ONE-STEP STEERING IS NOT SEQUENTIAL CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoder (SAE) features give steering methods activation directions, but a direction specifies only one step: once the first intervention moves the hidden state, the second must be evaluated at a state where it was never specified. We call the rule that resolves this the *execution law* and study it for encoder–decoder controllers. We derive an exact identity showing that re-encoding after each step adds an interaction defect driven by the encoder–decoder round trip, and prove that good reconstruction does not bound this defect. Across eight public GPT-2 SAE widths, wider dictionaries reconstruct better while their round-trip error grows monotonically. On 493 held-out feature pairs in GPT-2 and Gemma-2-2B, the identity predicts which application order produces less interaction (Spearman 0.98–0.99), although accuracy falls when a controlled feature changes support. Choosing the predicted order cuts the interaction residual by 24–38%, yet direct error to the joint target changes by about 1% or less, because interaction accounts for only 4–7% of endpoint error and can cancel singleton errors. Reconstruction, interaction fidelity, and target delivery are therefore separate evaluation axes, and sequential steering must specify its execution law and measure outcomes directly.

1 INTRODUCTION

Representation interventions modify internal activations to change attributes, reasoning patterns, or generated text. Many methods describe an intervention as a direction: linear steering adds a vector to one hidden state, while newer methods learn curved paths, flows, coordinates, or input-conditioned selectors (Turner et al., 2023; Zou et al., 2023; Wu et al., 2024; Wurgaft et al., 2026; Jin et al., 2026; Wang et al., 2026; Fereidouni et al., 2026). This representation is sufficient for one step. It is incomplete for two: after the first intervention moves the hidden state, the second direction must be reevaluated somewhere it was not originally specified.

Two common execution rules illustrate this choice. Ambient re-encoding encodes the current hidden state again after every step. Residual-preserving execution records the initial reconstruction residual and changes only the decoded coordinate. Both rules produce the same first velocity at the anchor, but they need not agree after the state moves. As a result, applying control v before w can differ from applying w before v. The leading local order effect is the Lie bracket

$$[X, Y](h) = J_Y(h)X(h) - J_X(h)Y(h), \tag{1}$$

which depends on how both fields vary around h, not only on their values at h. Figures 1 and 2 summarize this distinction.

The scientific issue is not whether one extension is always preferable. Ambient re-encoding follows the deployed encoder at every visited state; residual-preserving execution follows the algebra declared in latent coordinates. The issue is whether either implementation executes the claimed composition, how its error can be measured before joint execution, and whether that measurement says anything about control quality.

Classical bracket, Taylor, and saltation identities provide a hierarchy for this update rule. We specialize that hierarchy to learned encoder–decoder controllers and test each level. Fixed linear TopK cells admit an exact ordered solution. Controlled-feature transitions replace the smooth formula with event transport. Across both regimes, interaction error forms a distinct component of endpoint error and can add to or cancel singleton errors. This establishes a measurement principle: reconstruction, interaction fidelity, and target delivery answer different questions and require separate evaluation.

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.

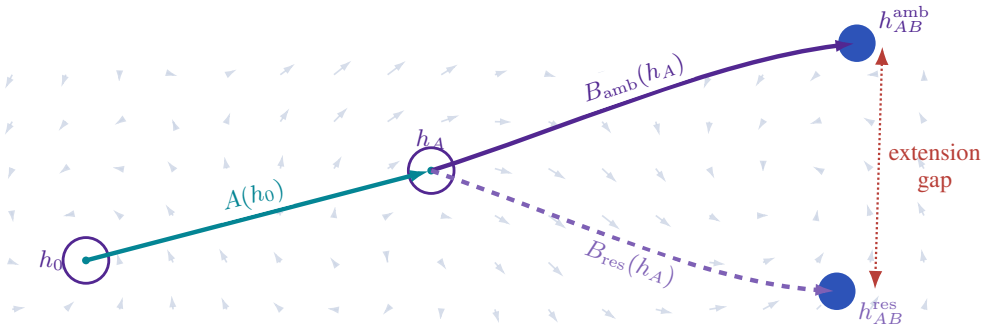


Figure 1: **A direction does not determine its sequential extension.** Both execution rules apply the same first control at h_0. After the state moves, ambient re-encoding (solid purple) and residual-preserving execution (dashed violet) evaluate the second control differently and reach distinct endpoints.

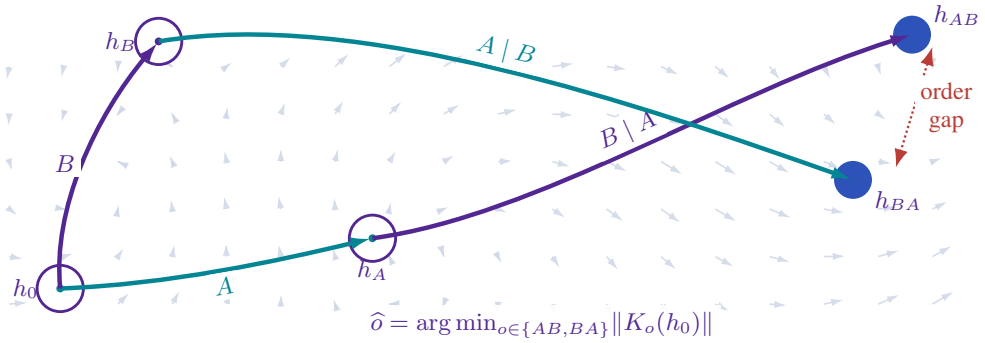


Figure 2: **Order follows different local fields.** Applying the same controls as AB or BA traces different paths. The SAE-specific ordered terms K_{AB} and K_{BA} forecast their interaction residuals, allowing the lower-error order to be selected before either joint trajectory is executed.

SAEs make the execution law explicit, but the principle applies to any controller that decodes learned coordinates and reevaluates them after the state moves.

Contributions. **(1) Execution-law decomposition.** We derive the exact re-encoding defect for encoder–decoder feedback and state, via a classical extension criterion, when on-manifold control values identify their sequential composition. The defect has a curvature term and a control-variation term, each driven by the differential cycle error, and connects to event transport at sparse boundaries. **(2) Empirical identification.** We measure differential cycles in 21 public SAEs, evaluate ordered interaction on 493 held-out feature pairs, and localize the transition from exact affine prediction to controlled-feature switching events. **(3) Consequences for control evaluation.** Geometry selects lower- interaction executions before either joint endpoint is observed. Comparing the same executions on interaction residual, direct error, named targets, and joint controllers establishes which quantities agree and which require separate evaluation.

2 Related Work

Representation steering. Linear activation addition and representation engineering treat concepts as directions that can be inserted into a model computation (Turner et al., 2023; Zou et al., 2023; Wu et al., 2024). Manifold, flow, and input-conditioned methods replace one fixed vector with curved trajectories or state-dependent selectors (Wurgaft et al., 2026; Jin et al., 2026; Wang et al., 2026; Fereidouni et al., 2026). GeoSteer similarly optimizes an adaptive geodesic for one learned objective (Ngo et al., 2026). These methods motivate state-dependent execution but do not determine the

ordered composition of two independently declared controls. The present question begins after that dependence is specified: when two controls are applied sequentially, which off-manifold extension and support-transition rule determine the second control? We analyze the resulting execution law rather than propose another semantic direction or selector. SAE features can produce behavioral steering when feature identification and intervention placement are carefully matched (He et al., 2025), while broad comparisons find prompting and supervised representation methods stronger than SAE steering on many tasks (Wu et al., 2025). These results motivate evaluating execution fidelity and behavioral success as separate outcomes.

Sparse representation quality. Sparse-autoencoder evaluation emphasizes reconstruction, sparsity, interpretability, and feature completeness (Cunningham et al., 2023; Lieberum et al., 2024; Chanin et al., 2024). Disentangled or interpretable directions need not remain compositionally consistent after an earlier intervention changes the state. The proposed metric is the differential encoder-after-decoder cycle on realized supports. This quantity tests tangent fidelity and is distinct from ordinary value reconstruction. Related inverse-consistent autoencoders enforce encoder–decoder compatibility architecturally (Otto et al., 2023); the present setting is overcomplete, sparse, and support-dependent, with sequential intervention error as the target. Feature dissection and sparse operator closure address semantic basis quality and native model operators (Hiramatsu et al., 2026; Feng & Li, 2026); neither implies closure of externally imposed intervention fields.

Differential and hybrid control. Lie-bracket naturality, accessibility, and extension results are classical (Lee & Lee, 2012; Jurdjevic, 1996). Saltation matrices describe first-order sensitivity at discrete switching events (Burden et al., 2014; Kong et al., 2023). We specialize these tools to learned encoder–decoder controllers, derive the exact re-encoding defect, and evaluate it on held-out feature pairs. The contribution is the execution-specific defect, its measurement separation from reconstruction and target delivery, and its empirical use for predicting ordered errors. Sequential-learning work uses brackets to rank parameter-update orders (Sweeney, 2026; Piontkovskaia & Nikolenko, 2026); our object is the activation-space execution law of an encoder–decoder controller. Prediction of singleton SAE side effects (Duan, 2026) and feature-composition instability (Zhou, 2026) are complementary: we isolate the pairwise error added by sequential execution and test whether reducing it improves direct joint control. Appendix M gives the expanded comparison.

3 A CONTROL NEEDS AN EXTENSION

A steering direction is an arrow attached to the current hidden state. A sequential controller must also specify which arrow to use after the first step has moved the state. We call this second rule an *extension*: it turns a direction in coordinate space into a vector field on every state the execution visits. Two implementations can agree on the first arrow and disagree after one step, just as two maps can give the same initial heading but different headings along the route. This section defines the two extensions used in practice and states when the declared controls determine their composition.

Notation. Hidden states live in $\mathcal{H} = \mathbb{R}^d$ and learned coordinates in $\mathcal{U} \subset \mathbb{R}^k$. The decoder $D : \mathcal{U} \to \mathcal{H}$ maps coordinates to the learned surface $M = D(\mathcal{U})$, and the encoder $E : \mathcal{H} \to \mathcal{U}$ assigns coordinates to a hidden state. For a sparse autoencoder, the relevant coordinates are the active features; Section 4 treats overcompleteness. A latent control $v(z) \in \mathbb{R}^k$ specifies how the coordinates should change at z, and the decoder converts it into the hidden velocity $J_D(z)v(z)$. For two controls, the Lie bracket $[v, w] = J_w v - J_v w$ is the leading-order effect of applying v then w rather than w then v. Its value depends on how each control varies around the anchor, not only on the two arrows at the anchor. The exact bracket identities below require $D \in C^2$ and $E, v, w \in C^1$ near the anchor. Finite-step $O(\epsilon^3)$ remainders additionally require C^3 maps with bounded derivatives along every compared trajectory. Sparse support changes violate these assumptions and are treated in Section 5 and Appendix T.

Residual-preserving execution. The first rule holds fixed the part of the hidden state that the decoder does not reconstruct. Let $z_0 = E(h_0)$, define the anchor residual $r_0 = h_0 - D(z_0)$, and write $H_r(z) = D(z) + r$. For the latent flow ϕ_t^v,

$$\Psi_t^{r,v}(H_r(z)) = H_r(\phi_t^v(z)). \tag{2}$$

In words: evolve the coordinates under v, decode them, and add back the same residual at every time. This is a local flow on the surface $M_r = H_r(\mathcal{U})$, not a global foliation of hidden space. Its hidden field is $X_v^{(r)}(H_r(z)) = J_D(z)v(z)$.

Proposition 1 (Algebra preservation)**.** *On every neighborhood where H_r is embedded and the latent trajectory remains inside its domain, Equation* (2) *is the pushforward of the latent flow. Consequently,*

$$\left[X_v^{(r)}, X_w^{(r)}\right](H_r(z)) = J_D(z)\,[v, w]\,(z). \tag{3}$$

Residual-preserving execution therefore reproduces in hidden space exactly the algebra declared in latent space. Decoder curvature enters both orders and cancels (Appendix B).

Ambient re-encoding. The common alternative re-encodes the current hidden state after every step,

$$W_v(h) = J_D(E(h))\,v(E(h)). \tag{4}$$

At the anchor, both rules produce the same velocity. They separate only once the state moves: one keeps the anchor residual, while the other asks the encoder for new coordinates. A one-step steering check cannot distinguish them. The bracket also connects directly to code. For the Euler step $F_v^\epsilon(h) = h + \epsilon W_v(h)$ used by many steering implementations,

$$F_w^\epsilon \circ F_v^\epsilon(h) - F_v^\epsilon \circ F_w^\epsilon(h) = \epsilon^2\,[W_v, W_w]\,(h) + O(\epsilon^3). \tag{5}$$

When declared controls determine their composition. The following classical result states exactly what an on-manifold direction leaves unspecified.

Theorem 1 (Classical extension criterion; cf. bracket naturality and extension lemmas in Lee & Lee (2012))**.** *Let $M \subset \mathcal{H}$ be an embedded submanifold and suppose the values of smooth ambient fields X, Y are prescribed on M. Then $[X, Y]\,(h)$ is determined by $X|_M, Y|_M$ if and only if $X(h), Y(h) \in T_h M$. If either condition fails, then for every $q \in T_h\mathcal{H}$ there are smooth extensions with the same values on M whose brackets at h differ by q. If, in addition, both restrictions are tangent to M on a neighborhood of h in M, the determined bracket is the intrinsic bracket and is tangent to M.*

Knowing the control arrows on the learned surface is therefore enough only when both arrows are tangent to it. Otherwise, infinitely many off-surface continuations agree with every observed value yet produce arbitrary brackets at the anchor. Values on M are zeroth-order data, whereas composition depends on first-order transverse data: how the arrows change off M (Appendix B). The extension also changes what the controller can reach. Residual-preserving lifts stay within a distribution of dimension at most k, whereas an ambient extension enlarges the local accessibility distribution whenever its bracket has a transverse component (Jurdjevic, 1996). The new object in this paper is not Theorem 1 but the encoder–decoder defect it points to, which we derive next.

4 Exact re-encoding defect

Ambient re-encoding departs from residual-preserving execution only through the encoder. Each step decodes a latent direction into a hidden velocity, moves the state, and encodes again to choose the next direction. The question is whether this decode–encode round trip preserves an infinitesimal latent direction.

Differential cycle. For $z = E(h)$, the differential cycle and its error are

$$C_h = J_E(h)J_D(z), \quad A_h = C_h - \mathrm{I}_k. \tag{6}$$

C_h decodes an infinitesimal latent arrow with J_D and encodes the result with J_E. If every arrow returns unchanged, $C_h = \mathrm{I}_k$. The cycle error A_h records gain error on its diagonal and cross-feature leakage off it: entry $(A_h)_{ij}$ with $i \neq j$ measures how much moving along decoder direction j changes encoder coordinate i.

The defect. Applying the chain rule to Equation (4), differentiating along W_v, exchanging v and w, and subtracting gives

$$[W_v, W_w](h) = H_D(z)[C_h v, w] - H_D(z)[C_h w, v] + J_D(z)\{J_w(z)C_h v - J_v(z)C_h w\}. \tag{7}$$

Substituting $C_h = \mathrm{I}_k + A_h$ separates the intended bracket $J_D[v, w]$ from the exact re-encoding defect

$$\Delta_h(v, w) = H_D[A_h v, w] - H_D[A_h w, v] + J_D\{J_w A_h v - J_v A_h w\}. \tag{8}$$

The executed bracket equals the intended bracket plus Δ_h, and every term of Δ_h contains A_h. The first line is the *curvature term*: decoder curvature converts cycle error into interaction. The second is the *control-variation term*: cycle error changes the input to a state-dependent control. Appendix B derives each term, and Figure 5 confirms that the two terms reproduce the defect measured by automatic differentiation to numerical precision.

Which term is active. For a linear decoder, $H_D = 0$ and the curvature term vanishes. Our controlled experiments use linear SAE decoders with state-dependent feedback controls, so their entire defect comes from the control-variation term. Conversely, for constant, commuting latent translations the control-variation term vanishes and only curvature remains:

$$\boxed{[W_v, W_w] = H_D[A_h v, w] - H_D[A_h w, v].} \tag{9}$$

This case requires both decoder curvature and cycle error. It covers the most common intervention, adding a fixed decoder vector $T_i(h) = h + \alpha_i d_i$ at one layer. With a linear SAE decoder, Equation (9) is exactly zero for every cycle matrix, so standard additive SAE steering commutes. The phenomenon studied here belongs to state-dependent controllers that re-encode; fixed additions serve as a negative control.

Reconstruction does not bound the defect.

Proposition 2 (Reconstruction does not identify differential composition)**.** *For every sufficiently small $\delta > 0$, there is a smooth encoder–decoder pair whose reconstruction error on every compact subset $\mathcal{K}$ of the decoded manifold is at most $C_{\mathcal{K}}\delta$, while commuting latent translations have a false interaction of magnitude one at anchors spaced $O(\delta)$ apart on that manifold. Hence every nonempty open subset of the manifold contains such an anchor for all sufficiently small δ.*

A dictionary can reconstruct every point on its decoded manifold increasingly well while its interaction error stays at one on an increasingly dense set of points. Reconstruction checks positions; composition checks derivatives (construction in Appendix B). The statement concerns the local derivative, not a finite-step gap. For exact ambient flows of duration ϵ at the origin,

$$h_{AB} - h_{BA} = \frac{\delta\epsilon}{2}\sin\left(\frac{2\epsilon}{\delta}\right)e_3, \qquad \|h_{AB} - h_{BA}\| \le \frac{\delta\epsilon}{2}, \tag{10}$$

so the gap normalized by ϵ^2 tends to one as $\epsilon \to 0$ at fixed δ, while the absolute gap vanishes as $\delta \to 0$ at fixed ϵ. The limits do not commute because higher encoder derivatives grow as $\delta \to 0$.

