SPARSE TRANSCODERS FOR MECHANISTIC ANALYSIS OF VLA ACTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

We adapt sparse time-conditioned transcoders (STCs) to the action expert of the flow-matching vision-language-action policy $\pi_{0.5}$, giving a sparse intervention interface for systematically studying continuous action prediction. We train one STC for each of the 18 action-expert MLP blocks; the resulting sparse codes activate only 5.02% of the dictionary on average and support feature discovery, circuit tracing, open-loop ablation, and steering. In a discovery-selected black-bowl / bottom-drawer case study, a rare feature firing on only 0.166% of probed observations yields a traced 16-node upstream circuit whose ablation produces about a $2\times$ larger action-chunk change than ablating the target alone, and about a $2\times$ larger effect than matched same-size random circuits in the evaluated circuit setting. Target ablation is zero on negative controls where the feature is inactive, while behavior-grounded steering increases the Cartesian-speed score on 100% of evaluated examples for target-feature steering and 85% for circuit steering; an agreement-filtered contrastive direction changes the gripper coordinate by about 1.6 at $\alpha = 2$, compared with about 0.05 for a matched random direction. Separately, we run a visual-counterfactual audit using non-time-conditioned TopK transcoders on the PaliGemma language-prefix MLPs. There, 83.72% of a capped set of visually selective features reproduce their selectivity on held-out frames, but small sparse patches have limited and direction-dependent action effects, and attribution-based action restoration does not improve closed-loop success over occlusion-only or random-restoration controls. Together, the two studies show that sparse transcoders can support feature and circuit interventions in VLA policies while separating feature selectivity, open-loop action effects, steering, closed-loop behavior, and evidence that a circuit carries its attributed visual content.

1 INTRODUCTION

Vision-language-action (VLA) policies map visual observations, language instructions, and robot state to continuous actions. For such policies, interpretability claims should distinguish several questions that are often conflated: whether a feature responds consistently to a condition, whether changing it alters the predicted action, whether it can steer the action in a desired direction, whether the intervention improves closed-loop behavior, and whether the feature or circuit carries the semantic content attributed to it.

Sparse representations offer a route to this kind of mechanistic analysis. Sparse autoencoders and transcoders expose features inside neural networks, and transcoder-based circuit tracing can turn sparse features into directed hypotheses about computation through MLP layers. Flow-matching VLA policies introduce an additional challenge: action-expert MLPs are evaluated across denoising timesteps and output continuous action chunks rather than language tokens. A sparse replacement model must therefore preserve action-relevant computation while remaining conditioned on the flow timestep.

We adapt sparse time-conditioned transcoder circuit tracing to a LIBERO-finetuned $\pi_{0.5}$ VLA policy. To the best of our knowledge, this is the first application of timestep-conditioned transcoder circuit tracing to the action expert of a flow-matching VLA policy. We train one sparse time-conditioned transcoder (STC) per target action-expert MLP, use probe-mode activations for feature

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.

discovery, use replacement mode for circuit tracing and ablation, and use feature-level interventions for open-loop steering.

Empirically, the STCs reconstruct held-out action-expert MLP outputs with similar train, validation, and test losses. They reveal rare LIBERO features, including a black-bowl bottom-drawer feature whose traced 16-node circuit has a larger open-loop action effect than matched random circuits. Behavior-grounded steering further moves predicted action chunks along speed and gripper axes. To complement the action-expert analysis, we separately use non-time-conditioned TopK transcoders on the PaliGemma language-prefix MLPs to audit visual-feature selectivity, action transfer, and closed-loop recovery under controlled counterfactual perturbations. This audit asks whether features that respond selectively to a visual change also mediate its effect on action and behavior. It shows why load-bearing action effects and content-carrying explanations should be distinguished: selectivity can replicate on held-out frames, and sparse interventions can affect predicted actions, without guaranteeing closed-loop recovery or establishing that the selected features carry the attributed visual content. Our contributions are:

1. **Action-expert STCs:** We adapt timestep-conditioned sparse transcoders to the 18 MLP blocks of $\pi_{0.5}$'s flow-matching action expert and evaluate their reconstruction and sparsity.
2. **Feature and circuit analysis:** We use these STCs for feature discovery, candidate circuit tracing, open-loop ablation, and behavior-grounded steering, with matched controls for the reported case study..
3. **Separate visual-counterfactual audit:** Using non-time-conditioned TopK transcoders on the PaliGemma language-prefix MLPs, we test held-out feature selectivity, action-gap transfer, and closed-loop restoration under controlled visual perturbations.
4. **Evidence distinctions:** Across the two analyses, we distinguish feature selectivity, open-loop action effects, steering, closed-loop behavioral recovery, and content-carrying explanation.

1.1 Background and Related Work

Vision-language-action policies. VLA models connect visual-language representations to robot control. OpenVLA provides an open-source VLA model for manipulation (Kim et al., 2025), while the π_0 family uses a flow-based action model for continuous robot control (Black et al., 2025). The $\pi_{0.5}$ model extends this line toward open-world generalization (Physical Intelligence et al., 2025). We evaluate on LIBERO, a benchmark for language-conditioned manipulation and knowledge transfer (Liu et al., 2023).

Sparse representations and circuit tracing. Sparse autoencoders and transcoders decompose neural activations or MLP computations into sparse features (Gao et al., 2024; Dunefsky et al., 2024). Sparse feature circuits use such representations to construct and edit causal graphs in language models (Marks et al., 2025), and DifFRACT extends timestep-conditioned transcoder circuit tracing to diffusion models (Mazur et al., 2026). Our work adapts this idea to a flow-matching VLA action expert, where the sparse replacement model must be conditioned on denoising timestep and evaluated through action-chunk effects.

VLA mechanistic interpretability and steering. Recent work has begun to study mechanistic interpretability and steering in VLA models, including sparse VLA features, behavior-grounded SAE analysis, representation steering, and distribution-matching steering for robot policies (Swann et al., 2026; Jin et al., 2026; Häon et al., 2025; Buurmeijer et al., 2026; Khayatan et al., 2026). We build on this direction but emphasize a more explicit separation between feature discovery, circuit construction, open-loop causal effects, steering, and controlled tests of feature content.

