40 points). These are aggregate summaries; task-level variation and the repeated evaluation denominators are not represented in this table.

Condition	Frozen policy	FIPAS	Change
Clean	56.40	56.19	-0.21
Background randomized	47.80	53.63	$+5.83$
Object randomized	43.40	50.94	$+7.54$
Sensor randomized	44.60	51.73	$+7.13$
All randomized	29.50	38.90	$+9.40$

The mean clean change is -0.21 points. This point estimate alone does not establish clean non-inferiority; that conclusion requires a predeclared margin and a paired task-level confidence interval. The largest displayed shifted difference occurs where the frozen policy is weakest.

These condition means do not establish invariance to unseen factors or identify independent factor contributions: the composed gain cannot be decomposed by subtracting the isolated-condition means.

5.3 Clean Preservation and Taskwise Evidence

The author-reported RoboTwin aggregate has a $+0.94$-point clean difference alongside the $+16.02$-point randomized difference. These means show a small average nominal change in the reported evaluation; they cannot establish that every task preserves clean performance or that the clean policy is non-inferior under a specified margin. Similarly, the aggregate shifted gain does not establish that every task benefits.

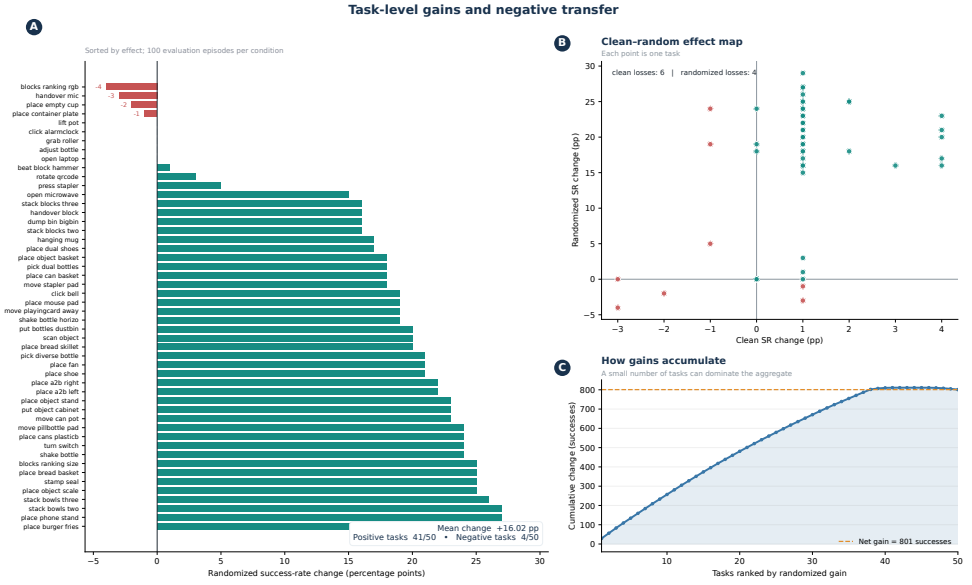

Figure 2: **Preset task-level gains and negative-transfer layout for the 50-task RoboTwin study.** Task-level gain and negative-transfer layout for the 50-task RoboTwin study. The displayed task ordering and per-task signs are an illustrative allocation constructed from aggregate endpoints. The historical preset shows 41 of 50 tasks with positive change and 4 of 50 with negative change; the remaining tasks are shown as unchanged. These counts and taskwise values are not reconstructed from episode-linked paired evaluation records.

Figure 2 is a historical illustrative allocation rather than a run-linked paired evaluation. Earlier subset diagnostics and Clean50 calibration checks are reported in Appendices A and B.

6 Discussion and Limitations

The aggregate results address whether a frozen policy can gain robustness under evaluated visual shifts while retaining clean-task ability on average. FIPAS differs from robustness training, which updates policy weights (Guo et al., 2026), and from steering based on successful versus failed trajectories (Miao et al., 2026) in both adaptation budget and the contrast used to estimate an intervention. These differences describe the setting; they do not establish superiority or empirical complementarity. A matched-checkpoint and matched-budget comparison, together with component ablations, is needed to attribute the reported gain to pairing, task protection, site selection, and gating.