Three separate measurements. A control must also reach its named target. For feedback controls $v_i(z) = (\tau_i - z_i)e_i$ with $\tau_i \neq E_i(h_0)$, define the singleton delivery error

$$T_i = \left| \frac{E_i(\Phi^1_{W_i}(h_0)) - E_i(h_0)}{(1 - e^{-1})\big(\tau_i - E_i(h_0)\big)} - 1 \right|, \tag{11}$$

where the denominator is the unit-time coordinate change under the ideal latent dynamics $\dot{z}_i = \tau_i - z_i$. Let $I = \|[W_1, W_2] - J_D[v_1, v_2]\|$ at h_0 denote the interaction defect.

Proposition 3 (Pointwise measurement separation)**.** *Reconstruction error, differential interaction defect, and singleton target delivery do not certify one another. More precisely:*

1. *reconstruction can converge uniformly to zero on compact subsets of the decoded manifold while $I = 1$ on a grid with spacing $O(\delta)$;*

2. *there are smooth linear encoder–decoder systems with exact anchor reconstruction and* $I = 0$*, while* $\max_i T_i \to 1$*;*

3. *there are smooth linear systems with exact anchor reconstruction and* $T_1 = T_2 = 0$*, while* $I = 1$.

Reconstruction asks whether a state is reproduced, interaction asks whether two controls compose as declared, and delivery asks whether one control reaches its target. Passing one test does not imply passing another; the proposition delimits what each measurement certifies under minimal pointwise premises (explicit systems in Appendix B). Section 5 shows that the separation is not only logical: large reductions in interaction coexist with negligible changes in direct error.

Overcomplete dictionaries. A sparse autoencoder with $m > d$ features cannot satisfy $C_h = \mathrm{I}$ globally, since $\mathrm{rank}(W_{\mathrm{enc}} W_{\mathrm{dec}}) \le d < m$. Sequential control needs only a restricted property. Inside a cell with active support S, the relevant cycle is the $|S| \times |S|$ matrix $C_S = W_{\mathrm{enc},S:} W_{\mathrm{dec},:S}$, which can be close to identity even though the global product cannot. Appendix C defines a coordinate-invariant cycle error $\|A_S\|_{G_S}$ weighted by the decoder metric and an exact fixed-support repair.

Public SAEs We compute C_S exactly on 210,000 realized supports from 21 public GPT-2 Small SAEs. No dictionary is near identity on typical supports (Table 2). In a controlled sweep of eight widths at GPT-2 block 8 residual input (`blocks.8.hook_resid_pre`), reconstruction improves as width grows (FVU falls from .400 to .172) while median cycle error rises monotonically from 1.13 to 3.91 (Spearman -1.00, exact permutation $p = 4.96 \times 10^{-5}$). The trend holds within every support-size bin (Spearman ≤ -0.93; Figure 8). Decoder conditioning amplifies the trend but does not explain it: after adjusting for conditioning, 44% of the endpoint log effect remains (Appendix N). In in-house SAEs, cycle error can also be trained down. A support-cycle penalty lowers it by 28–42% for a 2–7% increase in FVU, although the downstream benefit for control depends strongly on the feature cohort (Appendices D and G). These statistics measure a missing premise rather than a behavioral failure: by Equation (8), cycle error matters only through decoder curvature or state-dependent controls.

Measuring execution fidelity. We report five quantities: value reconstruction, which tests state preservation; the cycle error $\|J_E J_D - \mathrm{I}\|$, which tests tangent accuracy; a commuting-null bracket and the defect in Equation (8), which test extension error; and a finite flow loop, which checks the differential calculation against finite execution. The released Manifold Steering implementation falls outside this smooth setting. Its nearest-centroid encoder is piecewise constant, so $J_E = 0$ almost everywhere, and its paths do not re-encode; Appendix O analyzes it separately.

5 From execution geometry to measured control

We test whether the defect in Section 4 predicts interaction before joint execution, where that prediction fails, and whether lower interaction improves joint control.

5.1 Experimental setup

SAEs and controller. We train in-house linear TopK SAEs (width 16,384, 64 active features) with three seeds on GPT-2 Small block 8 residual output (`blocks.8.hook_resid_post`) and `google/gemma-2-2b-it` block 12 residual output (`blocks.12.hook_resid_post`); blocks are indexed from zero. Each control drives one SAE coordinate toward a target, $v_i(z) = (\tau_i - z_i)e_i$, with strength $0.02\,\|h\|$. Controls are executed by ambient re-encoding for unit time with RK4; conclusions are unchanged across 4, 8, and 16 substeps (Appendix J).

Interaction residual. Each control can miss its target even when applied alone, so we isolate the extra error created by combining two controls. Let $\bar{h}_A, \bar{h}_B$ be the singleton residual-preserving references and $\bar{h}_{AB}$ the joint residual-preserving reference, which adds both latent increments and is therefore order-invariant. For ambient endpoints h_A, h_B, h_{AB}, h_{BA}, the interaction residuals are

$$\begin{aligned} R_{AB} &= (h_{AB} - \bar{h}_{AB}) - (h_A - \bar{h}_A) - (h_B - \bar{h}_B), \\ R_{BA} &= (h_{BA} - \bar{h}_{AB}) - (h_A - \bar{h}_A) - (h_B - \bar{h}_B). \end{aligned} \tag{12}$$

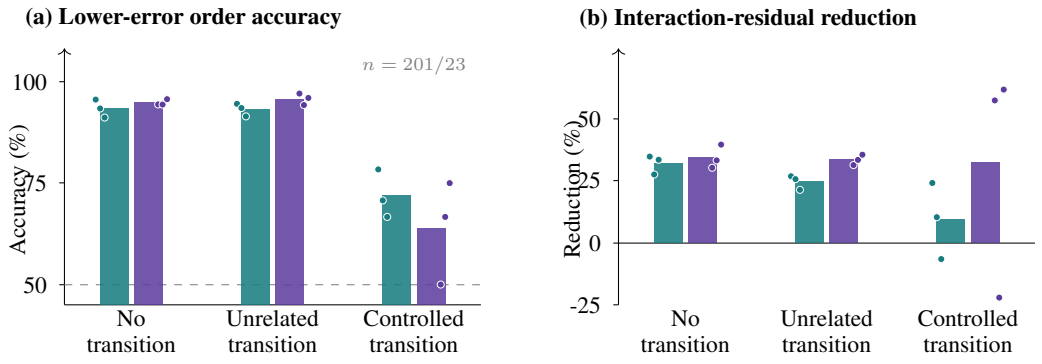


Figure 3: **Geometry predicts order until a controlled feature changes support.** (a) Accuracy of choosing the lower-interaction order. (b) Reduction in interaction residual relative to the default order. Bars show model-family means (teal: GPT-2; purple: Gemma-2-2B); points show seeds; n gives the number of controlled-transition rows (GPT-2/Gemma).

Here $e_o = h_o - \bar{h}_{AB}$ is the *direct error*. Subtracting the singleton errors leaves the non-additive part, so a small R_o means that singleton errors add, not that the endpoint is close to its target. The algebraic reference itself has 33–60% larger named-target errors than geometry-ordered execution (Table 1).

Protocol. The primary outcome is row-weighted interaction reduction relative to an order chosen on development pairs. Pair identities are excluded from earlier fits, and decisions precede joint execution. Pair-cluster intervals remain conditional on three seeds. Each model contributes 3,072 rows (1,024 per seed) and 381–462 distinct pairs per seed; Appendix A gives the complete protocol.

5.2 Geometry predicts interaction inside a support cell

Inside a fixed TopK cell, the affine encoder gives exactly $R_{AB} = f_i(1)f_j(1)K_{AB}$ and $R_{BA} = f_i(1)f_j(1)K_{BA}$, where $K_{AB} = J_{W_B}W_A$ and $K_{BA} = J_{W_A}W_B$ (Appendix J). These SAE-computable ordered Taylor terms provide an implementation check; the nontrivial tests are the controls below and rows where the fixed-cell premise fails.

On 242 GPT-2 and 251 Gemma-2-2B held-out pairs, the ordered terms predict interaction magnitude with Spearman .98 and .99 and median vector cosine 1.00, and at RK4-16 they identify the lower-interaction order 94% and 93% of the time (Figure 13). Three controls show that this signal comes from pair-specific geometry rather than intervention scale. First, an exact cycle ($A_h = 0$) predicts no interaction at all. Second, cyclically reassigning predictions among pairs, which preserves their distribution but breaks pair identity, drops Spearman (at RK4-8) from .98 to $-.01$ on GPT-2 and from .99 to .16 on Gemma, and order accuracy to chance (53% and 49%). Third, keeping the two orders separate improves Spearman by .13 and .12 over the best direction-blind score (Table 8). A smaller ridge comparison on 49 pairs points the same way: pair geometry is the only pre-execution predictor with positive held-out R^2 on both models, reaching Spearman .85 and .86 versus at most .34 for reconstruction, conditioning, or aggregate cycle error (Table 7). The original predictor used checkpoint rather than execution coordinates; correcting this after the outcomes were known raises Spearman to 1.00 and order accuracy to 97–98% (Appendix D). We report the original prospective values throughout.

5.3 Prediction fails when a controlled feature changes support

Unrelated support changes preserve the controlled affine coordinates: order accuracy remains 91–97%, with 21–36% lower interaction. Controlled-feature transitions break this premise; accuracy falls to 50–78%, and two of six cells reverse. From RK4-16 to RK4-64, 75% of these rows change support path and 8% change the preferred order, unaffected by float64 or batching. Event identity and timing therefore matter. On a discovery cohort, saltation corrections recover 32 of 33 cases (Appendix T).

Table 1: **Lower interaction residual leaves direct error nearly unchanged.** Means across three seeds, weighting feature pairs equally (the headline reductions in the text are row-weighted). Interaction residual is $\|R_o\|$ from Equation (12); direct error is the distance to the joint reference. Dashes are undefined or unevaluated; controller differences are descriptive.

Controller	Interaction residual	Direct hidden error	Direct logit error	Named-target error (A/B)	Worst named-target error
GPT-2					
Default order	1.783×10^{-4}	.002015	.854	.174/.221	.250
Geometry order	1.342×10^{-4}	.002013	.854	.173/.221	.251
Residual-preserving ref.	–	0	0	.254/.303	.342
Simultaneous	–	.002003	.848	.173/.221	.250
Symmetric split	–	.002004	.849	.173/.221	.250
Gemma-2-2B					
Default order	9.921×10^{-5}	.000644	.399	.080/.080	.098
Geometry order	6.497×10^{-5}	.000638	.396	.081/.080	.099
Residual-preserving ref.	–	0	0	.128/.127	.155
Simultaneous	–	.000635	.395	.080/.079	.097
Symmetric split	–	.000635	.395	.080/.079	.097

5.4 Lower interaction does not mean better joint control

For each prompt and held-out pair, the geometry rule chooses the order with the smaller predicted residual, $\widehat{o} = \arg\min_o \|K_o(h_0)\|$, before either joint endpoint exists, using one geometry evaluation and no rollout. It lowers row-weighted interaction by 24% on GPT-2 and 38% on Gemma-2-2B (25% and 35% with equal pair weighting). All seed-level effects are positive, final targets remain within tolerance, and centered-logit interaction falls by 17–21% and 32–39% with pair-cluster intervals below zero (Table 5).

Table 1 shows the other side. Direct error barely moves: it improves by at most 0.9% in hidden space and 1.2% in logits, some cells worsen slightly, and most pair-cluster intervals include zero. The worst named-target error worsens by 0.2–1.5% in all six cells, and on GPT-2, retention of the first-applied target falls from .945 to .909.

Two facts explain the gap. First, interaction is a small part of endpoint error: the median ratio $\|R_o\| \,/\, \|e_o\|$ is .04 on GPT-2 and .06–.07 on Gemma, although its 95th percentile reaches .19–.40. Second, interaction can cancel singleton error. Among rows where geometry lowers interaction, direct error still worsens in 38–57% of cases, and 68–89% of those reversals have the sign expected from losing a favorable cancellation. Reducing one component of endpoint error need not reduce the total.

Simple joint control is also competitive. A unit-time flow of $W_A + W_B$ and the symmetric split $A/2, B, A/2$ use the same field-evaluation budget as one ordered execution, yet both have slightly lower mean direct and worst named-target errors than geometry ordering in every model–seed cell (by 0.15–0.73% and 0.33–2.50%). An oracle that executes both orders and keeps the better one gains only 2.4–4.5% in direct error at twice the cost, while worsening worst-target error by 1.2–2.6%.

Geometry cheaply selects the lower-interaction execution, but does not replace direct target and behavioral evaluation or favor sequential over joint control.

6 Discussion

Interaction residual, total distance, named-target accuracy, behavior, and safety measure different properties. Order selection changes the first while leaving the larger direct error nearly unchanged.

Reconstruction, intervention magnitude, and local geometry answer different questions. Even exact prediction of the interaction term need not improve the user-facing objective because singleton error can dominate or cancel that term. An interaction score therefore cannot replace direct target and

behavioral evaluation. Appendix O shows a related sensitivity in a released controller: its geometric score depends strongly on residual handling even though its smooth defect is zero.

What geometry checks. Inside a fixed affine cell, the formula is exact and inexpensive, making it a direct test of whether the implementation follows its declared execution law. At controlled-feature events, smooth Taylor terms are insufficient and event identity, location, and ordering matter. Matched interaction vectors also differ across independently trained charts, and binary-order portability remains unknown. Geometry therefore provides an execution-specific consistency check under the stated controller and chart assumptions. Event-aware analysis and direct outcomes are needed beyond the measured execution law.

7 LIMITATIONS

Our experiments cover one state-dependent feedback law, in-house TopK SAEs on GPT-2 and Gemma-2-2B, and activation or centered-logit outcomes. Singleton controls moved activations and logits without improving behavior, and local loops changed no top-1 token even at low-margin states, so our claims concern execution geometry rather than task success. Prediction is least reliable at controlled-feature transitions (50–78% order accuracy) and across regularized training objectives (directional Spearman near $.55$). Cycle-regularization gains are large on a small selected cohort but modest on a broad one. The public-SAE analysis covers two public releases for GPT-2 Small. All intervals are conditional on three training seeds, auxiliary comparisons carry no multiple-comparison correction, and selection rules were timestamped before outcomes but not publicly preregistered.

8 CONCLUSION

One-step steering specifies a velocity; sequential control must also specify how that velocity changes once the state moves. Re-encoding adds an exact defect that reconstruction cannot bound, and in public SAEs the round-trip error behind it worsens as reconstruction improves. SAE geometry predicts the resulting interaction inside a support cell but loses reliability when a controlled feature changes support. Choosing the lower-interaction order cuts interaction by 24–38% yet leaves direct error nearly unchanged. Sequential steering therefore needs an explicit execution law, event-aware corrections, and direct outcome evaluation.

Reproducibility statement

The supplement provides sanitized configurations, seeds, split definitions, selection rules, standalone analysis scripts, row-level numerical outcomes, and machine-readable summaries. The selection rules were fixed before the corresponding outcomes were evaluated. The full code, execution scripts, tests, and remaining artifacts will be released upon acceptance. Headline values in the main text are rounded for readability, typically to two significant digits or whole percentages. Appendix tables retain additional digits when they are needed to reproduce a threshold, numerical check, or comparison.

Ethics Statement

This work studies the geometry of interventions in pretrained language models. It does not involve human participants, collect personal data, or deploy a system to users. Experiments use public model checkpoints and datasets or synthetic prompts under their applicable licenses. Steering methods can alter model outputs in unintended ways; activation- and logit-level improvements do not establish behavioral reliability, safety, or suitability for deployment. We therefore report failed behavioral tests and restrict our claims to the measured execution settings.

AI Use Statement

We used LLMs to assist with revising grammar, style and text flow in this manuscript and to help write figure code.

REFERENCES

Sergio Blanes, Fernando Casas, and Ander Murua. Splitting and composition methods in the numerical integration of differential equations. 2008. URL https://api.semanticscholar.org/CorpusID:14984715.

Samuel A. Burden, S. Shankar Sastry, Daniel E. Koditschek, and Shai Revzen. Event-selected vector field discontinuities yield piecewise-differentiable flows. *SIAM J. Appl. Dyn. Syst.*, 15:1227–1267, 2014. URL https://api.semanticscholar.org/CorpusID:11944930.

David Chanin, James Wilken-Smith, Tomás Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Isaac Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *ArXiv*, abs/2409.14507, 2024. URL https://api.semanticscholar.org/CorpusID:272827216.

Hoagy Cunningham, Aidan Ewart, Logan Riggs Smith, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *ArXiv*, abs/2309.08600, 2023. URL https://api.semanticscholar.org/CorpusID:261934663.

Evan Duan. Pre-intervention prediction of sparse autoencoder steering side effects. *ArXiv*, abs/2606.08365, 2026. URL https://api.semanticscholar.org/CorpusID:289099418.

Karthik Elamvazhuthi, Xuechen Zhang, Samet Oymak, and Fabio Pasqualetti. Learning on manifolds: Universal approximations properties using geometric controllability conditions for neural odes. *ArXiv*, abs/2305.08849, 2023. URL https://api.semanticscholar.org/CorpusID:258685460.

Ji-Ming Feng and Jun-Liang Li. Scaled idempotence in transformer attention: Paired ov geometry and shared-value algebras. 2026. URL https://api.semanticscholar.org/CorpusID:291602196.