Causal interventions and counterfactual evidence. Causal tracing, model editing, and activation patching show that internal interventions can localize and alter model behavior (Meng et al., 2022; Zhang & Nanda, 2024; Heimersheim & Nanda, 2024). However, patching and ablation establish that a representation is load-bearing under a chosen intervention; they do not automatically show that the representation carries the semantic content assigned to it. Our counterfactual audit follows this distinction by intervening on the input-side visual cause itself: a frozen simulator state is

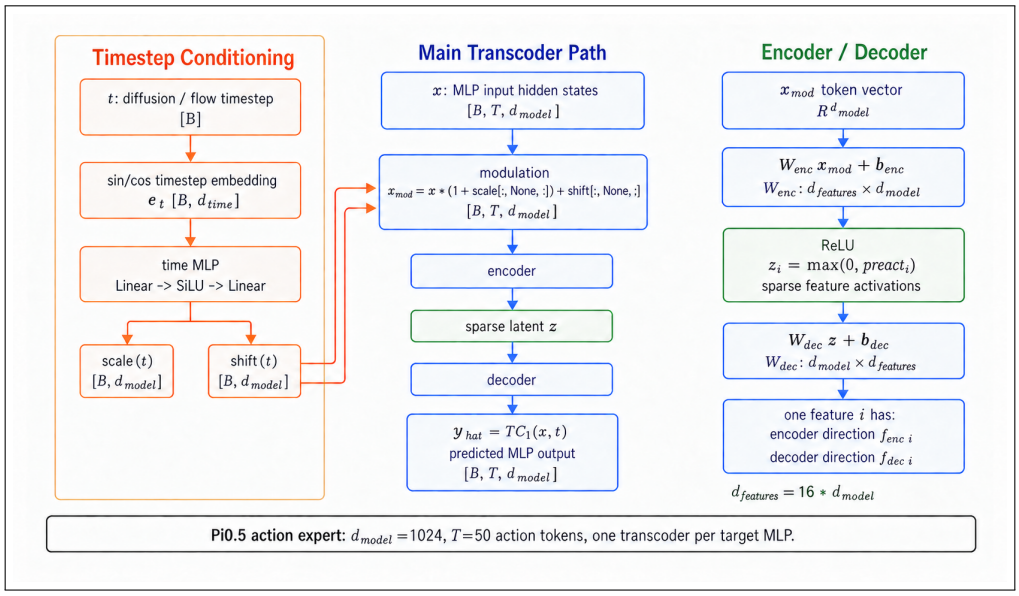


Figure 1: Sparse time-conditioned transcoder architecture. Each $\pi_{0.5}$ action-expert MLP receives a separate STC conditioned on the flow timestep τ. The sparse code $z_{\ell,r}$ is decoded back to the MLP output dimension, enabling probing, replacement-mode circuit tracing, and causal intervention.

rendered before and after a single-object perturbation, with matched non-referent and prompt-swap controls. This provides a complementary test of content-carrying explanations beyond action-level patch effects.

The rest of the paper first presents STC-based analysis of the $\pi_{0.5}$ action expert, then turns to controlled counterfactual perturbations of visual evidence, and closes with limitations and methodological lessons.

2 Sparse-Transcoder Circuit Analysis of the Pi0.5 Action Expert

2.1 Time-Conditioned Transcoder Architecture and Training

We train one sparse time-conditioned transcoder (STC) for each target MLP in the $\pi_{0.5}$ action expert. The policy predicts a chunk of $T = 50$ action tokens, and each target MLP processes hidden states $x_{\ell,r} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$, with $d_{\text{model}} = 1024$. The original $\pi_{0.5}$ policy remains frozen, and each STC is trained to reconstruct the corresponding frozen MLP output $y_{\ell,r}$.

Because $\pi_{0.5}$ is a flow-matching policy, the same MLP is evaluated at different flow timesteps. Each STC therefore conditions on the scalar timestep τ. A sinusoidal timestep embedding is passed through a small MLP to produce scale and shift parameters, which modulate the MLP hidden state before sparse encoding:

$$\tilde{x}_{\ell,r} = x_{\ell,r} \odot (1 + \gamma_\tau) + \beta_\tau, \qquad z_{\ell,r} = \text{ReLU}(W_{\text{enc}}\tilde{x}_{\ell,r} + b_{\text{enc}}),$$

with $d_{\text{features}} = 16d_{\text{model}}$. The decoder reconstructs the MLP output,

$$\hat{y}_{\ell,r} = W_{\text{dec}} z_{\ell,r} + b_{\text{dec}}.$$

Thus, each action-expert MLP has a time-conditioned sparse transcoder that can be used either as a probe or as a replacement for the MLP output.

We train the STCs with a normalized reconstruction loss plus an L_1 sparsity penalty:

$$\mathcal{L}_\ell = \frac{\mathbb{E}_r\left[\|\hat{y}_{\ell,r} - y_{\ell,r}\|_2^2\right]}{d_{\text{model}} \sum_{j=1}^{d_{\text{model}}} \text{Var}_r\left[y_{\ell,r,j}\right] + \epsilon} + \lambda \mathbb{E}_r\left[\|z_{\ell,r}\|_1\right], \qquad \lambda = 10^{-4}.$$

Table 1: Top-ranked sparse-feature rows from probe-mode discovery over 13,835 LIBERO observations. Context is summarized from the top-20 activating examples.

Rank	Feature	Interesting	Freq.	Top-20 context
1	L12 : $\tau = 1.0$: F7584	25.731	0.00181	yellow-white mug in microwave, close microwave
2	L17 : $\tau = 0.3$: F7883	25.712	0.00173	book placed in back compartment of caddy
3	L11 : $\tau = 0.7$: F9970	25.689	0.00166	black bowl, bottom drawer of cabinet, drawer closing

The total objective averages $\mathcal{L}_\ell$ over instrumented action-expert MLP layers. The final train, validation, and test losses are close: 0.0425, 0.0461, and 0.0458, respectively, suggesting limited overfitting. The learned representations are also sparse: across 72,000 probed tokens, the STCs activate an average of 823 out of 16,384 latent features per token, corresponding to a 5.02% active fraction. Additional training and sparsity plots are provided in Appendix A.

We use the trained STCs in three downstream modes: probe-mode feature discovery, replacement-mode circuit tracing, and feature-level causal intervention.

2.2 Feature Discovery

We first use the trained STCs to discover sparse features in the $\pi_{0.5}$ action expert. In probe mode, the frozen policy computation is unchanged, while each transcoder records sparse activations $z_{\ell,r}$ for LIBERO observations. For each layer ℓ, flow timestep τ, and feature i, we rank candidate features using their strongest activating examples relative to their background activation statistics:

$$\text{interesting}(i) = \frac{\text{meanTopK}(i)}{\text{std}(i) + \epsilon}.$$

Here, $\text{meanTopK}(i)$ is the mean activation over the top activating observations for feature i, and $\text{std}(i)$ is the standard deviation of that feature's activation over the full probe set. We also report firing frequency,

$$\text{freq}(i) = \frac{1}{N} \sum_{r=1}^{N} \mathbf{1}\left[z_{i,r} > 0\right],$$

where N is the number of probed observations. This ranking is used to nominate interpretable target features; it does not by itself establish causality.

Table 1 shows representative top ranked features from the discovery dashboard. The highest interesting score are rare, high activation features whose top examples often concentrate on coherent LIBERO task contexts. For example, different top features correspond to microwave closing, book placement, basket placement, and black-bowl cabinet-drawer behavior. This qualitative clustering is useful for hypothesis generation, but it also shows why causal validation is needed: a top feature can be task, episode, or event specific rather than a general behavioral mechanism.

We picked L11 : $\tau = 0.7$: F9970 as case study feature. It is rare, with firing frequency 0.00166, but has strong top activations: its top-20 activating examples have mean activation 0.697 and maximum activation 0.901. Manual inspection of these examples showed a coherent task context: black-bowl and cabinet-drawer observations, especially bottom-drawer closing behavior. Nearby timestep variants of the same feature also appear among the top candidates, suggesting that the feature is not a single-timestep artifact.