The scope is bounded by the quality of the clean–shifted pairs and the calibrated factors. Divergent physical states or observation histories can contaminate the estimated environmental subspace, while a linear task basis can protect only the task information it captures. The soft residual gate may edit clean states; stronger or higher-rank corrections may disrupt useful action information. The rollout-state gate must use signals available at decision time, without simulator-only contact labels or future events. Run-linked task and episode records are needed to measure uncertainty and negative transfer; aggregate clean success alone does not prove non-inferiority. The present evidence comes from simulation, so real camera variation, latency, and contact dynamics remain untested. The claim is therefore limited to the evaluated shifts rather than arbitrary factor invariance or a safety guarantee.

7 Conclusion

Visual appearance can change a VLA policy's internal computation even when the physical task state and intended action are held fixed. FIPAS targets this source of variation with paired clean–shifted activations: it estimates an environment-correlated low-rank subspace, removes estimated task-predictive directions, and steers selected layer–action-query states according to a task-conditioned clean reference and observable rollout state. The policy weights remain frozen. This formulation separates where to edit a hidden state from when to adjust the edit during closed-loop execution.

The author-reported RoboTwin aggregates show why clean-data scaling and post-hoc correction answer different questions. Increasing clean demonstrations from 50 to 500 per task raises clean success from 67.96% to 85.40%, but leaves a 64.52-point clean–randomized gap for clean500. On that frozen policy, FIPAS raises randomized success from 20.88% to 36.90% ($+16.02$ points), while clean success changes from 85.40% to 86.34% ($+0.94$ points); the displayed gap narrows by 15.08 points. The separate ManiSkill3 summary shows 5.83–9.40-point gains across four shifted conditions and a -0.21-point clean change.

These results motivate selective representation correction for calibrated, task-preserving visual changes. Whether the proposed components account for the gain without task-specific clean failures remains to be tested with verified paired evaluation records, matched-budget baselines, and component and strength ablations. Physical-robot evaluation will determine how the approach behaves under real camera variation and closed-loop dynamics.

AI Use Statement

Generative AI tools assisted with methodological and experimental feedback, implementation, translation, literature search and synthesis, brainstorming, coding, and editing. The authors reviewed all AI-assisted work, verified source claims and code against original references and experimental outputs, and take responsibility for the final manuscript. AI was not used to generate synthetic datasets, formulate or prove mathematical claims, or clean and reformat datasets; proof-related and qualitative or thematic data-analysis tasks are not applicable to this work.

REFERENCES

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems*, volume 36, pages 66044–66063, 2023. doi: 10.52202/075280-2884. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d066d21c619d0a78c5b557fa3291a8f4-Paper-Conference.pdf.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H. Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://roboticsproceedings.org/rss19/p025.html.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. doi: 10.48550/arXiv.2506.18088. URL https://arxiv.org/abs/2506.18088.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38574–38583, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.pdf.

Alex Ferrando de las Morenas, Xavier Suau, Jordi Gonzàlez, and Pau Rodríguez. Dynamically scaled activation steering. *Transactions on Machine Learning Research*, 2026. URL https://arxiv.org/abs/2512.03661.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://www.roboticsproceedings.org/rss20/p090.html.

Jianing Guo, Zhenhong Wu, Chang Tu, Yiyao Ma, Xiangqi Kong, Zhiqian Liu, Jiaming Ji, Shuning Zhang, Yuanpei Chen, Kai Chen, Qi Dou, Yaodong Yang, Xianglong Liu, Huijie Zhao, Weifeng Lv, and Simin Li. RobustVLA: On robustness of vision-language-action model against multi-modal perturbations. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.00037.

Asher J. Hancock, Allen Z. Ren, and Anirudha Majumdar. Run-time observation interventions make vision-language-action models more visually robust. *arXiv preprint arXiv:2410.01971*, 2024. doi: 10.48550/arXiv.2410.01971. URL https://arxiv.org/abs/2410.01971.

Bear Häon, Kaylene Caswell Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 2743–2762. PMLR, 2025. URL https://proceedings.mlr.press/v305/haon25a.html.