Moghis Fereidouni, Muhammad Umair Haider, Hassan Sajjad, and A. B. Siddique. Gaps: Dimension-level gates for conditional activation steering. 2026. URL https://api.semanticscholar.org/CorpusID:291611236.

Zirui He, Haiyan Zhao, Yi-Ran Qiao, Fan Yang, Ali Payani, Jing Ma, and Meng-Nan Du. Saif: A sparse autoencoder framework for interpreting and steering instruction following of language models. *ArXiv*, abs/2502.11356, 2025. URL https://api.semanticscholar.org/CorpusID:276409381.

Takeru Hiramatsu, Kyohei Atarashi, Koh Takeuchi, and Hisashi Kashima. Disentangling steering vectors. 2026. URL https://api.semanticscholar.org/CorpusID:291869341.

Md. Saiful Islam and Rahul Bhadani. Control-theoretic view of neural odes: Empirical controllability and observability. *ArXiv*, abs/2606.08431, 2026. URL https://api.semanticscholar.org/CorpusID:289099164.

Ze-Hao Jin, Ruixuan Deng, Jun-Ran Wang, Xin-Jie Shen, and Chao Zhang. Beyond steering vector: Flow-based activation steering for inference-time intervention. *ArXiv*, abs/2605.05892, 2026. URL https://api.semanticscholar.org/CorpusID:288013530.

Velimir Jurdjevic. *Geometric Control Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1996.

Nathan J. Kong, J. Joe Payne, James Zhu, and Aaron M. Johnson. Saltation matrices: The essential tool for linearizing hybrid dynamical systems. *Proceedings of the IEEE*, 112:585–608, 2023. URL https://api.semanticscholar.org/CorpusID:259138913.

Panagiotis Koromilas, Andreas D. Demou, James Oldfield, Yannis Panagakis, and Mihalis A. Nicolaou. Polysae: Modeling feature interactions in sparse autoencoders via polynomial decoding. *ArXiv*, abs/2602.01322, 2026. URL https://api.semanticscholar.org/CorpusID:285269421.

John M. Lee and John M. Lee. *Introduction to Smooth Manifolds*. Springer, New York, 2012. ISBN 978-1-4899-9475-2 978-1-4419-9982-5.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, J'anos Kram'ar, Anca D. Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *ArXiv*, abs/2408.05147, 2024. URL https://api.semanticscholar.org/CorpusID:271843380.

Walter A. Nelson, Theofanis Karaletsos, and Francesco Locatello. Toward identifiable sparse autoencoders. *ArXiv*, abs/2605.31245, 2026. URL https://api.semanticscholar.org/CorpusID:288815016.

Xuan Cuong Ngo, Hao Vo, and Ngan Hoang Le. Geosteer: Geodesic optimization for activation steering in large language models. 2026. URL https://api.semanticscholar.org/CorpusID:291902148.

Samuel E. Otto, G. Macchio, and Clarence W. Rowley. Learning nonlinear projections for reduced-order modeling of dynamical systems using constrained autoencoders. *Chaos*, 33 11, 2023. URL https://api.semanticscholar.org/CorpusID:260316286.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:265042984.

Irina Piontkovskaia and Sergey I. Nikolenko. First-order predictable but pairwise fragile: Local task adaptation in trained transformers. *ArXiv*, abs/2607.16821, 2026. URL https://api.semanticscholar.org/CorpusID:290377260.

Mehrshad Saadatinia, Parsa Razmara, Ardalan Aryashad, Ali Abbasi, and Seyedarmin Azizi. Circuitsteer: Geometrically aligned multi-layer steering via sparse autoencoder circuits. 2026. URL https://api.semanticscholar.org/CorpusID:290852200.

Benjamin Shih, John Winnicki, and Arianna Cao. How do language models choose between context and memory? 2026. URL https://api.semanticscholar.org/CorpusID:291544376.

John M. Sweeney. The geometry of sequential learning: Lie-bracket prediction of transfer order. *ArXiv*, abs/2606.24993, 2026. URL https://api.semanticscholar.org/CorpusID:289622206.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. *ArXiv*, abs/2310.15213, 2023. URL https://api.semanticscholar.org/CorpusID:264439657.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David S. Udell, Juan J. Vazquez, Ulisse Mini, and Monte Stuart MacDiarmid. Steering language models with activation engineering. 2023. URL https://api.semanticscholar.org/CorpusID:261049449.

Benjamin Walker, Andrew D. McLeod, Tiexin Qin, Yichuan Cheng, Haoliang Li, and Terry Lyons. Log neural controlled differential equations: The lie brackets make a difference. In *International Conference on Machine Learning*, 2024. URL https://api.semanticscholar.org/CorpusID:268041282.

Zheng Wang, Mu-Chen Li, Ren-Jie Liao, and Yan Leng. Ideea: training-free input-dependent steering via activation cluster matching. 2026. URL https://api.semanticscholar.org/CorpusID:291611100.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Daniel Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models. *ArXiv*, abs/2404.03592, 2024. URL https://api.semanticscholar.org/CorpusID:268889731.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Daniel Jurafsky, Christopher D. Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *ArXiv*, abs/2501.17148, 2025. URL https://api.semanticscholar.org/CorpusID:275931808.

Daniel Wurgaft, Can Rager, Matthew Kowal, Vasudev Shyam, Sheridan Feucht, Usha Bhalla, Tal Haklay, Eric J. Bigelow, Raphaël Sarfati, Tom McGrath, Owen Lewis, Jack Merullo, Noah D. Goodman, Thomas Fel, Atticus Geiger, and Ekdeep Singh Lubana. Manifold steering reveals the shared geometry of neural network representation and behavior. *ArXiv*, abs/2605.05115, 2026. URL https://api.semanticscholar.org/CorpusID:287988518.

Yun-Xiang Zhou. Structural instability of feature composition. *ArXiv*, abs/2605.05223, 2026. URL https://api.semanticscholar.org/CorpusID:288013288.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Troy Mallen, Steven Basart, Oluwasanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency. *ArXiv*, abs/2310.01405, 2023. URL https://api.semanticscholar.org/CorpusID:263605618.

A APPENDIX

The appendix contains the derivations, numerical controls, and reproducibility details. The main components are:

- **Detailed derivations** (p. 14): dimensions, chain rules, cancellation steps, and counterexamples.
- **Public SAE evidence and controlled-experiment details** (p. 22): cycle errors, conditioning controls, cycle-regularized training, order selection, direct error, joint controllers, support transitions, and the coordinate-convention correction.
- **Prediction and repair** (p. 31–34): held-out prediction, training tradeoffs, and operating regimes.
- **Support changes** (p. 56): event-time corrections, horizon effects, and matched nulls.
- **Gauge and conditioning controls** (p. 46): per-width values, support-size and condition-number analyses, and repair costs.

RELATION AMONG THE MEASUREMENTS

Direct joint control is the user-facing quantity. Reducing the interaction residual does not improve it in our experiments. The fixed-cell formula instead checks whether the code executes the derived local dynamics. Controlled-feature transitions mark the main boundary of that formula, while the effect of regularized training depends on the feature cohort and training objective. We also found no released workflow that requires the sequential feedback law used in the controlled experiments.

Appendix B first defines the notation and derives the defect. The remaining sections measure the differential cycle in public SAEs, predict held-out interactions, modify training, and analyze support transitions and limitations.

B MATHEMATICAL DERIVATIONS

This section gives the intermediate steps for every main mathematical claim. The derivations use multivariable calculus and elementary linear algebra.

1. **Objects:** dimensions and smoothness assumptions.
2. **Compare execution rules:** prove why residual preservation keeps the latent bracket and why arbitrary ambient extensions need not.
3. **Derive the defect:** apply the chain rule once, substitute the differential cycle, and separate intended from unintended terms.
4. **Separate measurements:** construct explicit examples showing why reconstruction, interaction, and target delivery require different tests.

B.1 NOTATION, DIMENSIONS, AND ASSUMPTIONS

Hidden states are column vectors $h \in \mathbb{R}^d$ and latent coordinates are column vectors $z \in \mathbb{R}^k$. The maps and vector fields have the following roles:

- $E : \mathbb{R}^d \to \mathbb{R}^k$ converts a hidden state into latent coordinates.
- $D : \mathbb{R}^k \to \mathbb{R}^d$ converts latent coordinates back into hidden space.
- $v, w : \mathbb{R}^k \to \mathbb{R}^k$ are latent control fields. At coordinate z, the vector $v(z)$ states the requested instantaneous latent change.
- $W_v, W_w : \mathbb{R}^d \to \mathbb{R}^d$ are the corresponding hidden-space fields after decoding and re-encoding.

The decoder and encoder derivatives have shapes

$$J_D(z) \in \mathbb{R}^{d\times k}, \qquad J_E(h) \in \mathbb{R}^{k\times d}. \tag{13}$$

For vector fields X, Y on $\mathbb{R}^n$, we use the Lie bracket convention

$$[X, Y](x) = J_Y(x)X(x) - J_X(x)Y(x). \tag{14}$$

In particular,

$$[v, w](z) = J_w(z)v(z) - J_v(z)w(z). \tag{15}$$

Multiplying $J_D(z)$ by $v(z)$ converts a latent velocity into a hidden velocity. The product

$$C_h = J_E(h)J_D(z) \in \mathbb{R}^{k\times k}, \qquad A_h = C_h - I_k, \tag{16}$$

first decodes an infinitesimal latent change and then encodes it again. Thus $C_h = I_k$ means that this infinitesimal round trip is exact. The matrix A_h records its error.

One entry has a simple interpretation. Let d_j be decoder direction j and let E_i be encoder coordinate i. Then

$$(C_h)_{ij} = \nabla_h E_i(h)^\top d_j. \tag{17}$$

The diagonal entry $(C_h)_{ii}$ measures how much of a requested change in coordinate i returns to the same coordinate. An off-diagonal entry $(C_h)_{ij}$ measures leakage: moving along decoder direction j also changes encoder coordinate i. Subtracting the identity turns perfect diagonal return into zero, so A_h contains only gain error and cross-coordinate leakage.

For a vector-valued decoder, $H_D(z)[a, b] \in \mathbb{R}^d$ denotes the vector whose jth component is $a^\top \nabla^2 D_j(z)b$. Each scalar Hessian is symmetric, so

$$H_D(z)[a, b] = H_D(z)[b, a]. \tag{18}$$

The exact bracket identities require $D \in C^2$ and $E, v, w \in C^1$. The finite-step expansion with an $O(\epsilon^3)$ remainder uses C^3 maps with bounded derivatives on a neighborhood containing every compared trajectory. All statements in this section are local to a smooth support region. A TopK support change can make the encoder nonsmooth, so the formulas below no longer apply across that event. Appendix T treats those cases.

B.2 Why residual-preserving execution preserves the latent bracket

Pushforward identity. Residual-preserving execution carries the latent bracket into hidden space without adding an error term.

Fix a residual r and define $H_r(z) = D(z) + r$. The residual is constant, so $J_{H_r}(z) = J_D(z)$. Let $\phi_t^v(z)$ be the latent trajectory generated by v, which means

$$\left.\frac{d}{dt}\phi_t^v(z)\right|_{t=0} = v(z). \tag{19}$$

Residual-preserving execution decodes this trajectory and then adds the same r at every time:

$$\Psi_t^{r,v}(H_r(z)) = H_r(\phi_t^v(z)). \tag{20}$$

Differentiating at $t = 0$ and applying the chain rule gives

$$X_v^{(r)}(H_r(z)) = \left.\frac{d}{dt}H_r(\phi_t^v(z))\right|_{t=0} \tag{21}$$

$$= J_{H_r}(z)v(z) = J_D(z)v(z). \tag{22}$$

Thus $X_v^{(r)}$ is exactly the pushforward of v through H_r. The fixed residual contributes no derivative.

Lie Bracket preservation. Using 15, we obtain

$$\left[X_v^{(r)}, X_w^{(r)}\right](H_r(z)) = dX_w^{(r)}\big|_{H_r(z)}\left[X_v^{(r)}(H_r(z))\right] - dX_v^{(r)}\big|_{H_r(z)}\left[X_w^{(r)}(H_r(z))\right] \tag{23}$$

$$= d(J_D(z)w(z))\,[v(z)] - d(J_D(z)v(z))\,[w(z)]. \tag{24}$$

The field $z \mapsto J_D(z)w(z)$ is a product of two state-dependent terms. Moving in direction $v(z)$ changes both terms. The product rule gives

$$d(J_D(z)w(z))\,[v(z)] = H_D(z)[v(z), w(z)] + J_D(z)J_w(z)v(z). \tag{25}$$

The first term is the change in the decoder Jacobian. The second is the change in the latent field w. Exchanging v and w and subtracting yields

$$\left[X_v^{(r)}, X_w^{(r)}\right](H_r(z)) = H_D[v, w] - H_D[w, v] \tag{26}$$

$$+ J_D(J_w v - J_v w) \tag{27}$$

$$= J_D(z)\,[v, w]\,(z). \tag{28}$$

The two Hessian terms cancel because the decoder Hessian is symmetric (18). The remaining expression is exactly the decoded latent bracket. This proves Proposition 1. Decoder curvature alone does not corrupt composition when both controls are pushed forward through the same map.

B.3 Proof of the extension-identifiability theorem

Dependence on transverse derivatives. The values $X|_M$ and $Y|_M$ tell us the arrows attached to points of M. A Lie bracket also uses derivatives of those arrows. The question is whether the values on M determine the derivatives needed at $h \in M$.

Let $M \subset \mathcal{H}$ be embedded and let $h \in M$. The bracket is

$$[X, Y]\,(h) = J_Y(h)X(h) - J_X(h)Y(h). \tag{29}$$

Tangent Case. If $X(h), Y(h) \in T_hM$, then there exist smooth curves γ_X, γ_Y in M passing through h with tangent vectors $X(h)$ and $Y(h)$ at $t = 0$, respectively, such that

$$\gamma_X(0) = \gamma_Y(0) = h, \qquad \gamma_X'(0) = X(h), \qquad \gamma_Y'(0) = Y(h).$$

Consider $X(\gamma_Y(t))$. By the chain rule,

$$\frac{d}{dt}X(\gamma_Y(t))\Big|_{t=0} = dX_{\gamma_Y(t)}\left[\frac{d}{dt}\gamma_Y(t)\right]\Big|_{t=0} \tag{30}$$

$$= dX_h(Y(h)) \tag{31}$$

$$= J_X(h)Y(h). \tag{32}$$

Similarly for $Y(\gamma_X(t))$, we get $\frac{d}{dt}Y(\gamma_X(t))\big|_{t=0} = J_Y(h)X(h)$

Because $\gamma_Y(t) \in M$, the derivative above depends only on $X|_M$. Similarly, the derivative of $Y(\gamma_X(t))$ depends only on $Y|_M$. Changing the fields away from M cannot change the bracket at h. When $X|_M$ and $Y|_M$ are tangent to M on a neighborhood of h, they define vector fields on M. Thus, their bracket is the intrinsic bracket on M, in particular, it is tangent to M.

Transverse Case. For the converse, assume first that $X(h) \notin T_hM$. Then there exists $\alpha \in N_h^*M$ such that $\alpha(X(h)) \neq 0$. Rescaling, assume $\alpha(X(h)) = 1$.

Since α annihilates T_hM, there exists a smooth local function f such that

$$f|_M = 0, \qquad df_h = \alpha,$$

and therefore $df_h[X(h)] = 1$.

For any $q \in T_h\mathcal{H}$, define $\widetilde{Y} = Y + fq$. Since $f|_M = 0$, we have $\widetilde{Y}|_M = Y|_M$. Also $f(h) = 0$, thus

$$\left[X, \widetilde{Y}\right](h) - [X, Y]\,(h) = J_{(fq)}(h)X(h) - J_X(h)(f(h)q) \tag{33}$$

$$= df_h[X(h)]q = q \tag{34}$$

Thus, the same restriction to M allows an arbitrary change in the bracket at h. Figure 4 illustrates this construction. If instead $Y(h) \notin T_h M$, choose f satisfying $f|_M = 0$ and $df_h[Y(h)] = -1$, and define $\widetilde{X} = X + fq$. Then $\widetilde{X}|_M = X|_M$, and

$$\left[\widetilde{X}, Y\right](h) - [X, Y](h) = -df_h[Y(h)]q = q \tag{35}$$

A bump function localizes either construction. This proves the non-identifiability and arbitrary bracket variation asserted in Theorem 1.

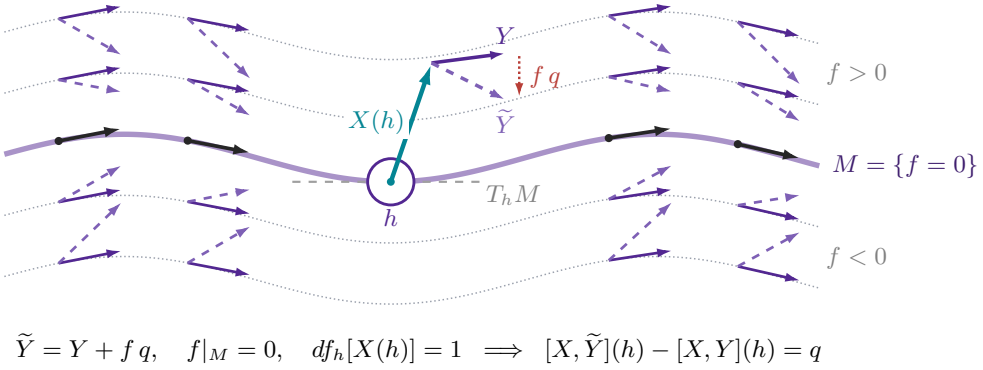


Figure 4: **Values on M do not determine the bracket.** The extensions Y (solid purple) and $\widetilde{Y} = Y + fq$ (dashed violet) agree on $M = \{f = 0\}$ (black arrows) and separate in proportion to f away from it (dotted level sets). Because $X(h)$ is transverse, differentiating along X detects this separation, so the two brackets at h differ by the arbitrary vector q (Theorem 1). If $X(h)$ and $Y(h)$ were tangent, both derivatives would be taken along M, where the extensions coincide.