We therefore use L11 : $\tau = 0.7$: F9970 as a feature-discovery-selected target for circuit tracing and causal validation. For this feature-level validation, positives are defined as the top-activating F9970 observations from discovery, while negatives include low-target observations where F9970 is inactive, semantically related hard negatives, and stricter same object/drawer/cabinet controls. This setup tests whether the discovered feature and its traced circuit are causally active on the examples that nominated the feature, and whether the target feature is specific to that visual-task context

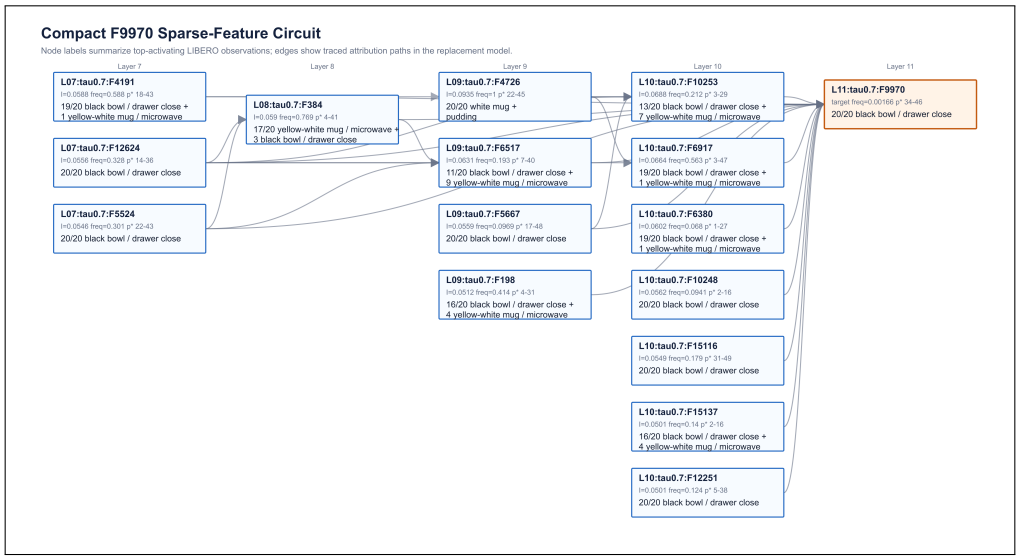


Figure 2: Compact circuit traced upstream of $\texttt{L11} : \tau = 0.7 : \texttt{F9970}$. Orange denotes the target feature; blue nodes are upstream parents; edges are attribution paths through the time-conditioned STC replacement computation.

2.3 Circuit Construction

We next construct a sparse-feature circuit upstream of $\texttt{L11} : \tau = 0.7 : \texttt{F9970}$. Circuit tracing uses the time-conditioned STCs in replacement mode, so earlier sparse features become inputs to later reconstructed MLP outputs. We initialize tracing from the top-20 activating $\texttt{F9970}$ examples at $\tau = 0.7$, score candidate parent edges by mean absolute activation-gradient attribution, and retain a compact graph by cumulative node and edge influence. Full tracing and pruning details are provided in Appendix B.

The resulting circuit contains 16 nodes and 26 directed edges, as shown in Figure 2. It includes the target feature and 15 upstream parent features across layers 7–10, with strongest retained edge mass 0.0226. We treat this compact graph as the main circuit hypothesis for causal validation. Qualitatively, the parent nodes are related to the same visual-task context as $\texttt{F9970}$, but are broader than the target feature, consistent with reusable upstream visual, state, or action-context components.

2.4 Open-Loop Causal Validation

We validate the target feature and its traced circuit using open-loop action-chunk ablations. For each selected observation, we hold the $\pi_{0.5}$ evaluation setup fixed and vary only the sparse-feature intervention. Our main metric is the mean ℓ_2 change in the predicted action chunk before versus after intervention.

On the top-20 activating examples for $\texttt{L11} : \tau = 0.7 : \texttt{F9970}$, target ablation changes the predicted action chunk, while ablating the upstream parents or full compact circuit produces larger effects (Table 2). Target ablation gives mean action ℓ_2 change 0.0254, parent ablation gives 0.0473, and circuit ablation gives 0.0482. Thus, the traced upstream circuit has roughly twice the open-loop effect of ablating the target feature alone.

The negative controls separate target specificity from parent breadth. Across low-target negatives, broad semantic negatives, and stricter same-object/drawer/cabinet controls, target ablation has zero effect: the target latent is already inactive, so ablating it does not change the predicted action chunk. This supports specificity of $\texttt{F9970}$ itself. Parent and circuit ablations can still affect negatives, with mean action ℓ_2 changes around 0.024–0.027, indicating that upstream parents are broader reusable components rather than a narrowly target-specific mechanism.

Table 2: Open-loop ablation and matched-control results for `L11 : τ = 0.7 : F9970`. Values are mean action ℓ_2 changes.

Condition	Mean action ℓ_2	Note
Target ablation, positives	0.0254	top-20 positives
Parent ablation, positives	0.0473	1.86× target
Circuit ablation, positives	0.0482	1.90× target
Target ablation, negatives	0.0000	all negative sets
Parent/circuit ablation, negatives	0.024–0.027	broader parents
Target ablation, matched run	0.0261	3.07× random-feature mean
Random single-feature controls	0.0085	range 0.0000–0.0256
Traced circuit, matched run	0.0527	positive examples
Random same-size circuits	0.0261	range 0.0255–0.0270

Table 3: Behavior-grounded steering results for speed-correlated feature `L15 : τ = 0.3 : F4340`. Positive-direction rate is the fraction of open-loop action chunks with $\Delta v > 0$, not a closed-loop success rate.

Intervention	Mean Δv	Positive-direction rate	Control-adjusted Δv
Target feature	$+0.000712$	100%	$+0.000904$
Parents	$+0.003500$	82.5%	–
Circuit	$+0.004171$	85.0%	$+0.006038$

Matched random controls show that the strongest evidence is circuit-level. Target-only ablation gives mean action ℓ_2 change 0.0261, compared with a random single-feature mean of 0.0085, though the strongest random single feature is close to the target effect. By contrast, traced-circuit ablation gives mean action ℓ_2 change 0.0527, while five random same-size circuit controls have mean 0.0261 and range 0.0255–0.0270. Thus, `F9970` is a rare, feature-discovery-selected target associated with the black-bowl / bottom-drawer context, and its compact traced circuit has a stronger open-loop causal effect on positive examples than matched random circuits.

2.5 Behavior-Grounded Steering

The causal-validation setup above is feature-discovery-driven: positives and negatives are defined relative to `F9970` activation and semantic context. For steering, we use behavior-grounded setups instead. Rather than forcing `F9970` into unrelated examples, we define steering targets using action speed, task phase, event-conditioned comparisons, and gripper-action agreement. This separates the circuit-validation claim from the steering claim: the former asks whether a discovered circuit is load-bearing on its discovery-selected examples, while the latter asks whether behavior-conditioned sparse directions can move predicted action chunks along chosen axes.