Bingqi Huang, Bingchuan Wei, Xuan Wang, Yingkai Cai, and Zhaokui Wang. Cross-view action consistency for camera-robust vision-language-action policies. *arXiv preprint arXiv:2608.06965*, 2026. doi: 10.48550/arXiv.2608.06965. URL https://arxiv.org/abs/2608.06965.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e2dd53601de57c773343a7cdf09fae1c-Abstract-Conference.html.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Miranda Muqing Miao, Subin Kim, Brandon Yang, and Lyle Ungar. Contrastive conceptor activation steering (COAST): Unlocking vision-language-action models through hidden states. *arXiv preprint arXiv:2605.17144*, 2026. URL https://arxiv.org/abs/2605.17144.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 4996–5013. PMLR, 2025. URL https://proceedings.mlr.press/v270/nakamoto25a.html.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://robotic-transformer-x.github.io/.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Demonstrating GPU parallelized robot simulation and rendering for generalizable embodied AI with ManiSkill3. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.021. URL https://www.roboticsproceedings.org/rss21/p021.html.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint*

arXiv:2308.10248, 2023. doi: 10.48550/arXiv.2308.10248. URL https://arxiv.org/abs/2308.10248.

Yuhan Xie, Yuping Yan, Yunqi Zhao, Handing Wang, and Yaochu Jin. STRONG-VLA: Decoupled robustness learning for vision-language-action models under multimodal perturbations. *arXiv preprint arXiv:2604.10055*, 2026. URL https://arxiv.org/abs/2604.10055.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Working Diagnostic Protocols

Earlier small-subset experiments help diagnose two design hazards without establishing component necessity in the current 50-task result. Under a three-task diagnostic protocol, a centered-only variant reaches 11.67% randomized success compared with 1.67% for uncentered projection. Adding an *action-query-index* gate to the centered variant reduces randomized success to 1.67% in that same protocol, with clean success reported as 80.00% for both centered variants. An action-query index is a position within the predicted chunk, not a physical manipulation phase, so this heuristic should not stand in for a deployable rollout-state gate.

In a separate 12-task strict diagnostic, global-centered projection changes randomized success from 8.33% to 9.17% while clean success changes from 77.08% to 75.83%. Table 3 records the protocols separately. Their checkpoints, task subsets, and intervention definitions cannot be pooled with the primary aggregates or presented as an ablation of the final FIPAS configuration.

Table 3: Working design diagnostics (success rate, %). The table records exploratory comparisons from a 12-task strict protocol and a separate 3-task ablation protocol. Within each block, the historical preset compares clean and randomized success for the listed variants, including frozen, globally centered, uncentered, centered-only, and centered with an action-query-index gate. Because the blocks use different task subsets and protocols, values must be interpreted within their block and not compared across blocks; they are not final same-protocol ablations.

Protocol	Variant	Clean	Random
12-task strict	Frozen policy	77.08	8.33
12-task strict	Global-centered	75.83	9.17
3-task ablation	Uncentered	81.67	1.67
3-task ablation	Centered only	80.00	11.67
3-task ablation	Centered + AQ-index gate	80.00	1.67

The proposed paired-state, residual, and action-change diagnostic layout is shown in Figure 3. Its unverified panels are included as a reporting template and do not establish the mechanism of the Clean500 results.