B.4 Chain-rule derivation of the exact re-encoding defect

Executed hidden-space field. At a hidden state h, let $z = E(h)$. The latent control v specifies the velocity $v(z)$, which is mapped to hidden space by the decoder Jacobian:

$$W_v(h) = J_D(z)v(z), \qquad z = E(h). \tag{36}$$

The field W_v depends on h through two factors: the decoder Jacobian $J_D(z)$ and the latent field value $v(z)$. Both dependencies pass through $z = E(h)$.

Perturbing the hidden state. Let $u \in \mathbb{R}^d$ be a small hidden-space direction. The encoder converts it into the latent perturbation $J_E(h)u$. Applied to $w(z)$, the change in the decoder Jacobian is

$$\big(dJ_D(z)(J_E(h)u)\big)\, w(z) = H_D(z)[J_E(h)u, w(z)]. \tag{37}$$

The latent field changes by

$$J_w(z)J_E(h)u. \tag{38}$$

To make the two contributions explicit, consider one hidden component ℓ,

$$W_w^\ell(h) = \sum_{a=1}^{k} \frac{\partial D_\ell}{\partial z_a}(z) w_a(z). \tag{39}$$

Differentiating this expression in direction u gives two sums:

$$\begin{aligned} dW_w^\ell(h)[u] = &\sum_{a,b=1}^{k} \frac{\partial^2 D_\ell}{\partial z_a \partial z_b}(z)(J_E u)_b w_a(z) \\ &+ \sum_{a,b=1}^{k} \frac{\partial D_\ell}{\partial z_a}(z) \frac{\partial w_a}{\partial z_b}(z)(J_E u)_b. \end{aligned} \tag{40}$$

The first line differentiates the decoder Jacobian. The second differentiates the latent control. Collecting these component equations into vector notation gives

$$J_{W_w}(h)u = H_D(z)(J_E u, w(z)) + J_D J_w(z) J_E u. \tag{41}$$

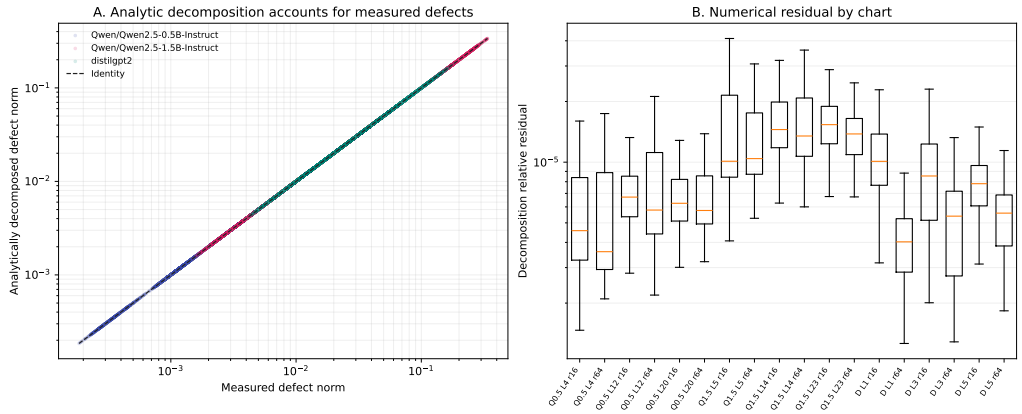


Figure 5: The analytic decomposition reproduces the measured local defect. (a) The sum of the curvature and control-variation terms lies on the identity line across three model families. (b) The remaining relative discrepancy is near numerical precision across charts, layers, and latent ranks.

Differentiating along the first control. The bracket needs the derivative of W_w in the direction generated by W_v. Set $u = W_v(h) = J_D v(z)$. Then $J_E u = J_E J_D v = C_h v$. Substituting this identity into Equation (41) gives

$$J_{W_w}(h)W_v(h) = H_D(z)(C_h v, w) + J_D J_w(z) C_h v. \tag{42}$$

Forming the bracket. The v-then-w contribution is $J_{W_w} W_v$, and the reverse contribution is $J_{W_v} W_w$. Subtracting the second from the first gives Equation (7):

$$[W_v, W_w](h) = H_D(C_h v, w) - H_D(C_h w, v) \tag{43}$$

$$+ J_D(J_w C_h v - J_v C_h w). \tag{44}$$

Separating the latent bracket from the re-encoding defect. Substitute $C_h = I + A_h$. Expanding each occurrence of C_h creates one group with I and one group with A_h. The terms containing I_k give

$$H_D[v, w] - H_D[w, v] + J_D(J_w v - J_v w) \tag{45}$$

$$= J_D[v, w]. \tag{46}$$

By Hessian symmetry, the two curvature terms cancel, leaving $J_D[v, w]$. This is the bracket prescribed by the latent controls. The A_h group is

$$\Delta_h(v, w) = H_D[A_h v, w] - H_D[A_h w, v] + J_D(J_w A_h v - J_v A_h w). \tag{47}$$

This is the re-encoding defect. The first line contains decoder curvature and describes how the cycle error changes the decoded hidden direction. The second line contains J_v or J_w and describes how the cycle error changes the latent command after re-encoding.

Numerical verification. Equation (8) is not only a formal identity: its separately computed terms account for the defect observed by automatic differentiation. The residual in Figure 5 measures numerical agreement, not a fitted prediction error.

B.5 Reconstruction counterexample and non-commuting limits

Construction. The decoder maps the latent plane (x, y) to the curved surface (x, y, xy), and the encoder adds a small oscillation to the first coordinate:

$$D(x, y) = (x, y, xy), \qquad E_\delta(h) = \left(h_1 + \frac{\delta}{2} \sin \frac{2h_1}{\delta}, h_2\right). \tag{48}$$

Because $|\sin t| \leq 1$, the encoder changes the first coordinate by at most $\delta/2$. For every compact subset $\mathcal{K}$ in the decoded manifold, the decoder derivative is bounded on a corresponding neighborhood

of the latent coordinates of $\mathcal{K}$. The mean-value theorem therefore gives a constant $C_{\mathcal{K}}$, independent of sufficiently small δ, such that

$$\sup_{h\in\mathcal{K}} \|D(E_\delta(h)) - h\| \le C_{\mathcal{K}}\delta. \tag{49}$$

Hence reconstruction error converges uniformly to zero on $\mathcal{K}$ as $\delta \to 0$.

Derivative behavior. The derivative of the perturbation is not small:

$$\frac{d}{dh_1}\left[\frac{\delta}{2}\sin\left(\frac{2h_1}{\delta}\right)\right] = \cos\left(\frac{2h_1}{\delta}\right). \tag{50}$$

The amplitude shrinks with δ, but its derivative retains the unit amplitude. Thus, value reconstruction can become arbitrarily accurate while the differential decode–encode cycle remains $O(1)$. Direct multiplication of the encoder and decoder Jacobians gives

$$A_h = J_{E_\delta}(h)J_D(E_\delta(h)) - I = \operatorname{diag}\left(\cos\frac{2h_1}{\delta}, 0\right) \tag{51}$$

for the two latent directions used below. With $v = e_1$, $w = e_2$, and $H_D(e_1, e_2) = e_3$, Equation (9) yields

$$\Delta_h(e_1, e_2) = \left(0, 0, \cos\frac{2h_1}{\delta}\right). \tag{52}$$

Its norm is one whenever $h_1 \in (\pi\delta/2)\mathbb{Z}$. The spacing between such anchors is $\pi\delta/2$, so these unit differential defects occur on an increasingly dense set as $\delta \downarrow 0$, even though reconstruction vanishes. Thus value reconstruction alone does not control the differential cycle without an additional uniform smoothness assumption.

Order of limits. Equation (10) does not imply a finite gap that stays large as $\delta \to 0$. If δ is fixed and the intervention duration ϵ tends to zero, Taylor expansion gives

$$\frac{\delta\epsilon}{2}\sin\left(\frac{2\epsilon}{\delta}\right) = \epsilon^2 + O(\epsilon^4/\delta^2), \tag{53}$$

so the coefficient of ϵ^2 is one. If ϵ is fixed and $\delta \to 0$, the absolute gap is at most $\delta\epsilon/2$ and therefore goes to zero. The two limits therefore answer different questions. The first extracts the local bracket coefficient for one fixed system. The second changes the encoder–decoder system while holding the flow duration fixed. For example, the second derivative of the oscillatory perturbation is

$$-\frac{2}{\delta}\sin\left(\frac{2h_1}{\delta}\right),$$

whose magnitude is $O(\delta^{-1})$. More generally, higher derivatives grow as inverse powers of δ, so the Taylor remainder is not uniform in δ.

Finite-step interpretation. For a fixed smooth system and sufficiently small flow duration ϵ, the ordered endpoint difference is governed at leading order by the Lie bracket:

$$h_{AB} - h_{BA} = \epsilon^2 [W_v, W_w](h_0) + O(\epsilon^3) \tag{54}$$

Thus quadratic scaling provides expected bridge from the differential calculation to finite executions. This local statement is not a uniform lower bound on finite-step separation across the family E_δ.

B.6 Proof of measurement separation

The reconstruction-versus-interaction separation is the preceding counterexample. The other two constructions are explicit.

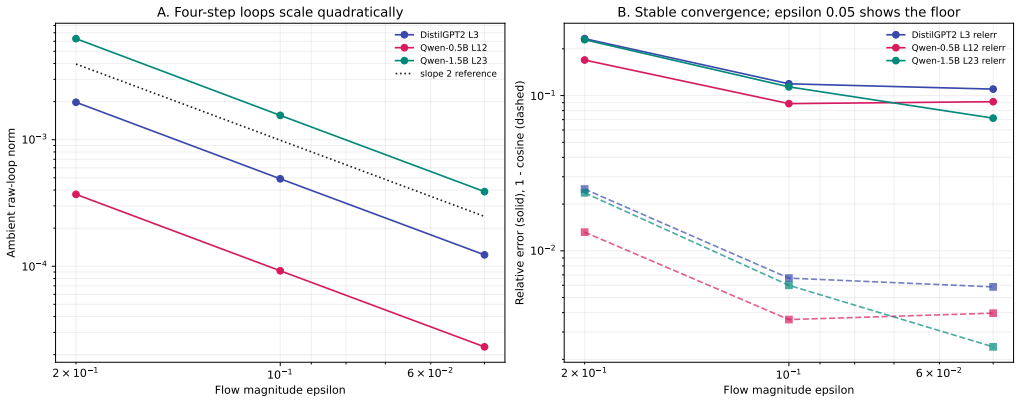


Figure 6: Finite-flow loops recover the local bracket in the smooth regime. (a) Four-step loop magnitudes follow the predicted quadratic scaling in the flow duration ϵ. (b) Directional error decreases and then reaches a numerical floor; solid curves show direction error and dashed curves show relative norm error.

Zero interaction does not imply correct target delivery. Perfect interaction with poor singleton delivery follows from $D = I$, $E_c(h) = ch$, $h_0 = 0$, and coordinate-wise feedback controls.

The executed field is $W_i(h) = (\tau_i - ch_i)e_i$. Different coordinates do not interact, so $[W_1, W_2] = [v_1, v_2] = 0$. Under singleton execution,

$$\dot{h}_i = \tau_i - ch_i, \qquad h_i(0) = 0,$$

whose solution is

$$h_i(t) = \frac{\tau_i}{c}(1 - e^{-ct}).$$

Since $E_i(h) = ch_i$, this gives

$$E_i(\Phi^1_{W_i}(0)) = \tau_i(1 - e^{-c}). \tag{55}$$

and therefore

$$T_i = \left| \frac{1 - e^{-c}}{1 - e^{-1}} - 1 \right| \longrightarrow 1 \quad \text{as } c \downarrow 0. \tag{56}$$

The two coordinates remain independent, so interaction is exactly zero. Yet the encoder gain c slows the executed feedback field. As $c \downarrow 0$, the delivered change approaches zero and the normalized target error approaches one. Thus perfect interaction does not imply useful singleton control.

Correct target delivery does not imply zero interaction. Perfect singleton delivery with nonzero interaction follows from $D = I$, $h_0 = 0$, and

$$E(h) = \begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix} h. \tag{57}$$

Under singleton i, the other hidden coordinate remains fixed, and the named coordinate satisfies $\dot{h}_i = \tau_i - h_i$. Thus $T_1 = T_2 = 0$. At the anchor, direct differentiation gives

$$[W_1, W_2](0) = a\tau_2 e_1 - b\tau_1 e_2, \tag{58}$$

whereas the declared coordinate controls commute. Choosing $a = 1/\tau_2$ and $b = 0$ gives interaction defect one. Each control reaches its own target because the other hidden coordinate is fixed during singleton execution. When both controls are available, the off-diagonal encoder coefficient a makes the second coordinate affect the first command. This produces nonzero interaction. All maps are linear or affine, so the construction does not use a support transition or another nonsmooth effect.

C OVERCOMPLETE DICTIONARIES AND SUPPORT COMPOSABILITY

For a piecewise-linear sparse autoencoder, use column-vector convention $W_{\mathrm{enc}} \in \mathbb{R}^{m\times d}$ and $W_{\mathrm{dec}} \in \mathbb{R}^{d\times m}$, so feature i has encoder row $W_{\mathrm{enc},i:}$ and decoder column $W_{\mathrm{dec},:i}$. Inside a cell with active support S, biases disappear from the differential and the active coordinate cycle is

$$C_S = W_{\mathrm{enc},S:}W_{\mathrm{dec},:S}, \qquad A_S = C_S - I_{|S|}. \tag{59}$$

This is a $|S| \times |S|$ matrix; the full $m \times m$ product need never be materialized. Since $m > d$ implies $\mathrm{rank}(W_{\mathrm{enc}}W_{\mathrm{dec}}) \le d < m$, global identity is impossible. Sequential control only needs a restricted property on supports that actually occur.

The rank argument is simple. The hidden state has only d dimensions, so a decode–encode round trip can preserve at most d independent latent directions. An overcomplete SAE has $m > d$ latent coordinates. It cannot preserve all m directions at once. The active support is much smaller, however, so the restricted matrix C_S can still be close to identity.

Let $G_S = W_{\mathrm{dec},:S}^{\top}W_{\mathrm{dec},:S}$ be the pullback of the hidden Euclidean metric to active decoder coordinates, assumed positive definite on the support, and write $\|B\|_{G_S} = \left\|G_S^{1/2}BG_S^{-1/2}\right\|_2$. The matrix G_S measures latent changes by the hidden-space distance they produce. This matters because rescaling one latent coordinate and inversely rescaling its decoder vector should not change the geometry. The G_S-weighted norm is invariant to that matched coordinate change, while the ordinary spectral norm need not be.

Definition 1 ($(\varepsilon, s_{\max})$-composability)**.** *A dictionary is $(\varepsilon, s_{\max})$-composable on a support family $\mathcal{S}$ if*

$$\sup_{S\in\mathcal{S},\,|S|\le s_{\max}} \left\|C_S - I_{|S|}\right\|_{G_S} \le \varepsilon. \tag{60}$$

The property is uniform when $\mathcal{S}$ contains all subsets of size at most $s_{\max}$, and empirical when $\mathcal{S}$ is a held-out collection of realized supports. The quantity $\|C_S - I\|_2$ is the release-gauge measure; it is useful within one normalized release but is not invariant under general latent changes of coordinates.

In plain terms, $(\varepsilon, s_{\max})$-composability says that every tested active set of at most $s_{\max}$ features approximately survives the infinitesimal decode–encode round trip. A smaller ε means less gain error, cross-feature leakage, and rotation inside those active coordinates. It is an infinitesimal support property; finite target delivery and generated behavior are measured separately.

Proposition 4 (Exact fixed-support repair)**.** *Let $D_S = W_{\mathrm{dec},:S} \in \mathbb{R}^{d\times|S|}$ have full column rank, so in particular $|S| \le d$. For decoder bias b, define the fixed-support encoder $E_S^{\mathrm{pinv}}(h) = D_S^{\dagger}(h - b)$. Its differential gives*

$$C_S^{\mathrm{pinv}} = D_S^{\dagger}D_S = I_{|S|}, \qquad \left\|A_S^{\mathrm{pinv}}\right\|_{G_S} = 0. \tag{61}$$

Thus a local left inverse of the decoder can be imposed without changing it. Moreover, $b + D_S E_S^{\mathrm{pinv}}(h)$ is the least-squares reconstruction of h in $b + \mathrm{span}(D_S)$. Writing $y = h - b$, for every code z on that support,

$$\left\|y - D_S D_S^{\dagger}y\right\|_2 \le \|y - D_S z\|_2 . \tag{62}$$

It therefore cannot have larger reconstruction error than the released code on the same fixed support.

On a fixed active support, the decoder pseudoinverse makes the infinitesimal cycle exact and gives the best Euclidean reconstruction in that decoder span. It is a local reference construction, not a global sparse encoder.