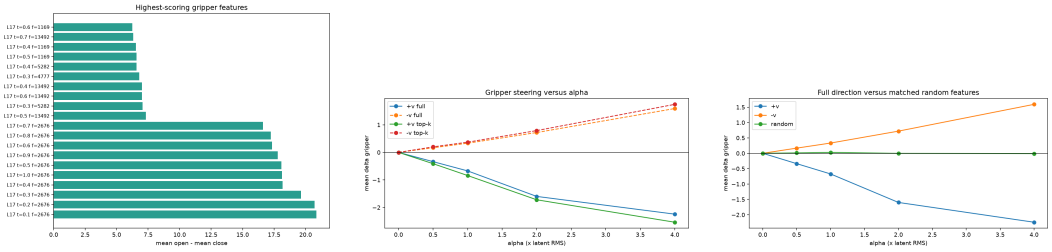
Speed steering. We focus on a speed-correlated feature, `L15 : τ = 0.3 : F4340`, on the LIBERO task "put the black bowl in the bottom drawer of the cabinet and close it." Positive examples come from a faster drawer-closing trajectory and negatives from slower trajectories in the same task, matched within late drawer-closing event bins. For an action chunk a, we define the open-loop speed score as

$$v(a) = \frac{1}{T}\sum_{t=1}^{T} \|a_{t,1:3}\|_2 , \qquad \Delta v = v(a_{\text{steered}}) - v(a_{\text{baseline}}).$$

We steer the target feature, its traced parents, or the traced circuit by setting selected sparse activations to the positive-example activation level, and compare against matched random controls using shared action noise. Table 3 shows that target-feature steering increases the speed score on all evaluated examples, while parent and circuit steering produce larger mean speed increases. Circuit steering gives mean $\Delta v = 0.004171$, positive-direction rate 85.0%, and remains positive after subtracting matched random-circuit controls.

Agreement-filtered gripper steering. We also test whether a contrastive direction in the trained time-conditioned STCs can steer the gripper coordinate of a frozen $\pi_{0.5}$ policy. LIBERO gripper actions concentrate on open and close modes. We keep only training frames where the unsteered policy's predicted gripper label agrees with the demonstration label, yielding 78 agreeing frames out of 168. At layer 17, we average STC latents over the first $H = 5$ action tokens and over denoising times $\mathcal{T} = \{1.0, 0.9, \ldots, 0.1\}$. Paired open and close frames define a contrastive direction v, with $+v$ pointing from close toward open. A sparse top-32 direction keeps the largest consistent coordinates, and a matched random control permutes feature axes.

At intervention time, the direction is added to layer-17 latents at every denoising time, while the original MLP output is left in place and the STC decoder supplies only the decoded change. As shown in Figure 3, feature 2676 dominates the open-minus-close gap at every denoising time. Adding $+v$ moves the predicted gripper action toward opening, while $-v$ moves it toward closing; the top-32 direction tracks the full direction, and the matched random control stays near zero. At $\alpha = 2$, the mean absolute gripper change is about 1.6 for the learned direction and about 0.05 for the matched random direction, while the next largest action-coordinate change is about 0.2 in z-translation and rotations remain near zero.



(a) Gripper-related sparse features (b) Steering strength (c) Random control

Figure 3: Agreement-filtered gripper steering at action-expert layer 17. The learned direction produces oppositely signed opening/closing changes and substantially larger gripper effects than the matched random control.

Together, the speed and gripper experiments support an open-loop directional-control claim: behavior-conditioned sparse features and directions can move $\pi_{0.5}$'s predicted action chunks along intended behavioral axes. These are not closed-loop rollout claims; future work should test whether such edits improve task success, contact timing, or trajectory quality during execution. Additional construction and intervention details are provided in Appendix C.

Having used time-conditioned STCs to discover, trace, and intervene on action-expert features, we now ask a complementary question: when a visual perturbation is known by construction, do selective features in the language prefix transfer its effect to action and behavior? We study this question with separately trained, non-time-conditioned TopK transcoders

3 Controlled Counterfactual Perturbation of Visual Evidence

This study instruments the 18 PaliGemma language-prefix MLPs. The transcoders here are TopK transcoders with 4,096 features and 64 active features per token; they are not denoising-time conditioned. We construct target-linked visual perturbations: Box occludes the target in both cameras, Screen targets the third-person view, and Paint replaces target pixels with gray. Miss, distractor, recolor, and absent controls distinguish target occlusion from broader visual changes; full contact sheets are provided in Appendix F. Feature sets are selected on discovery demonstrations and evaluated on disjoint test demonstrations, with additional configuration details in Appendix D.

For feature-set intervention, donor-recipient differences in decoded transcoder contributions are added to the real recipient MLP output, preserving the recipient reconstruction residual. We compare selectivity-selected features, attribution-ranked features, all-transcoder patches, full-MLP patches, and matched random controls. For recipient action a_s, target action a_t, and patched action a_p, the

Table 4: Held-out injection: mean percent action-gap closure [95% task-bootstrap CI].

Intervention	Spatial box	Spatial screen	Spatial paint	Object box
Attribution, 100	1.49	2.81	5.25	0.90
95% CI	[0.65,2.54]	[0.44,5.01]	[3.31,7.47]	[0.46,1.41]
Capped selective	0.16	0.45	−0.01	0.20
95% CI	[−0.11,0.50]	[0.00,1.17]	[−0.21,0.15]	[0.04,0.38]
All transcoder	56.85	55.10	59.02	47.70
All MLP	73.26	78.54	75.80	74.10

main open-loop metric is normalized action-gap closure,

$$C = 100 \left(1 - \frac{\mathrm{RMSE}(a_p, a_t)}{\mathrm{RMSE}(a_s, a_t)} \right).$$

Injection transfers occluded feature values into the clean run; restoration transfers clean values into the occluded run. This metric measures action-chunk closeness, not rollout success.

Selectivity replicates but does not imply mediation. Spatial discovery identifies 452 box-selective features, 731 screen-selective features, and 24 paint-selective features. For Spatial box, 108/129 capped features reproduce selectivity on held-out frames (83.72%), while the uncapped rate is 341/452 (75.44%). Object box shows the same qualitative pattern, with 75.19% capped and 64.11% uncapped replication. These rates validate the operational activation criterion, but they do not by themselves identify causal mediators.

Sparse action effects are measurable but limited. Table 4 reports held-out injection effects. Spatial box attribution exceeds matched random by 1.45 points [0.64, 2.47], with positive task means on all eight tasks, but remains below the ten-point practical margin. The capped selective set closes only 0.16% of the normalized action gap, and even all 452 selective features close only 0.46% [−0.25, 1.13]. In contrast, all-transcoder and full-MLP patches close much larger fractions of the gap, showing that the intervention site carries substantial action-relevant information.