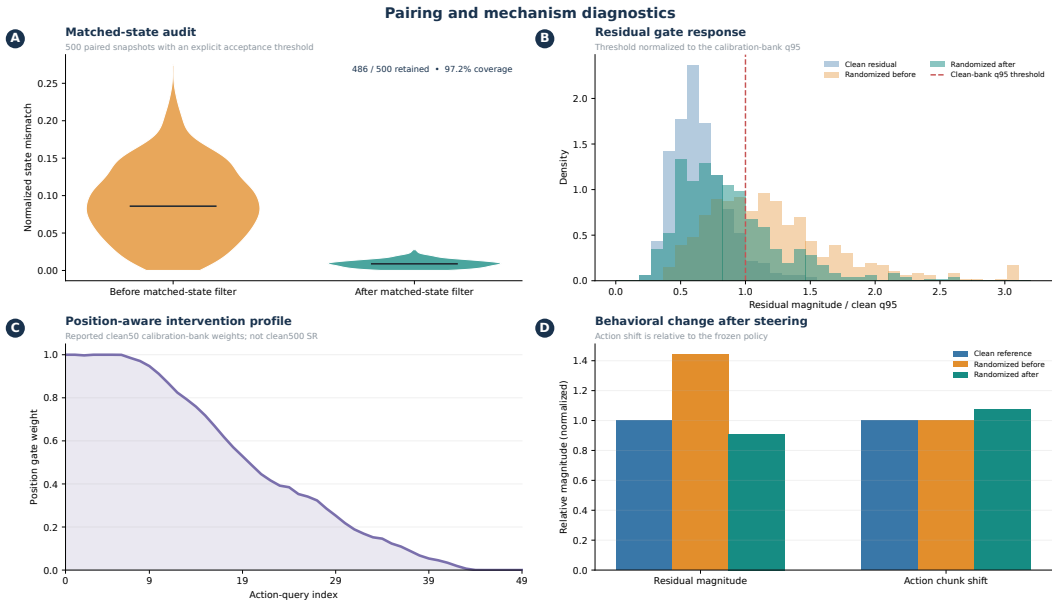

Figure 3: **Proposed mechanism diagnostics.** The panels organize matched-state retention, activation residuals, intervention position, and action change. The 486/500 retention value is a historical preset; panels A, B, and D lack run-linked source records, while panel C summarizes a separate Clean50 bank. This layout does not establish the mechanism of the Clean500 result.

B CLEAN50 CALIBRATION CHECKS

A separate clean50 action-head-input bank contains 294 activation samples from 10 tasks. Its aggregate clean/randomized nuisance basis has rank four; its stored isolated background, object, and sensor bases have rank zero. Figure 4 shows descriptive action-query weights and probe decodability for this bank, and Table 4 records its numerical checks. The measured task/protected-basis overlap is approximately 4.91×10^{-15}, which verifies the linear orthogonalization numerically but says nothing by itself about closed-loop preservation. The 50 action-query positions are token positions, not physical rollout stages. These bank diagnostics do not establish isolated-factor steering or the mechanism of the current clean500 policy's success gains.

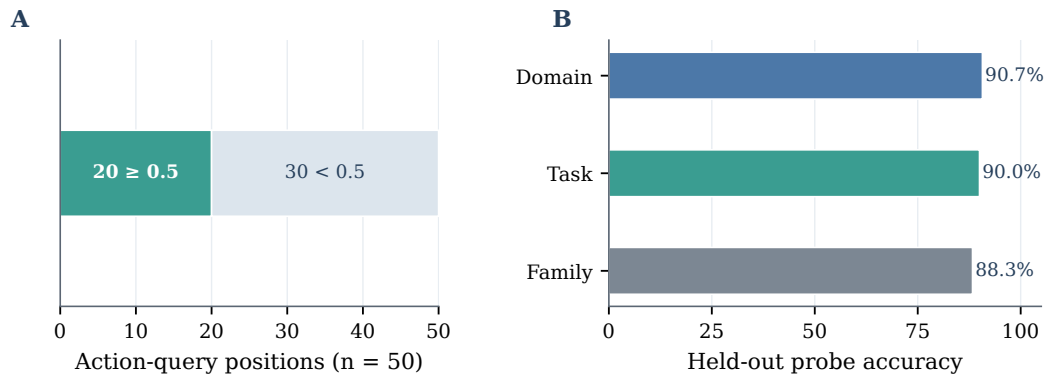

Figure 4: **Clean50 calibration bank diagnostics.** The plots summarize descriptive action-query weights and label decodability for an action-head-input activation bank containing 294 samples from 10 tasks. Domain, task, and family probes use different label spaces and split protocols. The plotted quantities describe this calibration bank; they are neither closed-loop FIPAS outcomes nor physical rollout-phase measurements.

Table 4: **Clean50 implementation checks.** The historical bank and identity-smoke record contains 294 activations from 10 tasks, an aggregate nuisance rank of 4, zero stored isolated background/object/sensor ranks, and 20 of 50 action-query positions with weights at least 0.5. The task/protected-basis overlap is recorded as 4.91×10^{-15}, and the $\beta = 0$ identity smoke passes on 2 tasks × 2 episode pairs. These checks document numerical behavior of the stored bank and smoke test; they do not establish policy benefit.