C.1 PROOF OF THE FIXED-SUPPORT REPAIR

Full column rank means that the Gram matrix $D_S^{\top}D_S$ is invertible. The Moore–Penrose pseudoinverse is therefore

$$D_S^{\dagger} = (D_S^{\top}D_S)^{-1}D_S^{\top}. \tag{63}$$

Multiplying by D_S on the right gives

$$D_S^\dagger D_S = (D_S^\top D_S)^{-1} D_S^\top D_S \tag{64}$$
$$= I_{|S|}. \tag{65}$$

Thus an infinitesimal active-coordinate change is decoded by D_S and recovered exactly by $D_S^\dagger$. This proves the cycle statement.

For the reconstruction statement, define $P_S = D_S D_S^\dagger$. This matrix is the orthogonal projector onto the column space of D_S. Let $y = h - b$. Every reconstruction using the fixed support has the form $D_S z$ and lies in this column space. Orthogonal projection is the closest point in that space, so

$$\|y - P_S y\|_2 \le \|y - D_S z\|_2 \tag{66}$$

for every support-restricted code z. The repair therefore cannot increase Euclidean reconstruction error while the support remains fixed. If the support changes, both $D_S^\dagger$ and the local coordinate map change; the proposition does not claim a global repair.

Diagonal gain error and off-diagonal leakage are reported separately. A common scalar gain error cancels from Equation (9) by Hessian symmetry, whereas heterogeneous gains and off-diagonal leakage can both contribute. In particular, if A_S is diagonal and $v = e_i, w = e_j$, then

$$\Delta_h(e_i, e_j) = (A_{ii} - A_{jj}) H_D[e_i, e_j]. \tag{67}$$

Thus $\max_{i,j \in S} |A_{ii} - A_{jj}|$ is the operative release-gauge gain statistic for constant controls under a curved decoder. Low reconstruction does not imply Equation (60). Exact identity $C_S = I$ preserves the declared bracket for every local smooth control pair that remains in that support cell; $C_S \neq I$ removes that universal guarantee without proving a failure for every pair. The reported percentiles describe sampled supports and certify neither unobserved supports nor finite trajectories.

D PUBLIC SAE EVIDENCE AND CONTROLLED-EXPERIMENT DETAILS

This appendix has three parts. Appendix D.1 measures the differential cycle in public SAEs and shows that it does not track reconstruction. Appendix D.2 trains SAEs with a cycle penalty and asks when this helps sequential control. Appendix D.3 gives the details of the order-selection experiments summarized in Section 5.

D.1 DIFFERENTIAL CYCLES IN PUBLIC SAES

Question and data. We ask whether better reconstruction also brings the local decode–encode cycle closer to identity, as repeated re-encoding requires. We use the `gpt2-small-res-jb` and feature-splitting SAE releases with GPT-2 Small activations on WikiText-103. The layer sweep contains 13 residual-stream SAEs. The controlled width sweep holds the site fixed at `blocks.8.hook_resid_pre` and varies width from 768 to 98,304 in eight steps. Each SAE contributes 10,000 realized supports after excluding special tokens, for 21 dictionaries and 210,000 supports in total, and Equation (59) is computed exactly on each support. FVU uses the same activations.

Metrics. The releases normalize decoder feature vectors (maximum deviation from unit norm 0.00221). This fixes one coordinate scaling, which we call the *release gauge*. Matrix norms in this gauge, such as $\|A_S\|_2$, can change under other latent reparameterizations. Our primary metric removes this dependence by weighting with the decoder-induced metric $G_S = W_{\mathrm{dec},:S}^\top W_{\mathrm{dec},:S}$:

$$\|A_S\|_{G_S} := \left\| G_S^{1/2} A_S G_S^{-1/2} \right\|_2, \tag{68}$$

which is invariant under $C_S \mapsto R_S C_S R_S^{-1}$ with the corresponding metric transformation. We call it the *covariant* cycle error. We also report the similarity-invariant spectral radius $\rho(A_S)$, per-entry RMS leakage, and analyses at fixed support size. With eight widths, the smallest attainable exact permutation p-value is $p_{\min} = 4.96 \times 10^{-5}$.

Table 2: Differential cycles in public SAEs. Every dictionary uses 10,000 held-out realized supports. Ranges are across dictionaries, not confidence intervals.

Sweep	SAEs	Median $\lvert S\rvert$	FVU	$\lVert A_S\rVert_2$	$\lVert A_S\rVert_{G_S}$	$\rho(A_S)$
Layer / hook	13	11–76	.031–.236	1.75–2.29	3.42–30.79	1.26–1.74
Width, block 8	8	38–47	.172–.400	.98–2.26	1.13–3.91	.90–1.64

Main result: reconstruction and cycle quality diverge. No evaluated dictionary is near identity on typical supports (Table 2). Across the width sweep, FVU falls from 0.400 to 0.172 while median covariant cycle error rises monotonically from 1.13 to 3.91 (Spearman -1.000, $p = p_{\min}$). The trend survives control for support size: the partial Spearman given median support size is -1.000 ($p = p_{\min}$), and within common support bins the association is -0.929 for $|S| \le 16$ and -1.000 in each of the bins 17–32, 33–48, 49–64, and ≥ 65, with at least 105 observations per width. Two secondary metrics agree, but less cleanly. Release-gauge $\|A_S\|_2$ has Spearman -0.929 ($p = .00223$): it rises through width 49,152 and then falls from 2.256 to 1.972 at 98,304, an inversion in the last of seven adjacent transitions. The spectral radius $\rho(A_S)$ has Spearman -0.857 ($p = 0.0107$), but this association does not survive the condition matching below. Tables 10 and 11 give per-width values and support-bin counts.

What the result does and does not show. This is a necessary-condition analysis. For a linear decoder and constant feature translations, $H_D = 0$ and Equation (9) vanishes even when $C_S \neq I$; cycle error matters only through decoder curvature or through the J_v, J_w terms of state-dependent controls in Equation (8). The result therefore does not claim that every evaluated SAE already exhibits a false behavioral interaction. It shows that good reconstruction does not supply the local right-inverse property needed to certify repeated re-encoding.

Is decoder conditioning responsible? Wider releases have worse-conditioned active decoders, and ill-conditioning inflates the covariant metric, so we test whether conditioning explains the trend. All metric calculations use float64 eigendecomposition with no pseudoinverse or eigenvalue floor, and reject supports with $\lambda_{\min}(G_S) \le 10^{-12}\lambda_{\max}(G_S)$; no support was rejected. Three checks show that the covariant trend survives. First, restricting every width to $\kappa(G_S) \le 100$ retains at least 7,606 supports per width and leaves Spearman -1.000, but this threshold removes little of the wide-release distribution. Second, a stricter match intersects the per-width 5th–95th percentile condition-number ranges. On the common interval $\kappa(G_S) \in [31.09, 78.46]$, which retains 48–6,657 supports per width, the FVU association is -0.929 for $\|A_S\|_{G_S}$ ($p = .00223$), -0.762 for $\|A_S\|_2$ ($p = .0368$), and -0.333 for $\rho(A_S)$ ($p = .428$). Because every within-width error–conditioning slope is positive, this match compares the narrow release's extreme high-condition tail against typical wide-release supports; it is therefore conservative for the observed association, although its 48-support endpoint is imprecise. Third, an all-support regression avoids that selection. Fitting $\log \|A_S\|_{G_S} = a_m + b_m \log \kappa(G_S)$ to all 10,000 supports at each width and evaluating at the common $\kappa_0 = 49.39$ gives fitted errors $1.91, 2.37, 2.65, 2.87, 3.26, 3.49, 3.50, 3.29$ (Spearman -0.929, $p = .00223$); evaluating at either end of the common interval gives Spearman -0.857 and -0.929.

How much conditioning explains. Conditioning amplifies the trend but does not produce it. The fitted slopes range from 0.346 to 0.453 and themselves associate with FVU (Spearman -0.738, $p = .0458$), so conditioning acts as a width-dependent amplifier rather than a common gain. Between the narrowest and widest releases, the raw error ratio is 3.46 (log contrast 1.242); adjusting for conditioning reduces it to 1.72 (log contrast 0.543). Conditioning therefore accounts for 56.2% of this descriptive log effect, and 43.8% remains. The regression R^2 declines from .930 to .526, but the absolute residual log-IQR does not grow (.286–.377; width association Spearman .071, $p = .882$), so the decline reflects a smaller explained fraction, not larger scatter. Medians in the fixed bands $\kappa \in [45, 55]$, $[40, 60]$, and $[35, 65]$, which need no functional form, retain associations $-.833$, $-.857$, and $-.905$; because the narrowest release contributes only 3, 9, and 26 supports to these bands, they check direction rather than estimate the endpoint. Finer pooled-$\log \kappa$ strata have uneven width overlap and mixed associations, and the spectral-radius trend is not independently supported after strict matching. Appendix N reports all counts and diagnostics.

An irregularity at the widest releases. The fitted covariant error falls from 3.505 to 3.290 at the widest release, release-gauge $\|A_S\|_2$ inverts at the same point, and diagonal gain error γ reverses nearby. Release metadata supplies no explanation for this shared irregularity, and we report it without assigning a cause.

Gain, leakage, and a random null. Could the trend reflect only gain, with the cycle uniformly shrinking or stretching features? On a seeded subset of 512 supports per width, the best common-gain correction $\min_{\alpha\in\mathbb{R}} \|A_S - \alpha I\|_{G_S}$ still rises monotonically from 0.996 to 3.913 as FVU improves (Spearman -1.000, $p = p_{\min}$). The best per-feature diagonal correction $\min_{\Lambda\in\mathrm{Diag}} \|A_S - \Lambda\|_{G_S}$, computed in the release gauge, also rises monotonically, from 0.815 to 2.810 (same Spearman and p). Neither shared nor per-feature gain absorbs the result: cross-feature leakage worsens with width. Heterogeneous gain adds a second mechanism under a curved decoder; the median active diagonal-gain spread rises from 0.178 to 0.573 (Spearman -0.929, $p = .00223$), with an anomalous 0.949 at width 49,152. These diagonal quantities describe the chosen release and are not invariant certificates. For scale, random unit-vector dictionaries matched on $(m, d, |S|)$, with $W_{\mathrm{enc}} = W_{\mathrm{dec}}^{\top}$, have gain-centered medians 0.406–0.465, so the learned-to-random ratio grows from 2.45 to 8.45.

Repairing the cycle, and its cost. The cycle can be closed exactly on a fixed support, but at a semantic cost. Applying Proposition 4 to the same 4,096 supports (512 per width) gives median decoder-metric error 9.5×10^{-15}–2.4×10^{-14} (maximum 7.7×10^{-10}) and reduces fixed-support reconstruction SSE by 13.2–29.0%. However, 1.5–15.4% of coefficients become negative, affecting 42.8–99.0% of supports, and median code norms grow by 1.11–$1.21\times$. Nonnegative least squares preserves signs and reduces SSE by 13.1–28.0% with a 1.09–$1.20\times$ norm increase, but its median free-coordinate fraction falls from 1.000 to 0.837, so it restores the cycle only on that free KKT set.

D.2 Cycle-regularized training

Training with a cycle penalty. The public results identify cycle error as a quality axis that reconstruction does not capture. We therefore add the decoder-gauge-aware penalty $\mathcal{L}_{\mathrm{cycle}} = B^{-1} \sum_b \|W_{\mathrm{enc},S_b:} W_{\mathrm{dec},:S_b} - I\|_F^2 / |S_b|$ to a linear TopK SAE on GPT-2 block-8 residual-output activations (`blocks.8.hook_resid_post`). We train four coefficients with three independent seeds and 300M tokens per run, and evaluate on a common set of 10,000 supports. Relative to each seed's unregularized checkpoint, $\lambda = 1000$ lowers median covariant cycle error by 27.8–28.4% for a 2.0–3.5% FVU increase, and $\lambda = 3000$ lowers it by 40.7–41.9% for a 5.1–7.0% increase; both reduce the cycle by at least 20% while increasing FVU by less than 10% in every seed. At $\lambda = 10{,}000$, the reduction reaches 55.6–56.9%, but FVU rises by 14.6–16.1%. Median support stays 64, numerical diagnostics remain stable, and the unobserved-feature fraction falls rather than rises (Figure 10). Cycle quality can therefore be improved at a measurable reconstruction cost.

Gemma counterpart. The matched Gemma comparison uses block 12 residual output (`blocks.12.hook_resid_post`), with ordinary and $\lambda = 1000$ cycle-regularized SAEs trained at width 16,384, TopK 64, 300M tokens, and seeds 17, 29, and 42. Its selected cohort contains 5, 5, and 6 baseline pair identities across the three seeds, evaluated over 512 contexts per seed with RK4-4.

Summary across cohorts. Table 3 places the prediction, repair, and selection analyses side by side. Its main lesson is that the repair effect depends strongly on which feature pairs are tested.

Large gains on the selected cohort, small gains on a broad cohort. On the selected cohort (Appendix F), cycle regularization reduces the mean hidden interaction residual by 85.04%, 63.97%, and 81.63% for GPT-2 seeds 17, 29, and 42. On the broad, identity-disjoint regularizer cohort (Appendix G), the same checkpoints reduce it by only 3.26%, 35.00%, and 14.24%, a gap of 81.79, 28.97, and 67.40 percentage points. Both comparisons use the same checkpoints, cycle coefficient, intervention strength, error definitions, and RK4-16 implementation. What differs is the cohort. The GPT-2 selected cohort contains only 3–4 distinct top-ranked pair identities per seed, repeated over 512 contexts; the broad cohort contains 197–219 lower-ranked identities, disjoint from the preceding prediction and repair experiments, over 1,024 contexts, and requires simultaneous four-arm matching. Support events are also rarer in the selected cohort (ordinary/cycle rates 49.0–59.6%/30.1–37.7%) than in the broad cohort (73.9–77.4%/78.7–81.2%). The effect difference therefore tracks the cohort

Analysis	Cohort and exclusion	Hidden/logit change	Delivery and result
Prediction	Ranks 3–4; new prompts/hash split; M held out from fitting	**Mean:** G -7–$73/-.4$–36%; M 33–57/23–54%. **p95:** G -24–$53/3$–37%; M 47–$53/23$–55%.	Target/displacement .990–1.036 / .956–1.011. Hidden Spearman $.850/.864$; G seed 17 worsens.
Ordered residuals	242/251 G/M pairs; excludes prior evaluated identities	**Mean:** G $35/28\%$; M $24/23\%$. **p95:** G $31/38\%$; M $30/25\%$.	Target/displacement .976–1.085 / .944–.997. Pair-aggregated direction Spearman $.983/.987$; row-level overall, fixed-support, and support-change strata span .977–.983.
Cycle repair	Top-2 matched; selected cohort (Appendices D and F; G,M)	**Mean:** G 64–$85/53$–74%; M 71–$82/66$–81%. **p95:** G 64–$83/46$–75%; M 42–$84/65$–81%.	Target/displacement .973–.990 / .943–1.096. Large reductions occur on the selected cohort.
Repair selection	Ranks 5–6; excludes earlier evaluated identities; new prompts	**Mean:** G 39–$48/32$–41%; M 29–$50/32$–50%. **p95:** G 54–$61/47$–63%; M 33–$48/35$–56%.	Target/displacement 1.030–1.101 / .949–.990. Selection adds only .68–2.29 pp over always-cycle.
Exact behavior	New exact JSON/arithmetic task; joint execution not run	No joint reduction is defined because neither singleton improves its exact metric.	The edits are delivered, but exact lift is zero.
Regularizers	197–219 G pairs/seed; excludes prior evaluated identities	**Mean:** 0.28–34.55 / 0.06–29.40%. **p95:** -26.46–31.93 / -9.97–40.91%.	Target/displacement 1.081–1.221 / .926–.990. Repair varies by seed; decoder conditioning reaches lower residuals at weaker delivery.

Table 3: **Prediction, repair, and specificity across disjoint cohorts.** H/L denotes hidden and centered-logit interaction residual; positive percentages are cycle-versus-ordinary reductions and negative values are degradations. Target and displacement are cycle/ordinary ratios. G and M denote GPT-2 and Gemma. Means and p95 values use uniform context weighting; the repair-selection experiment uses a separate pair-level estimand chosen before evaluation. Exact per-seed values and identity manifests are in Appendices F, J, and G. Displayed ranges summarize model–seed cells; they are not confidence intervals.

and operating regime, not a different trained treatment or evaluation code. Because rank, prompts, pair identities, matching, baseline defect, and support events co-vary, a crossed design would be needed to attribute the difference to one property; we report the two cohorts separately.

The broad-cohort result in detail. In the primary RK4-8 evaluation of the broad cohort, seed 17 improves mean hidden/logit interaction residual by only $.28\%/.06\%$ while hidden p95 worsens by 26.46%; seed 42 improves the means by $12.97\%/2.37\%$ while hidden/logit p95 worsen by $1.19\%/9.97\%$. Across RK4-4, 8, and 16, the seed-17 cycle mean moves from 6.18% worse to 3.26% better than ordinary. Against generic regularizers with coefficients selected by the same development budget, cycle training reaches the lowest cycle error, but control delivery differs. Encoder-Jacobian regularization is worse than ordinary in seeds 17 and 29 at all resolutions. Decoder conditioning has lower raw logit residual than cycle training in every seed and lower mean hidden error than ordinary in every cell, but it moves 12–14% less than cycle training ($.813$–$.847\times$ ordinary displacement). These operating points therefore trade interaction residual against delivered strength. Coefficients were selected by cycle error after 20M tokens, so the comparison describes these selected implementations (Appendix G).