Restoration is stronger than injection. Spatial box attribution restoration closes 12.52% [3.07, 22.29] of the occluded-to-clean action gap and exceeds random by 12.79 points [3.35, 22.49]. Screen and paint restoration close 7.29% and 18.73%, respectively, and Object box closes 7.12% [3.01, 12.41]. A blanket claim that sparse interventions have no causal effect would therefore be incorrect; the more precise claim is that the tested small sparse sets have limited and direction-dependent action effects. Budget curves, token-scope analyses, and ranking diagnostics are provided in Appendix D.

Action restoration does not guarantee behavioral recovery. Closed-loop evaluation uses 60 episodes per condition across five Spatial tasks. Box occlusion strongly reduces success, and full-MLP restoration nearly recovers the clean baseline. Attribution restoration, however, provides no reliable improvement over either covered or random-restoration baselines (Figure 4; exact counts are in Table 6 in Appendix E). Its normalized recovery fraction is -0.04 [-0.24, 0.06], compared with 1.02 [0.95, 1.16] for full-MLP restoration. Thus, small action-level restoration does not establish behavioral recovery.

Overall, the counterfactual study separates reproducible selectivity, finite action effects, and executed behavior. Selective features can replicate on held-out frames, and attribution-selected features can produce measurable action effects, but these facts do not by themselves imply closed-loop recovery or content-carrying explanation. Additional attribution-ranking, token-scope, restoration, and budget-curve diagnostics correspond to the compressed Section 3 results.

4 Conclusion

We presented two complementary sparse-transcoder studies of the LIBERO-finetuned $\pi_{0.5}$ policy. In the first study, we adapted sparse time-conditioned transcoders (STCs) to the flow-matching action expert. These STCs are conditioned on the denoising timestep and provide an intervention-ready

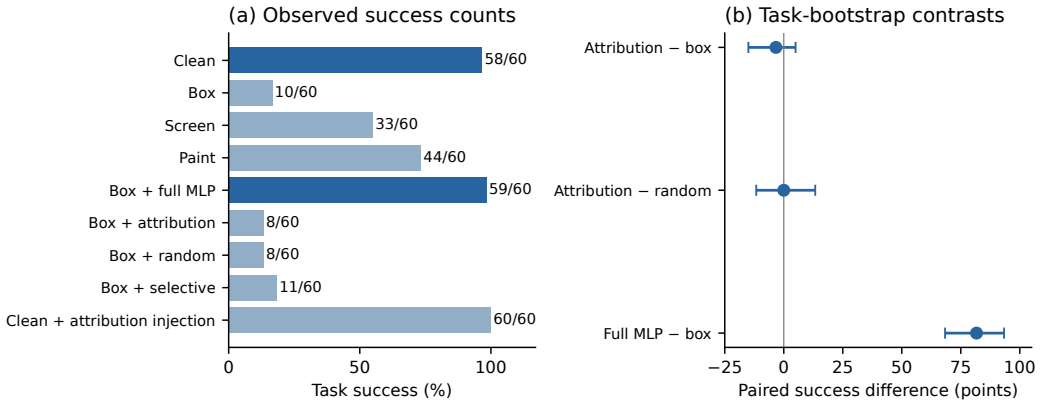


Figure 4: **Action restoration does not guarantee behavioral recovery.** Recorded successes and paired success-rate differences over 60 episodes per condition. Attribution restoration shows no reliable improvement over box or random restoration, while full-MLP restoration provides a strong positive control.

sparse interface for feature discovery, circuit tracing, ablation, and steering. In a discovery-selected case study, a rare feature associated with a black-bowl / bottom-drawer context yielded a traced 16-node upstream circuit whose ablation changed predicted action chunks more than matched same-size random circuits. Negative controls further showed that the target feature is inactive off-context, while broader parent and circuit features can still affect related negatives, supporting a target-specific feature embedded in a broader upstream support circuit. This case study demonstrates a general workflow that can be applied to other discovered action-expert features: nominate a feature in probe mode, trace an upstream circuit in replacement mode, and validate the target and circuit with controlled interventions. Behavior-grounded interventions further showed that STC features and directions can steer open-loop action predictions along a speed axis and an agreement-filtered contrastive gripper open/close direction.

In the second study, using non-time-conditioned TopK transcoders on the PaliGemma language-prefix MLPs, we ran a separate visual-counterfactual audit. This audit asks whether sparse features and traced sets carry controlled input-side visual content. Many visually selective features reproduced their selectivity on held-out frames, but small sparse patches produced limited and direction-dependent action effects. Attribution-based restoration changed open-loop actions but did not reliably improve closed-loop success over occlusion-only or random-restoration controls, while full-MLP restoration served as a strong positive control. Thus, feature selectivity and action-level causal effects do not, by themselves, establish behavioral recovery or show that a circuit carries its attributed visual content.

Taken together, these studies support sparse transcoders as practical tools for investigating VLA representations while underscoring the need to keep distinct claims separate: reconstruction quality, feature selectivity, open-loop intervention effects, steering, closed-loop behavior, and content-carrying explanation. Future work should apply the STC workflow to additional discovered action-expert features, test traced circuits on held-out episodes, evaluate steering interventions in closed-loop rollouts, and broaden counterfactual audits across tasks, visual perturbations, and policy architectures.

Reproducibility Statement

We report the evaluated policy, LIBERO suites, STC architecture, training split sizes, reconstruction losses, sparsity statistics, feature-discovery metrics, circuit-tracing parameters, pruning thresholds, ablation metrics, steering setup, and counterfactual perturbation protocol in Sections 2–3 and Appendices A–E. The appendix also includes the circuit-pruning procedure, gripper-steering construction, counterfactual feature-selection and patching details, budget diagnostics, and closed-loop behavioral counts. Code, checkpoints, generated figures, and run artifacts will be released as supplementary material where allowed by the submission policy.

AI ASSISTANCE STATEMENT

AI tools were used to help polish prose, organize drafted paragraphs, and convert author-written notes into LaTeX. The authors controlled the experimental design, analyses, numerical results, figures, citations, scientific claims, and final manuscript text.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2410.24164.

Hugo Buurmeijer, Carmen Amo Alonso, Aiden Swann, and Marco Pavone. Observing and controlling features in vision-language-action models, 2026. URL https://arxiv.org/abs/2603.05487.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.11944.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024. URL https://arxiv.org/abs/2406.04093.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024. URL https://arxiv.org/abs/2404.15255.

Bear Häon, Kaylene Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models, 2025. URL https://arxiv.org/abs/2509.00328.

Xinchen Jin, Aditya Chatterjee, Pranav Kumar, and Rohan Paleja. Event-grounded sparse autoencoders for vision-language-action policies, 2026. URL https://arxiv.org/abs/2605.17204.

Pegah Khayatan, Sara Meziane, Jayneel Parekh, and Matthieu Cord. DiMaS: Distribution matching for steering vision-language-action models, 2026. URL https://arxiv.org/abs/2607.14280.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, et al. 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*, pp. 2679–2713, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.03310.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2403.19647.