Check	Recorded value	Scope
Activations / tasks	294 / 10	One action-head-input hook
Aggregate nuisance rank	4	Clean/randomized basis
Background / object / sensor ranks	0 / 0 / 0	No isolated basis
Position weights ≥ 0.5	20 / 50	Action-query positions
Task/protected basis overlap	4.91×10^{-15}	Numerical orthogonality
$\beta = 0$ identity smoke	Pass	2 tasks $\times$ 2 episode pairs

C Evaluation Protocol Details

For a shifted condition c, we define

$$\mathrm{Gap}_c = \mathrm{SR}_{\mathrm{clean}} - \mathrm{SR}_c, \tag{9}$$

$$\mathrm{GapReduction}_c = \mathrm{Gap}_c^{\mathrm{frozen}} - \mathrm{Gap}_c^{\mathrm{FIPAS}}. \tag{10}$$

For RoboTwin, each reported per-task cell contains 100 trials in the author-corrected count table; 50 tasks give a denominator of 5,000 per condition. The same checkpoint is required for FIPAS-on and FIPAS-off comparisons. Run-linked episode IDs, seeds, checkpoint hashes, paired snapshot IDs, state and observation-history matching errors, and rejected pair counts are needed to audit implementation and uncertainty. The ManiSkill3 source does not establish task-level denominators or whether the factor-blind union rule was used in its reported runs.

The intended confirmation plan measures matched-budget contrasts against ungated steering, no task-basis protection, no task-specific clean reference, random subspace, and strength/rank sweeps. The existing 3-task and 12-task diagnostics in Appendix A use other checkpoints or definitions and cannot substitute for these comparisons.

D CONFIRMATORY ABLATION AND HYPERPARAMETER PROTOCOL

The newly supplied ablation and sensitivity archive describes a useful candidate experiment matrix but does not supply run-linked results for the clean500 policy. Its 49 summary rows are labeled `SYNTHETIC_PRESET_NOT_MEASURED`; the corresponding task-condition ledger has 4,900 rows whose run identifier is `NO_RUN_SYNTHETIC_PRESET` and whose checkpoint field is empty. Some rows rescale older diagnostic trends to the clean500 aggregate, and others explicitly have no corresponding numerical sweep in the source reports. These rows are therefore excluded from empirical figures, tables, task selection, and uncertainty estimates. Historical diagnostic values remain separate in Appendix A.

Same-protocol comparisons. Use the evaluated clean500 checkpoint and freeze its hash across variants. For every task and condition, match initial state, visual perturbation seed, episode count, success definition, calibration budget, and validation rule. The core controls are: no edit ($\beta = 0$), the complete intervention, unpaired or deliberately mismatched contrasts, no task-basis protection, global versus selected intervention sites, no residual gate, and no rollout-state gate. Direction controls include a random subspace and an opposite sign; success–failure steering or a COAST implementation should appear as a matched-budget comparison only if it is actually reproduced on the same policy and evaluation instances. A factor-label-aware bank selector, if tested, is an oracle control and must be distinguished from a factor-blind union. Every row reports clean and shifted success, not just the best shifted result. Rank-matched and edit-norm-matched controls are needed when two variants edit different numbers of dimensions or sites.

Candidate configuration search. Table 5 lists values proposed by the supplied planning configuration; it is a search specification rather than a list of measured settings. Search on disjoint calibration and validation trajectories, lock the chosen checkpoint, bank, site, strength, and gates before evaluating the held-out test episodes, and retain the full grid including unsuccessful settings. One-dimensional sweeps around a selected candidate reduce the search cost; a small joint strength–rank grid checks their interaction. The supplied configuration proposes 10 independent clean calibration episodes per task, five disjoint clean validation episodes per task, and three calibration repeats, but provides no split manifest or completed repeat logs. These quantities must be verified before being reported as actual execution details.

Table 5: validation grid, clean500 hyperparameters. The table lists candidate values for correction strength, nuisance rank, task-center shrinkage, clean-residual quantile, and related gate parameters. All entries are planned candidates rather than selected or measured hyperparameters. The final configuration is to be chosen using a predeclared selection objective and recorded run identifiers before held-out evaluation.

Parameter	Candidate values
Correction strength β (runtime alpha)	$0.01, 0.03, 0.05, 0.075, 0.10$
Nuisance rank	$1, 2, 4, 8, 16, 32$
Task-center shrinkage λ	$2, 4, 8, 16, 32$
Clean residual quantile	$0.90, 0.95, 0.975, 0.99$
Residual gate scale / slope	$0.5, 0.75, 1, 1.25, 1.5$ / $3, 6, 12, 18, 24$
Selected query fraction	$0.10, 0.20, 0.30, 0.50, 1.00$

The accompanying proposed configuration names a QwenOFT intervention immediately before its DiT action head, but neither the main-run checkpoint hash nor a run configuration is present in this archive. It also explicitly states that the referenced runtime has residual and domain gates but no separate rollout-phase gate. Thus a phase-gate ablation cannot be attributed to the current clean500 aggregate until a deployable gate is implemented, logged at each decision, and evaluated. The action-query index itself must not be used as a proxy for physical execution phase. Report the selected sites, realized union rank, threshold, intervention norm, fraction of activated steps, and added per-action latency alongside the final configuration.