Robustness of the selected-cohort gain. The selected-cohort gain is stable across numerical resolutions. Recomputing both arms at RK4-4, 8, and 16 on a saved audit sample preserves lower mean cycle-arm interaction residual in all 18 cells (six model–seed comparisons at three resolutions, not 18 independent replications). Hidden reductions remain 63.95–85.12%, hidden-order and centered-logit reductions keep their sign, and joint target-achievement ratios remain .985–1.001. Each cell contains 256 observations: 128 ordinary and 128 cycle rows drawn from the same selected pair identities (3–4

per seed for GPT-2 and 5–6 for Gemma). "128" therefore counts contexts per arm, not distinct feature pairs. Exact-flow convergence is analyzed separately.

Predicting which pairs benefit. We next ask whether local geometry identifies the pairs that benefit from cycle training. Development data define one rule: predict strong repair when the cycle/ordinary local directional-magnitude ratio is at most .5. Decisions are frozen on a new prompt seed before joint execution, in calibration runs that exclude all records from the preceding prediction and repair analyses, the generic-regularizer comparison, and the cross-objective evaluation identities. On 1,269 GPT-2 and 979 Gemma pair identities, the rule selects 367 and 211. Cycle training helps 85.8% and 90.0% of the selected pairs, much more often than in the unselected pairs, and predicted and observed ratios have Spearman $.652/.692$. Mean pair-level reductions are 40.6% and 25.1%, below the 50% threshold chosen on development data, and Gemma seed 17 is negative because a few large reversals dominate. Retrospectively, medians are $70.2\%/68.8\%$ and ratio-of-aggregate-means reductions are $57.3\%/55.7\%$. Local geometry therefore stratifies where repair is likely to help, while the large reversals mark the remaining tail risk.

Transfer across training objectives. We apply the same ordered-residual scalar, without refitting, to 698 new pair identities, five strengths, and all four training objectives. For the ordinary SAE, directional/order Spearman remains $.886/.901$, with median vector cosine .999. For the three regularized objectives, directional Spearman falls to .550–.570 and order cosine to .796–.828. In every arm, the RK4-8 versus RK4-16 p95 relative discrepancy exceeds .05. Pair geometry therefore remains informative for the ordinary chart, but its calibration and ranking become objective-dependent under regularized training. A selective stable-subset analysis raises regularized-arm order correlations to .874–.914, whereas directional correlations remain .625–.727 versus .926 for the ordinary SAE. The full-data comparison mixes objective dependence with numerical error (Appendix H).

D.3 Order selection: details for Section 5

Prediction summary. Section 5 evaluates the geometry score before either joint trajectory is observed. The fixed-cell ordered-residual calculation reaches Spearman $.983/.987$ and median vector cosine 1.000 across RK4-4/8/16, as expected analytically, and the small fitted comparison (Table 7) improves over reconstruction, conditioning, and aggregate-cycle baselines. The remaining question is how reducing the predicted residual changes direct joint control; Table 1 shows both outcomes side by side.

Decision rule. For each fresh prompt and previously excluded feature pair, the geometry rule chooses AB or BA before either joint endpoint is available. Decisions are hashed, and both orders are evaluated only afterward. We also compare a two-singleton heuristic, which needs two singleton probes, and a hindsight oracle.

Reduction and uncertainty. All six seed-level reductions are positive (21.0–25.8% for GPT-2 and 29.7–47.8% for Gemma-2-2B). The paired row-bootstrap 95% intervals for the absolute advantage are $[3.74, 4.59] \times 10^{-5}$ and $[3.72, 4.27] \times 10^{-5}$. Because prompts repeat across seeds and feature pairs repeat across contexts, rows are not independent; prompt-cluster and pair-with-seed-cluster intervals also remain strictly positive. Recomputing the same decisions at RK4-4/8/16/32, geometry reduces hidden interaction residual in all 24 comparisons. Five of six model–seed cells vary by less than two percentage points; GPT-2 seed 42 exceeds that range by .230 points at RK4-4 without reversing the benefit. Centered-logit error was not recomputed at the additional resolutions.

Cost and side effects. The rule needs one local geometry evaluation and no singleton or joint rollout. On one H200 at batch size 16, an isolated synchronized geometry call takes 22.8–24.0 ms per batch after one warmup, or 1.4–1.5 ms amortized per decision. This excludes model loading, anchor encoding, joint execution, and suffix evaluation, and is neither single-example latency nor an end-to-end speedup. Mean displacement and support-event rate are nearly unchanged, and each final target changes by less than one percentage point. Retention of the first-applied intervention, however, falls from .945 to .909 for GPT-2, with seed-level paired losses relative to the default order of $-.007$, $-.034$, and $-.067$ (row-weighted) and $-.004$, $-.027$, and $-.053$ (equal-pair-weighted). Because changing order can change which feature comes first, this is an execution-level measure rather than

an identity-matched change for one named target; the identity-specific final A and B achievements remain within their specified tolerances. Order selection therefore improves composition fidelity under the specified final-target constraints, not control uniformly. For this controller, the geometry rule coincides with the ordinary order-aware Taylor rule: the contribution is the SAE-computable score and its prospective use, while the underlying expansion is classical.

Direct error. Using the same order choices, an H200 replay exactly reproduces all saved source quantities. Unlike the singleton-subtracted interaction residual, direct error (distance to the shared joint reference) changes little: hidden reductions range from approximately -0.0% to $.9\%$, and centered-logit reductions from $-.3\%$ to 1.2%. Most pair-cluster intervals include zero. Named A/B target errors are small and mixed, while the worst named-target error worsens by .2–1.5% in all six cells. Geometry therefore selects lower non-additive interaction error, not uniformly lower total endpoint error.

Sequential and joint controllers. We compare ordering with direct and conventional joint controllers on the same cohort. Directly constructing the residual-preserving reference minimizes reference distance by definition, but it worsens named A/B target errors by 33.1–60.2% relative to geometry ordering, so the analytic reference is not a drop-in target-delivery controller. More importantly, a unit-time flow of $W_A + W_B$ and the symmetric split $A/2, B, A/2$ each use the same 128 equivalent field evaluations as one ordered execution, and both have slightly lower means than geometry ordering in every model–seed cell: direct hidden error falls by .15–$.73\%$, centered-logit error by .09–$.82\%$, and worst named-target error by .33–2.50%. Collateral latent change is essentially unchanged. Pair-cluster bootstrap intervals sharpen this boundary. Simultaneous feedback has lower worst-target error than geometry ordering in all six model–seed cells, with every interval excluding zero; symmetric splitting does so in five of six cells. Hidden and logit improvements are clear for GPT-2 seeds 29 and 42, while most Gemma direct-error intervals include zero. No equivalence claim follows, because we did not define a practical equivalence margin. An oracle that evaluates both orders finds only 2.4–4.5% additional direct-error headroom, costs twice as much, and worsens worst-target error by 1.2–2.6%. Geometry is therefore useful for choosing among externally required sequential executions, but this comparison does not show that sequential feedback should replace simple joint control. For reference, the controller comparison uses RK4-16 endpoints: two controls, 16 substeps per control, and four stages per substep give 128 field evaluations. The 64-to-128 count in Appendix J instead uses RK4-8, where the joint word costs 64 evaluations and two singleton calibration flows add another 64.

Support transitions. We replay every order-selection row and record all RK4 stages of singleton A, singleton B, AB, and BA. Prediction remains strong when unrelated TopK coordinates enter or leave support: order accuracy is 91.5–97.1%, and geometry reduces hidden interaction by 21.4–35.6% across the six cells. When either controlled feature changes active-set status, accuracy falls to 50.0–78.4% and two cells reverse the mean benefit; this stratum is sparse for Gemma. A separate 36-row numerical comparison sharpens this distinction. Between RK4-16 and RK4-64, rows without transitions have zero support-path mismatches and zero lower-order flips, whereas rows with controlled-feature transitions have 75.0% path mismatch and 8.3% order flips. Neither singleton batching nor promoted float64 removes this boundary.

Coordinate-convention correction. The first analytic check used the wrong coordinate convention. This paragraph documents the error, verifies the corrected formula, and explains why the main text reports the original prospective values. Trajectories and targets were executed in the norm-scaled coordinates returned by the SAE, but the initial check combined those increments with checkpoint-coordinate encoder and decoder matrices. Because these checkpoints rescale activations by decoder norm, the executed maps are $d_i = W_{\mathrm{dec},:i} / \|W_{\mathrm{dec},:i}\|_2$ and $g_i = \|W_{\mathrm{dec},:i}\|_2 W_{\mathrm{enc},i:}^\top$. Writing $s_i = \|W_{\mathrm{dec},:i}\|_2$, the diagonal coefficient $g_i^\top d_i = W_{\mathrm{enc},i:} W_{\mathrm{dec},:i}$ is unchanged, but the ordered terms for the saved increments satisfy $K_{AB}^{\mathrm{eff}} = K_{AB}^{\mathrm{ckpt}} / s_i$ and $K_{BA}^{\mathrm{eff}} = K_{BA}^{\mathrm{ckpt}} / s_j$.

Inside one fixed linear support cell, let $c_i = g_i^\top d_i$ and $f_i(t) = (1 - e^{-c_i t})/c_i$, with $f_i(t) = t$ at $c_i = 0$. The feedback field then gives exactly $R_{AB} = f_i(1) f_j(1) K_{AB}$ and $R_{BA} = f_i(1) f_j(1) K_{BA}$. To verify this under the corrected convention, we trace 96 rows (16 per model–seed cell): we test the affine map at the anchor, evaluate both controlled coordinates at 257 points along each executed

Table 4: **Controlled-coordinate affine reconciliation.** "Valid/invalid" tests the affine map at the anchor. Error columns report median/p95 deployed disagreement. Near-zero means either exact residual is at most 10^{-6} of anchor norm and is excluded only from the residual-relative column. Support changes are observations at 257 sampled states per segment, not a certificate against between-sample crossings.

Cell	Valid/ invalid	Affine max	Anchor norm. median/p95	Residual % median/p95	Near zero	Sampled changes
GPT-2, 17	16/0	2.91×10^{-5}	$4.31/6.61\times10^{-7}$	$0.87/3.60$	1	0
GPT-2, 29	15/1	1.34×10^{-5}	$4.03/4.41\times10^{-7}$	$0.77/3.30$	0	0
GPT-2, 42	13/3	9.54×10^{-6}	$4.05/4.86\times10^{-7}$	$0.58/8.44$	0	0
Gemma-2-2B, 17	16/0	2.34×10^{-5}	$3.82/4.33\times10^{-7}$	$0.63/6.62$	1	0
Gemma-2-2B, 29	16/0	3.05×10^{-5}	$4.16/4.72\times10^{-7}$	$0.63/11.78$	0	0
Gemma-2-2B, 42	16/0	2.72×10^{-5}	$4.03/5.74\times10^{-7}$	$1.10/10.74$	1	0

segment, and, in a separate replay, replace their encoder evaluations by explicit affine maps. Ninety-two rows satisfy the anchor premise; four GPT-2 rows contain an inactive selected coordinate and are retained as specification failures. Across the 92 valid rows, no controlled or unrelated coordinate enters or leaves support at the sampled states, the maximum controlled-coordinate affine error is 3.1×10^{-5}, and the median residual disagreement normalized by anchor norm is 3.8–4.3×10^{-7} for deployed execution and 3.1–3.5×10^{-7} for affine replay (Table 4). The exact fixed-cell formula is therefore internally consistent under the implemented convention; it does not apply to rows that fail the anchor-support premise.

Effect of the correction on reported values. The convention affected the analytic predictor, not the saved executions. For Table 7, we repeat the fit with the effective map while preserving the original split, ridge penalty, outcomes, and unaffected features; only pair conditioning and pair-defect norm change. Spearman/R^2 changes from $.850/.843$ to $.971/.953$ on 15 GPT-2 pairs and from $.864/.725$ to $.949/.884$ on 34 Gemma pairs. On the 242/251 held-out pairs, directional Spearman rises from $.983/.987$ to $.999/.999$ and order accuracy from $94.2\%/93.2\%$ to $97.1\%/98.4\%$, changing $2.9\%/5.2\%$ of pair decisions. On the original order-selection cohort, corrected choices agree with the original choices on $95.3\%/95.9\%$ of rows and yield $25.1\%/38.5\%$ reduction rather than $24.3\%/38.2\%$. The four anchor-invalid rows occur in the primary GPT-2 evaluation; removing them changes the corrected reduction from 25.1% to 25.3%, so they do not drive the conclusion. Because these corrections were computed after the joint outcomes were available, we treat them as a retrospective sensitivity analysis, not a second prospective result, and Table 7 retains the original checkpoint-coordinate values. Support-cycle matrices reported elsewhere retain their explicitly labeled checkpoint/release gauge; coordinate-covariant cycle statistics and all saved execution outcomes are unchanged by the diagonal rescaling.

Why interaction residual and direct error diverge. The same replay explains the gap between the two outcomes. The median ratio $\|R_o\|/\|e_o\|$ is .040–.042 for GPT-2 and .060–.073 for Gemma-2-2B, although p95 reaches .188–.402: interaction is a small part of endpoint error. Among rows where geometry lowers $\|R_o\|$, direct error still worsens 37.8–57.4% of the time, and 67.8–88.6% of those reversals have the sign expected from losing a favorable cancellation with singleton error. Figure 9 shows how a smaller ordered residual can still increase direct error. Finally, an additive reference grid leaves the hidden interaction residual invariant to 2.4×10^{-10} but frequently changes the sign of direct-error controller comparisons. Interaction residual, direct error, and target delivery therefore answer different questions.

Relation to broader control-quality measures. We next test whether the interaction residual explains broader measures of control quality. A five-fold analysis holds out complete feature-pair identities and predicts worst named-target error or off-pair latent change from singleton errors, singleton target delivery, joint displacement, support events, and order. Adding the interaction residual changes out-of-pair R^2 by $-.013$ to $+.028$ for worst-target error and by $-.005$ to $+.002$ for collateral change; Spearman changes are similarly negligible and inconsistent. Because the same outcomes appear in the ordering and joint-controller comparisons, and because joint displacement

Table 5: **Output-level consequence.** Equal-feature-pair-weighted reduction in centered-logit interaction residual relative to the default order. Intervals resample feature-pair means within each seed.

Model	Seed	Reduction	Pair-cluster 95% CI
GPT-2	17	21.2%	$[1.38, 2.09] \times 10^{-2}$
GPT-2	29	17.4%	$[0.92, 2.33] \times 10^{-2}$
GPT-2	42	17.6%	$[1.02, 1.78] \times 10^{-2}$
Gemma-2-2B	17	33.2%	$[1.52, 2.29] \times 10^{-2}$
Gemma-2-2B	29	31.6%	$[1.72, 2.57] \times 10^{-2}$
Gemma-2-2B	42	39.0%	$[2.05, 3.27] \times 10^{-2}$

and support-event count are observed only after execution, this result is descriptive rather than an independent prediction test. It shows that the interaction residual adds little conditional explanatory value; it does not test pre-execution screening of broader failures.

Output-level interaction residual. We pass the order-selected hidden endpoints through the remaining transformer layers, keeping prompts, feature pairs, checkpoints, increments, and hashed choices unchanged. Recomputed hidden endpoints match the saved values exactly (maximum discrepancy 0.0). We then evaluate the centered-logit interaction residual, that is, Equation (12) applied to centered next-token logits. The shared residual-preserving joint reference is order-invariant because its two latent increments add commutatively. Table 5 reports the resulting reductions.

All six feature-pair-cluster intervals exclude zero, and row-weighted analyses agree in sign. The singleton ordering heuristic is worse than the default order in five cells and improves one Gemma cell by only 1.8%. Final latent-target changes remain within the specified tolerances throughout. The output-level analysis uses seven suffix evaluations per row, but the prospective selector requires none. This is a follow-up to the order-selection experiment, not an independent replication, and it establishes an output-level residual consequence rather than better generated behavior or task success.

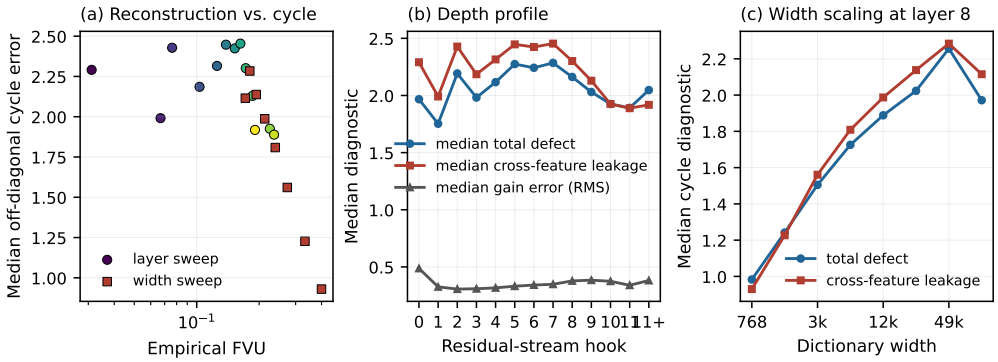


Figure 7: **Overview of the public-SAE result.** Left: reconstruction error and differential-cycle error do not move together. Middle: the mismatch appears across layers and hook locations. Right: at block 8 (`hook_resid_pre`), wider dictionaries reconstruct better while their local decode–encode cycles move farther from identity. This overview complements the gauge-robust and conditioning-controlled analysis in Figure 8.