Artyom Mazur, Nina Konovalova, and Aibek Alanov. Diffract: Diffusion feature reconstruction and attribution for circuit tracing, 2026. URL https://arxiv.org/abs/2606.15796.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2202.05262.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Aiden Swann, Lachlain McGranahan, Hugo Buurmeijer, Monroe Kennedy III, and Mac Schwager. Sparse autoencoders reveal interpretable and steerable features in vla models. In *Conference on Robot Learning*, 2026. URL https://arxiv.org/abs/2603.19183.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

A STC Training and Sparsity Details

This appendix provides the training-loss and sparsity plots summarized in Section 2. The main text reports the corresponding train, validation, test, and active-feature statistics.

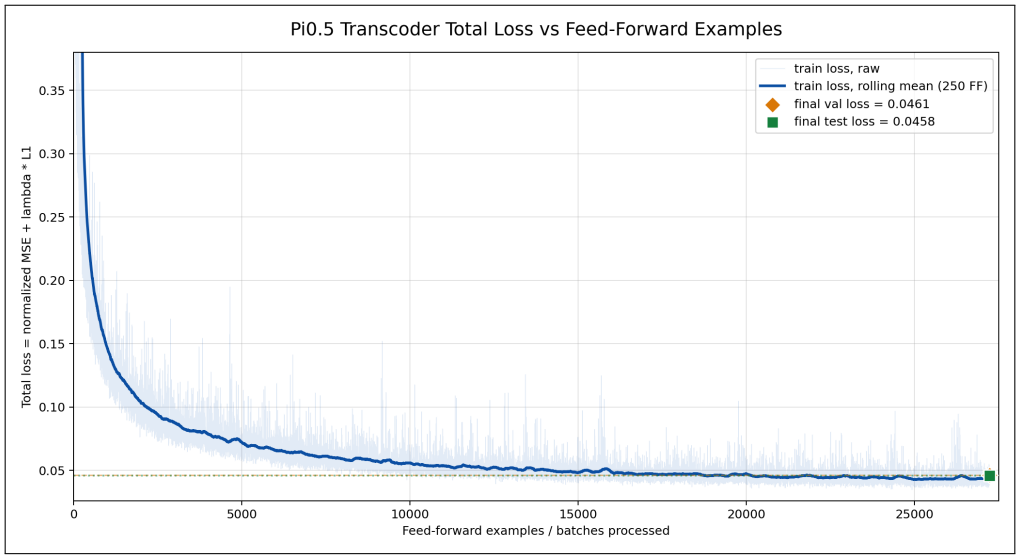


Figure 5: Training summary for sparse time-conditioned transcoders. We train on 217,857 LIBERO frames, with 26,655 validation frames and 28,953 test frames. The final train, validation, and test losses are 0.0425, 0.0461, and 0.0458, respectively, indicating that the STCs reconstruct frozen $\pi_{0.5}$ action-expert MLP outputs with similar held-out performance.

Sparsity varies with the flow timestep. Most layers exhibit fewer active features from early to intermediate denoising timesteps, although the magnitude and temporal pattern differ across layers. Together, these results show that the sparsity penalty produces sparse latent representations while maintaining accurate reconstruction of the original MLP outputs.

B Circuit-Tracing and Pruning Details

For the F9970 causal-validation circuit, tracing is initialized from the top-20 activating examples of F9970. For each expanded node, we run the time-conditioned STCs in replacement mode at the target timestep $\tau = 0.7$, compute local feature attributions from earlier layers using the product of sparse activation and gradient, and aggregate attribution over the selected examples. Candidate parent edges are scored by mean absolute attribution, and we keep candidate parents whose mean absolute attribution is at least 0.003. The trace uses all earlier source layers, maximum depth 3, and an expansion budget of 8 frontier nodes. This expansion budget controls how many nodes are expanded with expensive attribution computations; it is not the final graph size.

After expansion, we prune the graph in two stages. First, we compute indirect node influence by column-normalizing incoming attribution mass and propagating influence backward from the root target feature. We then keep the highest-influence nodes until they account for 60% of cumulative

non-target node influence. Second, we prune edges by cumulative edge influence, where an edge's pruning score is its normalized incoming attribution mass times the influence of its target node. We keep the strongest edges until they account for 90% of cumulative edge influence. This produces the compact circuit used for causal validation.

C AGREEMENT-FILTERED GRIPPER STEERING DETAILS

Gripper labels. LIBERO actions use coordinate $g = 6$ for the gripper. Demonstration values concentrate on two modes. Let c_- and c_+ be their centers, $m = \frac{1}{2}(c_- + c_+)$, and $\delta = 0.15(c_+ - c_-)$. Values in $(m - \delta,\, m + \delta)$ are left unlabeled. Finger opening is computed as the mean of the last two dimensions of the observation state. The mode with the larger mean finger opening is labeled as open. On `libero_spatial`, the high action mode corresponds to close, so

$$\text{label}(a) = \begin{cases} \text{open}, & a \le m - \delta, \\ \text{close}, & a \ge m + \delta, \\ \varnothing, & \text{otherwise.} \end{cases}$$

The fitted thresholds are $m - \delta = -0.8$ and $m + \delta = 0.3$. A frame is labeled only when the next $H = 5$ demonstration gripper commands share the same label as that frame. Episodes are split 70/30 into training and holdout sets. Train frames are capped at two per task, progress bin, and label; holdout frames are capped at eight per label.

Agreement filter. The demonstration label is not always the gripper action emitted by the policy. On each probed training frame, we run the unsteered policy and compute