Component-study planning layout. Table 6 preserves the seven-row structure from the supplied planning sheet. Its values are constructed scenarios, not evaluations of the stated variants. The added

Table 6: Component ablation across clean and randomized domains. Success rates are equally weighted over 50 tasks with 100 episodes per task and domain. Changes are percentage points relative to A0.

Configuration	Clean SR	Random SR	Δ Clean	Δ Random	Suppress.	Task prot.	Position	Residual	Phase	Full config.
A0 Frozen	85.40	20.88	+0.00	+0.00	–	–	–	–	–	
A1 Suppression	82.18	28.16	-3.22	+7.28	✓	–	–	–	–	–
A2 Task preserving	83.96	30.80	-1.44	+9.92	✓	✓	–	–	–	–
A3 Position gate	84.52	32.46	-0.88	+11.58	✓	✓	✓	–	–	–
A4 Residual gate	85.72	34.30	+0.32	+13.42	✓	✓	✓	✓	–	–
A5 Phase gate	86.08	33.34	+0.68	+12.46	✓	✓	✓	✓	✓	–
A6 Full FIPAS	86.34	36.90	+0.94	+16.02	✓	✓	✓	✓	✓	✓

Full configuration column distinguishes A6 from the A5 phase-gate variant in the layout; this does not validate the component effect, and the source runtime does not contain a separate rollout-phase gate.

Sensitivity search plan. The candidate grid is a planning specification, not an empirical sweep. Table 5 lists the proposed values and the proposed calibration/validation budget. The source archive has no split manifest or completed sweep logs, so this appendix reports no sensitivity outcomes and selects no test-set optimum.

E Task-Level Selection and Matched Rollout Cases

All 50 tasks should first be shown in a complete paired taskwise analysis, using original evaluation records rather than a task allocation reconstructed from aggregate totals. For each task, report frozen and edited clean and randomized success counts, denominators, $\Delta\mathrm{SR}$ in percentage points, checkpoint, seed, and episode-pair identifier. Under 100 trials per cell, the thresholds below are descriptive screening rules, not individual task significance tests. A task-cluster interval should accompany the mean improvement after real task pairs are available. Clean preservation requires a separately declared non-inferiority margin and paired interval. Figure 2 shows the supplied taskwise layout, whose per-task signs and ordering must be replaced from original runs.

For qualitative inspection, predefine three categories from the verified randomized task difference: improvement ($\Delta\mathrm{SR} \geq 5$ points), near zero ($|\Delta\mathrm{SR}| \leq 2$ points), and degradation ($\Delta\mathrm{SR} \leq -5$ points). Separately flag any clean decline of at least five points, even if the randomized result improves. Within the improvement category, choose the largest and a median gain; within the near-zero category, choose the task closest to zero; within degradation, choose the largest randomized loss and the largest clean loss if present. Break ties by task identifier and report empty categories rather than manufacturing examples. This selection is for explanation only; it must not replace the complete task distribution or determine the reported mean.

For each selected task, pair frozen and FIPAS rollouts by the same initial physical state and randomized rendering seed. Show the instruction, clean and shifted frames, normalized rollout progress, clean-reference residual, residual and rollout-state gate values, intervention norm, action difference, and outcome. A failure vignette should distinguish a pair mismatch, an unremoved nuisance direction, and an over-strong edit when the logs allow it. The supplied task-level ablation archive cannot identify these cases: its taskwise rows carry synthetic run identifiers and no evaluated checkpoint or episode IDs.

F Draft Replacement Checklist

Before empirical interpretation of the draft component layout, replace every constructed cell in Table 6 with run-linked clean500 outcomes, including success counts, denominators, task pairs, seeds, checkpoint and bank hashes, and exact intervention configuration. Reconcile the A5 phase-gate row with deployed code; if that gate is absent, remove the row and revise the Method, overview figure, abstract, and claims. The candidate grid in Table 5 is a search plan only. Confirm the calibration/validation split and matching-error report; select case tasks only after the full task distribution

is verified. Finally, update the Results, Discussion, Conclusion, and captions to state exactly the comparisons supported by run-linked records.