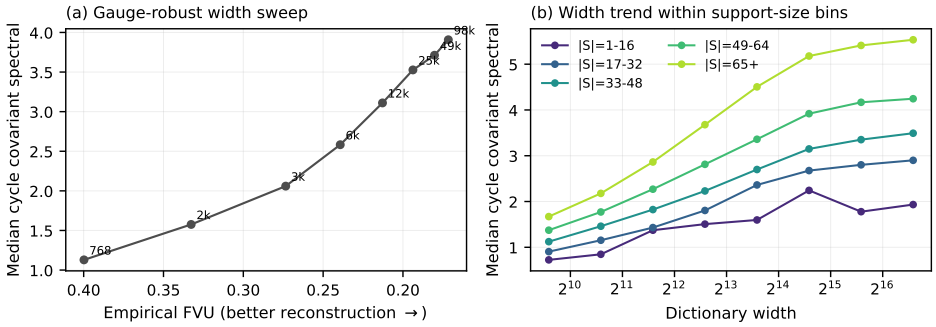


Figure 8: Better reconstruction does not imply better intrinsic composability. (a) Across the controlled block-8 residual-input width sweep, FVU falls while decoder-metric cycle error rises monotonically (Spearman -1.000, exact permutation floor $p_{\min} = 4.96 \times 10^{-5}$ for $n = 8$). (b) The trend persists when every dictionary is conditioned on the same realized-support-size bin; all bins have Spearman ≤ -0.929.

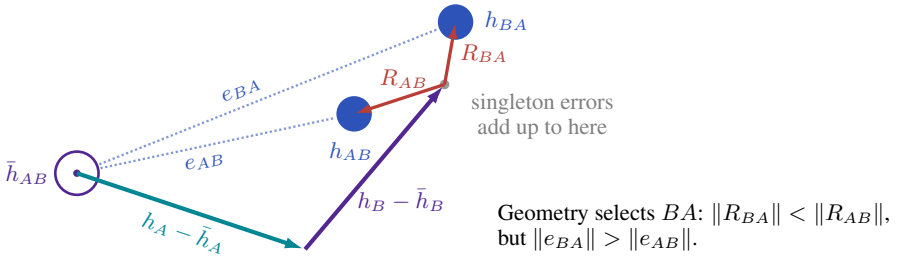


Figure 9: **A smaller interaction residual can still increase direct error.** Each direct error decomposes as $e_o = (h_A - \bar{h}_A) + (h_B - \bar{h}_B) + R_o$ (Equation (12)). Here R_{AB} points back toward the joint reference and cancels part of the singleton errors, whereas the smaller R_{BA} points away from it. The lower-interaction order therefore ends farther from $\bar{h}_{AB}$ (dotted). Residuals are enlarged for visibility; the median ratio $\|R_o\|/\|e_o\|$ is .04–.07.

E IMPLICATIONS FOR CONTROLLER EVALUATION

The extension rule, local cycle error, commuting-null bracket, and finite-flow loop describe different properties of an intervention. A nonidentity cycle removes the universal guarantee that ambient re-encoding preserves the declared latent algebra. Residual preservation supplies a reference against which this departure can be measured. Pseudoinverse re-encoding closes the cycle but breaks sign semantics on up to 99.0% of supports, while NNLS is defined only on part of the state space. Appendix D and Table 14 report these repair costs together with the cycle statistics.

Cycle error alone also does not determine downstream composition. The regularization experiments therefore compare reconstruction, cycle geometry, target delivery, and sequential error at the same operating points. Nearest-centroid, TopK, and MoE switches require event semantics (Appendix T). Horizon nulls and a zeroth-order Manifold Steering projection perturbation reveal implementation sensitivity rather than a behavioral advantage (Appendices K and O).

F CYCLE-REGULARIZED SEQUENTIAL CONTROL

This appendix tests whether training with the cycle penalty of Appendix D reduces the execution error of a sequential feedback controller. The test uses a small selected cohort of feature pairs; Appendix G repeats it on a broad cohort.

Setup. We use GPT-2 Small at block 8 residual output (`blocks.8.hook_resid_post`) and compare SAEs trained with $\lambda \in \{0, 1000, 3000\}$, where $\lambda = 0$ is the ordinary SAE, for training seeds 17, 29, and 42. Evaluation uses 512 held-out prompts balanced across factual, reasoning, instruction, and dialogue templates.

Matching features across SAEs. Feature indices are not comparable across independently trained SAEs. We therefore treat a baseline feature as present in a regularized SAE only when its decoder column and a regularized decoder column are mutual nearest neighbors with cosine at least .70, in both regularized dictionaries. Each prompt uses two such matched features that are active at the anchor. Median matched cosine is .855–.883. Across the 512 prompts, this yields only 3–4 distinct feature pairs per seed.

Controller and error measure. Each matched feature receives a feedback control,

$$v_i(z) = (\tau_i - z_i)e_i, \qquad v_j(z) = (\tau_j - z_j)e_j, \tag{69}$$

scaled so that each control's initial decoded displacement is $.02\,\|h\|$. The ambient controller re-encodes after every RK4 substep (RK4-4) and runs in both orders. The residual-preserving reference integrates Equation (69) exactly while keeping the anchor residual fixed. We report the *ambient-to-reference error*: the gap between each ambient endpoint and its residual-preserving reference, in hidden space and in centered next-token logits.

Success criteria, fixed before evaluation. A coefficient counts as a consistent improvement only if, in every seed, (i) mean hidden and centered-logit errors fall by at least 20%, with paired-bootstrap intervals excluding zero; (ii) target achievement stays at least 90% relative and 50% absolute; (iii) at least 80% of prompts retain matched features; and (iv) a negative control passes: constant translations, which commute exactly under a linear decoder, have p99 order gap below 10^{-6}.

Results. $\lambda = 1000$ meets every criterion. For seeds 17, 29, and 42, hidden error falls by 80.84%, 71.87%, and 77.85%, and logit error by 69.17%, 65.70%, and 75.13%. Target retention is 100.37%, 99.89%, and 99.27%. Pooled paired log-ratio intervals are $[-1.601, -1.538]$ for hidden error and $[-1.242, -1.183]$ for logit error, where negative values mean lower error. The constant-translation control reaches at most 1.32×10^{-7}, and no top-1 token changes in the primary condition. $\lambda = 3000$ fails: its medians improve, but seed 29 develops a severe upper tail in which mean hidden and logit errors rise by $14.17\times$ and $5.08\times$.

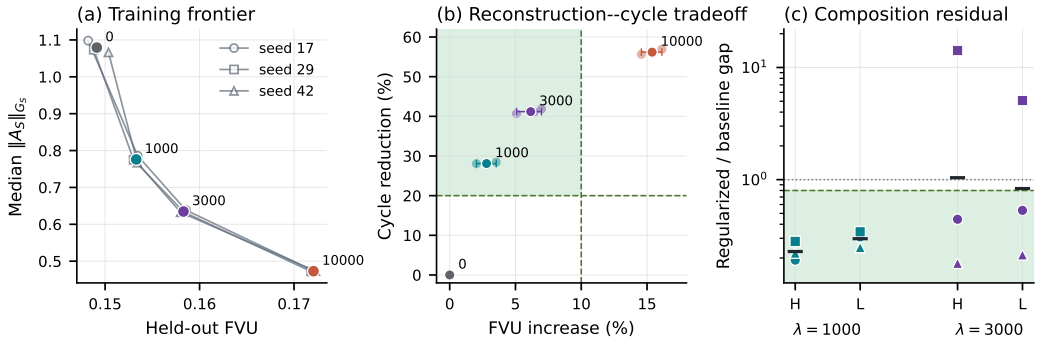


Figure 10: Training for composability changes a selected sequential operation. (a) All three 300M-token seeds trace the reconstruction–cycle frontier; colored points are across-seed means. (b) $\lambda = 1000$ and 3000 occupy the low-cycle, reconstruction-matched region in every seed. (c) Regularized-to-baseline ambient-reference gap on the selected feature cohort (H: hidden; L: centered logits). $\lambda = 1000$ reduces both errors in every seed; $\lambda = 3000$ has a severe upper tail in seed 29. Table 3 shows why this selected-cohort result is not a typical effect-size estimate.

Relation to other reported values. The reductions above measure the full endpoint gap, which includes errors already present when each control runs alone. Subtracting those singleton errors isolates the interaction residual (Equation (12)). On the same RK4-4 executions, this gives 85.04%, 63.97%, and 81.64% (Table 3); the RK4-16 replay in Appendix D gives 85.04%, 63.97%, and 81.63%. The two sets of values differ because they measure different errors, not because of different checkpoints, strengths, or integration. Because this cohort contains only a few top-ranked pairs, these reductions are not typical effect sizes; the broad-cohort comparison in Appendix G finds much smaller gains.

G Cycle regularization and generic alternatives

This appendix asks whether the benefit of cycle training comes from the cycle penalty itself or would come from any regularizer that changes encoder or decoder geometry. It also measures cycle training on a broad cohort of feature pairs rather than the small selected cohort of Appendix F.

Two generic alternatives. We compare the support-cycle loss with two generic penalties on the same GPT-2 block-8 residual-output TopK architecture:

$$\mathcal{L}_{\text{enc}}(S) = \frac{\|E_S\|_F^2}{|S|}, \qquad \mathcal{L}_{\text{cond}}(S) = \frac{\left\|\bar{D}_S^\top \bar{D}_S - I\right\|_F^2}{|S|}. \tag{70}$$

The first penalizes the size of E_S, the effective encoder Jacobian after decoder-norm rescaling. The second penalizes non-orthogonality among the active decoder columns $\bar{D}_S \in \mathbb{R}^{d\times|S|}$, each normalized to unit length. Both are invariant to the reciprocal encoder–decoder rescaling used by the implementation.

Coefficient selection. Each objective receives exactly four 20M-token development runs. These select $\lambda_{\text{enc}} = 100$ and $\lambda_{\text{cond}} = 1000$, after which we train six new 300M-token runs (two objectives, seeds 17, 29, and 42). The saved runner configurations match the ordinary and cycle checkpoints of the cycle-regularization experiment in architecture, data, optimizer, token budget, and decoder normalization. Selection minimizes held-out cycle error after 20M tokens, not downstream error after 300M, so the comparison concerns these selected implementations, not the best possible member of each objective family.

Cycle error falls, but interaction residual barely moves. The final test uses 1,024 fresh contexts per seed and excludes every GPT-2 feature-pair identity used in the earlier prediction and repair experiments. Cycle training achieves the lowest cycle error: its median is .767–.786, versus .927–.932 for encoder regularization and .972–1.078 for decoder conditioning. Relative to the ordinary

Table 6: Mean hidden interaction-residual reduction relative to ordinary training at RK4-4/8/16. Positive values mean lower error than the ordinary SAE.

Seed	Cycle	Encoder Jacobian	Decoder conditioning
17	$-6.18/0.28/3.26\%$	$-18.28/-15.45/-14.24\%$	$25.41/24.29/25.24\%$
29	$32.07/34.55/35.00\%$	$-4.92/-4.67/-4.64\%$	$30.02/29.62/29.94\%$
42	$11.81/12.97/14.24\%$	$-2.95/1.17/2.03\%$	$3.47/2.94/5.33\%$

SAE, however, it reduces mean hidden/logit interaction residual by only $.28\%/.06\%$ in seed 17, $34.55\%/29.40\%$ in seed 29, and $12.97\%/2.37\%$ in seed 42, and the tails can get worse: hidden p95 rises by 26.46% in seed 17 and 1.19% in seed 42, and seed-42 logit p95 rises by 9.97%. The large gains on the selected cohort are therefore not typical pair-level gains.

Sensitivity to numerical resolution. We recompute the same 1,024 contexts per seed at RK4-4 and RK4-16, around the original RK4-8 evaluation, with identical pair identities at every resolution. Table 6 reports the results. The 36 cells (four SAEs, including the ordinary reference, at three seeds and three resolutions) contain only three independent training seeds, not 36 independent replications. Cycle training's seed-17 result changes sign across resolutions. Encoder-Jacobian regularization is worse than ordinary in seeds 17 and 29 at every resolution. Decoder conditioning stays better than ordinary on average, but its interventions move the state only $.813$–$.847\times$ as far as the ordinary SAE's.

The arms deliver different control strengths. The regularizers change not only interaction residual but also how strongly a control moves the state, so the comparison is not matched in delivered strength. Against the encoder-Jacobian SAE, cycle training has lower interaction residual, with equal-pair-weighted cycle-minus-encoder intervals of $[-4.42, -1.64] \times 10^{-5}$ in hidden space and $[-.02491, -.01133]$ in centered logits. But encoder controls move the state 1.13–$1.20\times$ as far as cycle controls while reaching only $.73$–$.77\times$ their target achievement. Decoder conditioning is closely matched to cycle training in reconstruction (FVU-cost differences below $.5$ percentage points), yet its controls move the state only $.856$–$.881\times$ as far and reach $.893$–$.922\times$ the target achievement. Its raw logit interaction residual is lower than cycle training's in every seed (cycle-minus-decoder interval $[.00427, .01445]$), while the hidden interval $[-1.82, 18.97] \times 10^{-6}$ includes zero. Decoder conditioning therefore trades weaker delivery for lower logit residual.

Conclusion. All arms use 300M tokens and one B200 per checkpoint, with recorded training times of 1:06:49–1:09:01 and median support 64, so compute is matched. The comparison nonetheless couples each objective with a different delivered strength and reports performance at these operating points, not at matched delivery; we apply no retrospective strength correction and keep the comparison on GPT-2. We therefore treat cycle training as the constructive repair and the two generic regularizers as mechanistic comparisons, and we identify delivered strength as an additional axis for comparing regularizers.

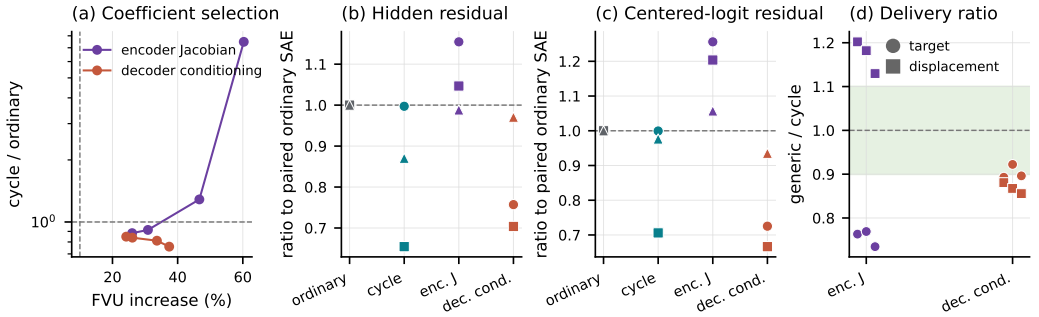


Figure 11: Direct cycle and generic regularizers make different tradeoffs between geometry and delivered strength. (a) Development curves; rings mark the coefficients selected before final training. (b–c) Hidden and centered-logit interaction residuals relative to each paired ordinary SAE (circles, squares, and triangles are seeds 17, 29, and 42). (d) Singleton target and displacement ratios for each generic arm relative to cycle training; the shaded band marks the intended comparability interval. The generic regularizers deliver different control strengths.

H Prediction across strengths and training objectives

This appendix asks whether the geometric predictor transfers, without refitting, to new feature pairs, new intervention strengths, and SAEs trained with different objectives.

Setup. We reuse the two scalar predictors of Appendix J unchanged: the directional score, which predicts interaction magnitude, and the antisymmetric score, which predicts which order has lower interaction. We exclude every pair identity used in the main prediction and repair analyses and evaluate 698 new identities at strengths .005, .01, .02, .04, and .08, in four SAEs: ordinary, cycle-regularized, encoder-Jacobian-regularized, and decoder-conditioned. No outcome is used for refitting.

Prediction quality. Prediction transfers well for the ordinary SAE but degrades for the three regularized ones. On the interpolation strengths, directional/order Spearman is $.886/.901$ for the ordinary SAE and $.550$–$.570/.599$–$.709$ for the regularized SAEs. In every SAE, the predicted and observed residual vectors for each order point in nearly the same direction (median cosine at least .999). The difference between the two orders is predicted less well: its cosine falls below .83 in all three regularized SAEs. Reconstruction and conditioning baselines have near-zero correlation with the outcome. The pair-cycle baseline reaches $.256$–$.632$ and, in the cycle-regularized SAE, beats the directional predictor.

Error sizes and numerical convergence. Mean relative error is very large ($72.8\times$–$212.4\times$) because some observed pair means are close to zero, which inflates relative error; median directional relative error is 42.7%–59.1%. Numerical convergence is also uneven. Between RK4-8 and RK4-16, the median relative change is at most 3×10^{-4}, but the 95th percentile is $.194$–$.805$ for the directional quantity and $.389$–$.840$ for the order quantity, above our $.05$ threshold in every SAE. Support events occur in 72.1%–78.6% of cases.

Is the discrepancy caused by near-zero residuals? Only partly. Most observations whose RK4-8/16 discrepancy exceeds $.05$ (60.6%–90.3%) fall in the smallest-residual quartile, where dividing by a near-zero value inflates relative error. But large discrepancies also occur in 25.8% and 34.5% of the largest-residual quartile for the cycle-regularized and decoder-conditioned SAEs, so near-zero normalization is not the whole explanation. Restricting to numerically stable pairs raises order Spearman to $.874$–$.920$ across SAEs, but directional Spearman stays at $.625$–$.727$ for the regularized SAEs, versus $.926$ for the ordinary SAE. This restricted comparison is descriptive and cannot tell us what the full data would show under exact integration.