$$\hat{g} = \frac{1}{H} \sum_{t=1}^{H} a_{t,6},$$

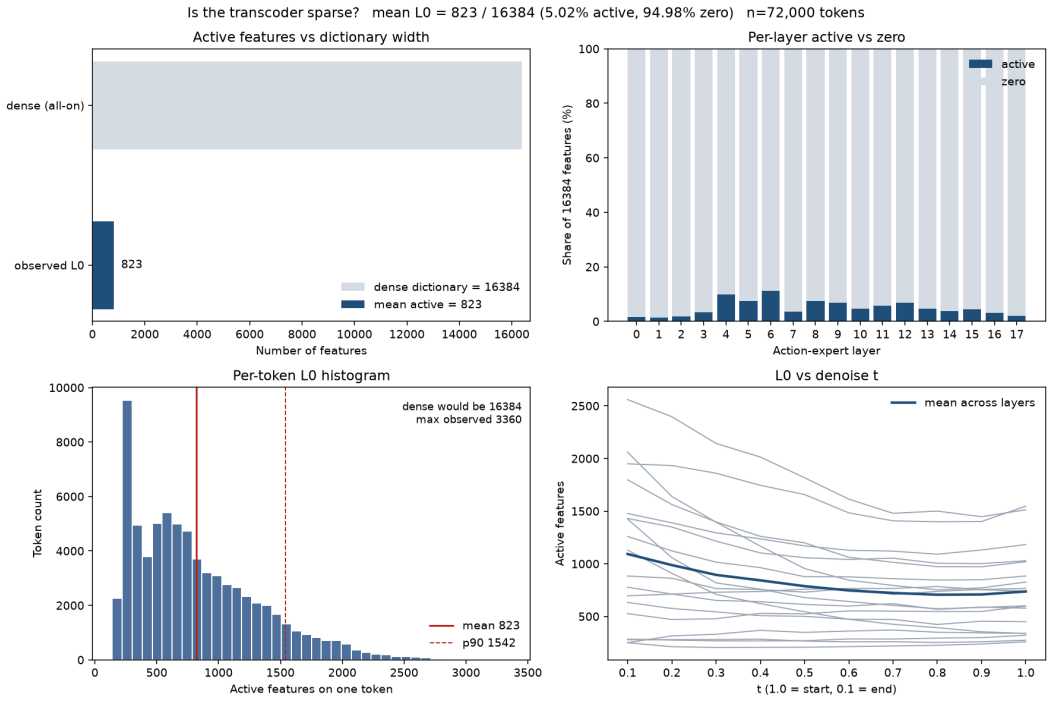


Figure 6: Sparsity of the learned STC representations across the $\pi_{0.5}$ action-expert layers. Across 72,000 probed tokens, an average of 823 out of 16,384 latent features are active per token, corresponding to 5.02% active features and 94.98% zero activations. The panels summarize the overall active dictionary size, layerwise sparsity, the per-token L_0 distribution, and variation in L_0 across flow timesteps.

where $a_{t,6}$ is the gripper coordinate of the predicted action chunk. The frame is kept only when $\mathrm{label}(\hat{g})$ equals the demonstration label. In this probe, agreement held for 78 of 168 training frames, with mismatches concentrated on the open side of the action. The contrastive direction is estimated only from agreeing frames, and both classes must be present.

Contrastive direction. Let $z_{\ell,\tau}(x) \in \mathbb{R}_+^{16384}$ be the STC latent at action-expert layer ℓ and flow time τ, averaged over the first H action tokens. Denoising uses ten times,

$$\mathcal{T} = \{1.0, 0.9, \ldots, 0.1\}.$$

For each agreeing frame, we pool layer $\ell = 17$ over denoising time:

$$\bar{z}(x) = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} z_{17,\tau}(x).$$

Open and close frames that share a task and progress bin are paired. With N such pairs, we define

$$v = \frac{1}{N} \sum_{i=1}^{N} \big(\bar{z}(x_i^+) - \bar{z}(x_i^-)\big),$$

so $+v$ points from close toward open. If no within-task pair exists, v is computed from equal-sized open and close subsets.

For inspection, coordinates are ranked by

$$s_j = |\mu_j^+ - \mu_j^-|\, c_j \, \max(f_j^+, f_j^-),$$

where μ_j^+ and μ_j^- are class-mean activations, f_j^+ and f_j^- are class firing fractions, and c_j is a consistency score measuring how often the feature's paired open-minus-close difference has the same sign as the mean paired difference. Zero paired differences do not count as agreement. The steering direction itself is v; the coordinate score is used only for inspection and sparse-coordinate selection.

The sparse control keeps the $K = 32$ largest coordinates of v among coordinates with $c_j \geq 0.7$, and rescales the resulting vector to have the same ℓ_2 norm as v. The random control permutes the feature axes of v, preserving the entries and sparsity pattern while breaking their alignment to learned feature coordinates.

Intervention. Let ρ be the root-mean-square of the layer-17 latents on the agreeing frames, over all $\tau \in \mathcal{T}$, and let $\mathrm{rms}(v)$ be the root-mean-square of v. A signed step α is applied as

$$\beta(\alpha) = \alpha \, \frac{\rho}{\mathrm{rms}(v)}, \qquad z' \;=\; z + \beta(\alpha)\, v.$$

The added vector has root-mean-square $|\alpha|\rho$, so $\alpha = 1$ matches the scale of the latent. The same v and $\beta(\alpha)$ are added at every $\tau \in \mathcal{T}$ in layer 17 only. The original MLP output y is left in place and the time-conditioned STC decoder supplies only the decoded change,

$$y' \;=\; y + \mathrm{Dec}(z') - \mathrm{Dec}(z).$$

We sweep $\alpha \in \{0, 0.5, 1, 2, 4\}$, with the sign flip $-\alpha$, on episode-held-out frames. The reported gripper effect is the mean change in coordinate 6 over the first H action steps. Specificity is measured as the mean absolute change of the other six action coordinates.

D Counterfactual Experiment Details

The controlled counterfactual study uses the LIBERO-finetuned $\pi_{0.5}$ policy on Spatial and Object suites. Spatial discovery uses demonstration IDs 0 and 1; test uses 2–4; extra training uses 5–16. Object uses discovery IDs 0 and 1, test IDs 2 and 3, and extra training IDs 4–11. Probe frames target phases 0.30, 0.55, and 0.80 of the approach before the first target lift exceeding 0.01 m, with a three-step pre-lift margin. A local search within six steps enforces visibility and edit-placement constraints, including a target mask of at least 80 pixels.

The realized Spatial sample contains 46 discovery and 70 test frames across eight tasks, despite ten configured tasks. Tasks 4 and 6 supply no valid probes. Object supplies 60 discovery and 60 test frames across ten tasks. Extra training uses 419 Spatial and 299 Object frames. Dictionary training includes discovery data; test demonstrations are excluded. The dictionary's random token holdout is not an independent episode holdout.

Training uses 396,551 Spatial and 386,859 Object tokens per layer, with holdouts of 20,871 and 20,361. The auxiliary TopK is 256, coefficient 0.03125, inactivity window 100, warmup 5%, and decay 20%; seeds are based on zero plus layer index. Dictionary dimensions are 4,096 features and TopK 64 for every layer. The primary attribution budget is 100; injection budgets are 10, 30, 100, 300, 1,000, and 3,000. Restoration budgets are 30, 100, 300, and 1,000. The primary random intervention uses eight draws; secondary and restoration controls use three. Three single-feature interventions are also evaluated.

Feature selection and patching. For each layer ℓ, a TopK transcoder with 4,096 features and 64 active features per token approximates the MLP output:

$$\widehat{y}_\ell(x) = D_\ell f_\ell(x) + b_\ell.$$

To patch feature set S from donor d into recipient r, we add

$$\Delta y_{\ell,t} = \sum_{j \in S_\ell} D_{\ell,j}\big(f^d_{\ell,j,t} - f^r_{\ell,j,t}\big)$$

to the real recipient MLP output. This preserves the recipient reconstruction residual. We separately patch all transcoder features, the residual error only, and full MLP outputs. Their measured effects are nonlinear and are not an additive causal decomposition.

Discovery-time selectivity sums activation over tokens. A feature must increase on at least 90% of frames, rise by at least one typical firing magnitude, and meet specificity 0.5 relative to recolor, absent, and miss responses. A target-specific variant also compares a distractor. Let $\bar{\delta}_j$ be the mean target-condition increase and q_j the largest mean absolute response to the quiet controls. Specificity is $1 - q_j/\bar{\delta}_j$ for $\bar{\delta}_j > 0$; the rise statistic divides $\bar{\delta}_j$ by a typical firing magnitude. Capped sets retain at most eight features per layer.

For attribution ranking, decoded feature differences are scored against gradients of a directional action-projection objective:

$$\alpha_{\ell,j} = \sum_t (f^o_{\ell,j,t} - f^b_{\ell,j,t}) \left\langle D_{\ell,j}, \frac{\partial m}{\partial y_{\ell,t}} \right\rangle.$$

Injection and restoration endpoint scores are averaged, then aggregated over discovery frames. The top 100 alive features are ranked in layers 0–13, selected by a discovery residual-swap criterion. Full-transcoder patches span all 18 layers; this support mismatch limits a pure feature-count interpretation of sparse-versus-full comparisons. Primary random controls match per-layer counts and decoded per-token edit norms, with repeated draws averaged within frame. Zero random edits remain zero. Budget-curve random controls use raw edits.

Metrics and statistics. For recipient action a_s, target action a_t, and patched action a_p, normalized action-chunk gap closure is

$$C = 100 \left(1 - \frac{\mathrm{RMSE}(a_p, a_t)}{\mathrm{RMSE}(a_s, a_t)}\right).$$

Negative values indicate increased distance. A separate projection metric measures movement along the start-to-target direction. Injection transfers occluded feature values into the clean run; restoration transfers clean values into the occluded run. Rollouts execute postprocessed actions. Causal tests require an action gap of at least 0.1, distinct from the 0.02 validity threshold. Retained Spatial box/screen/paint frame counts are 70/69/63; Object box/paint counts are 59/48. Confidence intervals use 10,000 whole-task bootstrap resamples of a frame-weighted mean. Paired one-sided sign-flip tests enumerate signs over task-mean differences. The stored protocol sets a ten-point practical margin for the primary attribution contrast; exploratory tests have no multiplicity correction.

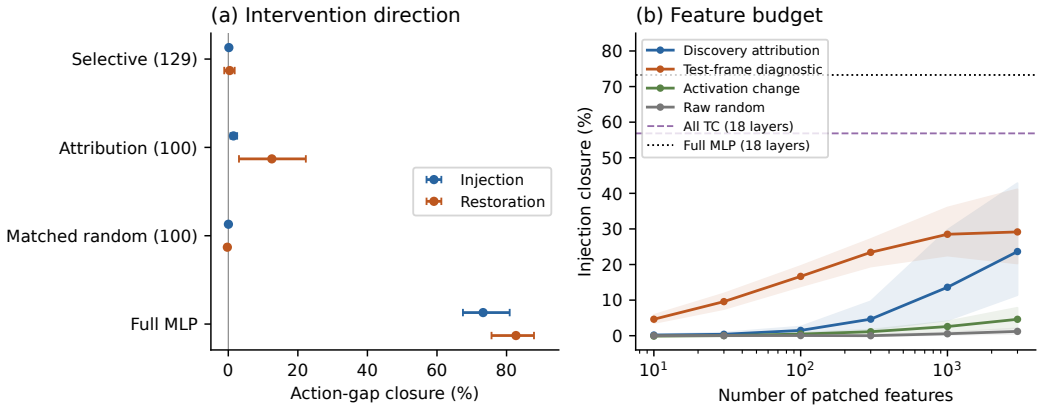


Figure 7: Action effects depend on direction and budget. Spatial box, 70 test frames across eight tasks. (a) Means and 95% whole-task bootstrap intervals for injection and restoration. (b) Injection versus feature budget, with shaded task-bootstrap intervals. The test-frame diagnostic uses test information. Dashed reference lines patch all transcoder features or full MLP outputs over 18 layers; ranked sets use layers 0–13, so these references are not layer-matched feature-count controls.

Table 5: Spatial box injection, mean percent closure. Test-frame ranking is a diagnostic; activation-change and raw random controls use the same displayed budgets.

Ranking	10	30	100	300	1,000	3,000
Discovery attribution	0.19	0.40	1.49	4.65	13.61	23.65
Test-frame attribution	4.6	9.6	16.7	23.4	28.5	29.2
Activation change	−0.1	0.0	0.5	1.1	2.6	4.6
Raw random	0.0	0.0	0.0	0.0	0.6	1.2

Additional diagnostics. Only eight of the top 100 attribution features overlap the 452 Spatial box-selective features. Their selectivity-rise/attribution Spearman correlation is approximately -0.0194; Object box has zero top-set overlap and correlation -0.0151. Thus the operational rankings dissociate despite reproducible selectivity.

Full-MLP patches restricted to occluder-overlapping tokens close 45.62% [38.53, 56.79], versus 9.89% [4.86, 15.31] for other tokens. Corresponding attribution-set effects are 1.13% and 0.25%. These token-scope effects are not additive. Across pooled Spatial box intervention observations, predicted and measured projection correlate at $r = 0.28$, slope 0.06; screen and paint correlations are approximately 0.74 and 0.77. Repeated frames and overlapping sets limit independent-sample interpretation.

Attribution's advantage over random is 5.17 points on discovery frames and 1.45 on held-out frames. Cross-task attribution closure is 0.65%, with a reused-control contrast of approximately 0.62 points. The dictionary and control qualifications prevent treating this as fully isolated cross-task training.

The full supplementary tables preserve training and validity outputs, all family-specific method results, restoration, token scopes, budget curves, paired contrasts, selectivity replication, first-order diagnostics, in-sample comparisons, and every rollout outcome. The results archive duplicates extracted files and should not be counted as independent replication.

E Evaluation Stages and Behavioral Counts

The protocol separates dictionary learning, feature selection, finite action patches, and executed rollouts. Action-chunk measurements and closed-loop success are evaluated separately. Clean donor observations during restoration are privileged counterfactual inputs.

Table 6: Closed-loop success. Each condition contains 60 episodes across five tasks. Paired differences use task-bootstrap intervals; counts are not 60 independent task replications.

Condition	Successes	Rate (%)
Clean	58/60	96.67
Box	10/60	16.67
Screen	33/60	55.00
Paint	44/60	73.33
Box + restore all MLP	59/60	98.33
Box + restore attribution	8/60	13.33
Box + restore random	8/60	13.33
Box + restore selective	11/60	18.33
Clean + inject attribution	60/60	100.00
Paired contrast	Percentage points [95% CI]	
Attribution restore − box	−3.33 [−15.00, 5.00]	
Attribution − random restore	0.00 [−11.67, 13.33]	
Full-MLP restore − box	81.67 [68.33, 93.33]	

F Visual-Condition Contact Sheets

The following contact sheets show the visual conditions used in the controlled counterfactual study. Each row is one retained Spatial probe observation and each column is one rendered condition. These figures are illustrative checks of condition construction, not additional behavioral trials.



Figure 8: Third-person-camera contact sheet. Columns show base, recolor, absent, paint occlusion, paint miss, distractor paint, box, box miss, distractor box, screen, and screen miss. Rows show task IDs 0, 1, 2, 3, 5, 7, 8, and 9.



Figure 9: Wrist-camera contact sheet for the same retained observations and conditions. Screen construction targets the third-person camera; retained wrist information is a limitation of interpreting screen results as complete visual deprivation.