Is it integration error? No. The original records did not save the refined vectors, so we recompute a fixed subset and keep full vectors at RK4-8, 16, and 32. If truncation error were the cause, the

RK4-16/32 discrepancy should be much smaller than the RK4-8/16 discrepancy. Instead, their median ratio is 1.28–1.42, and 14.5%–36.8% of rows still exceed the threshold. Rows whose support-event counts differ across resolutions (10.3%–14.8% of regularized rows) do not account for this: removing them does not restore convergence.

Is it batching or floating-point precision? Mostly not. On a separate 36-row diagnostic chosen before its outcomes were inspected, singleton and fixed-batch runs give identical vectors and identical logged supports. Recomputing in float64, including the stored weights, changes anchor-normalized results by about 10^{-7}. This is below 1.22% of the directional residual for the 33 rows with residuals above 10^{-5}, but it dominates the two residuals near 10^{-7}. An analytic smooth-field control recovers the expected fourth-order RK4 convergence in float64, so the integrator itself behaves correctly.

Conclusion. The predictor's calibration depends on the training objective. For rows with ordinary-sized residuals, results are stable under higher precision, and the two near-zero discrepancies are consistent with arithmetic-floor effects. The larger discrepancy tail in the regularized SAEs remains unexplained; unrecorded nonsmooth events, finite precision, and implementation sensitivity are the remaining candidates.

I SMOOTH MECHANISMS AND PUBLIC IMPLEMENTATIONS

This appendix checks the defect formula on smooth systems, where every assumption of Section 4 holds.

Setup. We test three kinds of system: analytic vector fields, smooth embeddings, and 18 learned nonlinear charts (combinations of model, layer, and latent rank) for DistilGPT2 and two Qwen2.5 models. We vary the chart seed, the denoising objective used to train the chart, and the smooth coordinate system. Uncertainty estimates resample field pairs.

Results. In every configuration, the defect computed by automatic differentiation matches Equation (8). The median cosine between the two and their norm correlation both round to 1.000, and even the worst p99 relative residual is below 4.0×10^{-5}. We also measure the order effect without derivatives, using finite commutator loops with step size ϵ. The loop gap grows as $\epsilon^{1.98}$–$\epsilon^{2.02}$, close to the ϵ^2 predicted by Equation (5), and its direction matches the differentiated bracket with cosine $.993$–$.996$. Under residual-preserving execution, loops of commuting controls return to their starting point to reported precision, as Proposition 1 predicts. Finally, a strongly curved decoder (Hessian norm 5.0) with an exact differential cycle produces no defect for commuting controls. Curvature alone therefore does not cause the effect; cycle error is required. These experiments test the geometric mechanism only and do not measure generated behavior.

Public implementations. Pinned revisions of three public code bases use three different extension rules (Appendix P). FLAS re-evaluates an ambient field at every step, Manifold Steering keeps the anchor residual fixed, and CircuitSteer adds fixed decoder vectors.

A curved-decoder check. Our main SAEs have linear decoders, so the curvature term in Equation (8) is zero for them. To test a curved decoder, we train an order-3 PolySAE on GPT-2 Small block 8 residual output (`blocks.8.hook_resid_post`). It uses a shared U, width 16,384, TopK 64, and seed 42, and is trained for 20,000,768 tokens ($\approx$ 20M; batch-aligned). Of twelve frozen feature pairs, we keep ten; the other two do not give a consistent valid target token on the development prompts. We use 64 development and 64 held-out prompts. The local expansion predicts the output-level order effect accurately: the predicted and measured centered-logit effects agree in direction with cosine > 0.99999, and at $\epsilon = 0.001$ the p90 relative prediction error is 7.0×10^{-4}. The effect itself, however, is tiny, with a median size of only 1.65×10^{-6}.

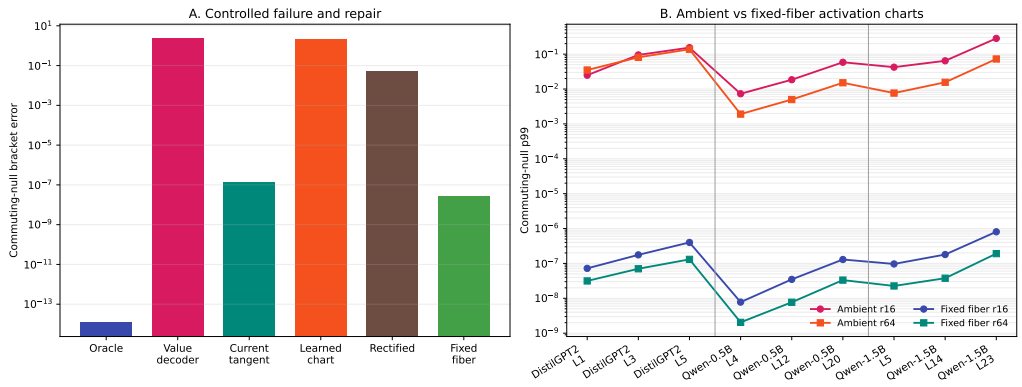


Figure 12: Controlled tests isolate the extension mechanism. (a) A curved decoder without cycle error, and residual-preserving execution, both pass the commuting-null test (zero order gap for commuting controls); value-only and learned ambient charts fail it, and restoring the current tangent repairs the controlled case. (b) The gap between ambient and residual-preserving execution persists across layers, model families, and two latent ranks.

J Predicting composition residuals on unseen feature pairs

This section separates three questions. First, how much error appears only when two controls are composed? Second, can anchor-local geometry predict that error before either joint order is executed? Third, can the prediction decide whether an unseen feature pair should use an ordinary or cycle-regularized SAE?

J.1 Removing errors already present in singleton controls

For order AB, the total endpoint error relative to the declared joint reference is $h_{AB} - \bar{h}_{AB}$. Part of this error may already occur when control A or control B is executed alone. Equation (12) removes those two singleton contributions:

$$R_{AB} = (h_{AB} - \bar{h}_{AB}) - (h_A - \bar{h}_A) - (h_B - \bar{h}_B), \tag{71}$$

$$R_{BA} = (h_{BA} - \bar{h}_{AB}) - (h_A - \bar{h}_A) - (h_B - \bar{h}_B). \tag{72}$$

The same reference $\bar{h}_{AB}$ appears in both orders because the declared residual-preserving joint update is additive and therefore order-invariant. The scalar outcome is

$$Q = \frac{\|R_{AB}\|_2 + \|R_{BA}\|_2}{2\,\|h\|_2}. \tag{73}$$

The numerator averages the non-additive error of the two orders, and the anchor norm makes values comparable across hidden states. A small Q means that the two singleton errors add approximately. It does not mean that either endpoint is close to its requested target. We apply the same subtraction to centered next-token logits when reporting output-space residuals.

J.2 Held-out fitted comparison

Before deriving the ordered predictor, we test whether pair-specific geometry adds information beyond common diagnostics. Each candidate predictor is fitted with ridge penalty 1 to standardized, log-transformed pair means. Development uses 17 GPT-2 feature pairs. Evaluation uses disjoint sets of 15 GPT-2 and 34 Gemma-2-2B pairs. Pair assignment and the ridge penalty are set before the test outcomes are evaluated.

Table 7: Held-out prediction from local pair geometry. "Basic" contains singleton and conditioning controls; aggregate cycle and pair geometry add support-wide and pair-specific defects.

Predictor	Added information	GPT-2: 15 pairs		Gemma: 34 pairs	
		Spearman	R^2	Spearman	R^2
Reconstruction	reconstruction only	.34	$-.03$	$-.23$	-1.23
Basic / conditioning	singleton and condition controls	.28	$-.28$	.11	$-.75$
Aggregate cycle	basic + support-cycle norm	.13	$-.50$	.33	$-.85$
Pair geometry	basic + pair-defect norm	**.85**	**.84**	**.86**	**.73**

Only the pair-geometry model has positive held-out R^2 for both model families. Reconstruction, singleton statistics, conditioning, and an aggregate cycle norm do not predict Q on both test sets. The evaluation sets are small, so this fitted comparison is supporting evidence rather than the main prediction result. Table 7 uses the original checkpoint-coordinate fields. Appendix D reports a retrospective refit with coordinate-consistent fields while keeping these original test results visible.

J.3 Why the two ordered terms appear

Let X denote control A and Y denote control B. Applying A and then B means evaluating Y after the state has already moved under X. That change of evaluation point creates an ordered cross term. For one smooth field X, the duration-t flow has the local expansion

$$\Phi_X^t(h) = h + tX(h) + \frac{t^2}{2} J_X(h)X(h) + O(t^3). \tag{74}$$

The linear term follows the field at the anchor. The quadratic term accounts for the fact that the field itself changes along the trajectory.

Now apply X first and Y second. Equation (74) moves the state to $h + tX + O(t^2)$. Taylor expansion of the second field at this new state gives

$$Y(h + tX + O(t^2)) = Y(h) + tJ_Y(h)X(h) + O(t^2). \tag{75}$$

Combining the two expansions yields

$$\Phi_Y^t \circ \Phi_X^t(h) = h + t(X+Y) + t^2\big(\tfrac{1}{2}J_X X + J_Y X + \tfrac{1}{2}J_Y Y\big) + O(t^3), \tag{76}$$

$$\Phi_X^t \circ \Phi_Y^t(h) = h + t(X+Y) + t^2\big(\tfrac{1}{2}J_Y Y + J_X Y + \tfrac{1}{2}J_X X\big) + O(t^3). \tag{77}$$

Both expansions contain the same first-order movement $t(X+Y)$. The self terms $J_X X/2$ and $J_Y Y/2$ also occur when the controls run alone, so the composition residual removes them. What remains at second order is $J_Y X$ for order AB and $J_X Y$ for order BA. Their difference, $J_Y X - J_X Y$, is the Lie bracket. The bracket predicts the difference between orders, but it cannot predict the two residual magnitudes separately. For that purpose we must retain both ordered terms.

The implemented controllers run for unit time. Their size is controlled by the declared target increments, not by taking t toward zero. The expansion above therefore identifies which vectors should appear, but its factor t^2 is not an exact unit-time magnitude formula. An exact unit-time multiplier is available inside a fixed affine SAE cell and is derived in the next subsection.

At the anchor, let $C_h = J_E(h)J_D(E(h))$ and $A_h = C_h - I$. For a general encoder–decoder controller, the ordered cross terms written in the stored checkpoint coordinates are

$$K_{AB}^{\mathrm{ckpt}} = H_D[C_h v, w] + J_D J_w C_h v, \tag{78}$$

$$K_{BA}^{\mathrm{ckpt}} = H_D[C_h w, v] + J_D J_v C_h w. \tag{79}$$

Subtracting the corresponding identity-cycle terms, obtained with $C_h = I$, gives the execution defects

$$K_{AB}^{\mathrm{def}} = H_D[A_h v, w] + J_D J_w A_h v, \tag{80}$$

$$K_{BA}^{\mathrm{def}} = H_D[A_h w, v] + J_D J_v A_h w. \tag{81}$$

The checkpoint-coordinate terms describe the complete ordered cross effect. The defect terms remove the cross effect that would remain if the differential cycle were exact, $C_h = I$. These quantities differ in general. In the evaluated setting, however, the decoder is linear, the controls act on distinct coordinates, and the identity-cycle cross terms vanish. The two definitions then coincide.

For a short duration ϵ, the separate vectors predict the mean residual magnitude through

$$\widehat{Q}_{\mathrm{dir}}^{(\epsilon)} = \epsilon^2 \frac{\left\|K_{AB}^{\mathrm{ckpt}}\right\|_2 + \left\|K_{BA}^{\mathrm{ckpt}}\right\|_2}{2\left\|h\right\|_2}, \tag{82}$$

whereas $K_{AB}^{\mathrm{ckpt}} - K_{BA}^{\mathrm{ckpt}}$ predicts the antisymmetric quantity $R_{AB} - R_{BA}$. We never use the bracket alone as the predictor of Q.

Inside a fixed linear SAE support cell, let d_i, d_j be decoder columns, g_i, g_j the corresponding encoder gradients, and δ_i, δ_j the declared target increments. The ordered terms reduce to

$$K_{AB}^{\mathrm{ckpt}} = K_{AB}^{\mathrm{def}} = -(g_j^\top d_i)\delta_i d_j = J_{W_B} W_A, \qquad K_{BA}^{\mathrm{ckpt}} = K_{BA}^{\mathrm{def}} = -(g_i^\top d_j)\delta_j d_i = J_{W_A} W_B. \tag{83}$$

The scalar $g_j^\top d_i$ measures how movement along decoder direction i changes encoded coordinate j. Thus K_{AB} points along d_j and is scaled by the cross-coordinate effect of A on B. The second equation has the opposite roles. These are ordinary ordered Taylor terms written in quantities available from the SAE. Below, K_{AB} and K_{BA} abbreviate these common fixed-cell quantities.

J.4 Exact fixed-cell multiplier

Inside one fixed TopK support cell, the encoder is affine. For controlled coordinate i,

$$E_i(h) = E_i(h_0) + g_i^\top (h - h_0). \tag{84}$$

The feedback controller moves along decoder direction d_i in proportion to the remaining distance from target τ_i:

$$W_i(h) = \big(\tau_i - E_i(h)\big) d_i. \tag{85}$$

Define that remaining target error as $q_i(t) = \tau_i - E_i(h(t))$. While control i is active, $\dot{h} = q_i(t) d_i$. Differentiating q_i along this trajectory gives

$$\begin{aligned} \dot{q}_i(t) &= -g_i^\top \dot{h}(t) && (86)\\ &= -g_i^\top d_i \, q_i(t) && (87)\\ &= -c_i q_i(t), \qquad c_i = g_i^\top d_i. && (88) \end{aligned}$$

The coefficient $c_i = g_i^\top d_i$ is the cycle's self-gain for coordinate i. The scalar differential equation has solution

$$q_i(t) = q_i(0) e^{-c_i t}. \tag{89}$$

so the target error decays exponentially when $c_i > 0$. Integrating the hidden velocity gives

$$\begin{aligned} h(t) - h(0) &= q_i(0) d_i \int_0^t e^{-c_i s}\, ds && (90)\\ &= q_i(0) f_i(t) d_i, && (91) \end{aligned}$$

where

$$f_i(t) = \begin{cases} (1 - e^{-c_i t})/c_i, & c_i \neq 0, \\ t, & c_i = 0. \end{cases} \tag{92}$$

The value $f_i(t) = t$ at $c_i = 0$ is the continuous limit of the first case. Therefore $f_i(t)$ converts the initial target error into the total distance moved along d_i.

Now consider composition. After control A acts on coordinate i, coordinate j changes by $g_j^\top (h_A - h_0)$. Control B therefore begins with a different target error than it had in its singleton execution. This is the cross effect captured by K_{AB}. Applying the same argument in reverse order and subtracting the singleton displacements gives

$$R_{AB} = f_i(1) f_j(1) K_{AB}, \qquad R_{BA} = f_i(1) f_j(1) K_{BA}. \tag{93}$$

Table 8: Spearman correlation with the separate observed AB and BA residual magnitudes at RK4-16. Scalar diagnostics assign the same score to both orders and therefore cannot identify which execution has lower error. Separate ordered terms retain that information.

Score computed before joint execution	GPT-2	Gemma
Reconstruction	.09	.03
Decoder overlap	.51	.53
Pair conditioning	.06	.00
Aggregate / pair cycle	−.03 / .31	.03 / .38
First-order singleton defect	.33	.32
Pair-defect norm	.75	.82
Standard-bracket norm	.83	.85
Direction-agnostic K mean	**.85**	**.87**
Separate K_{AB}, K_{BA}	**.98**	**.99**

Equation (93) is exact while both controlled coordinates remain in one affine support cell. In that regime, agreement with the formula validates implementation and coordinate conventions but is not an independent discovery. The empirical questions begin when the support changes, when numerical integration approximates the flow, and when the local quantities are used to rank previously unseen feature pairs.

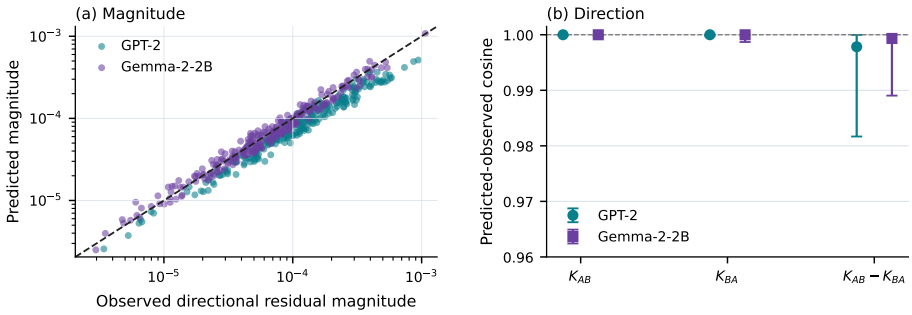


Figure 13: Held-out prediction on feature pairs. Each point compares a prediction made from local SAE geometry with the observed interaction residual after execution. The left panel tests magnitude; the right panel tests vector direction. Results are shown separately for GPT-2 and Gemma-2-2B. The original prospective predictor used checkpoint-coordinate fields while execution used decoder-norm-scaled target increments. The coordinate-consistent retrospective calculation is reported separately in Appendix D.

The implemented flows run for $t = 1$, while intervention strength is changed by varying the target increments δ_i and δ_j. These operations are not equivalent: changing a target alters W_A or W_B, but it does not scale their Jacobians in the same way as shortening time. We therefore use the Taylor derivation to identify the ordered vectors, not to assert an uncalibrated unit-time magnitude.

One positive scalar a is fitted on development pairs and then held constant:

$$\widehat{Q}^{(1)}_{\mathrm{dir}} = a \frac{\|K_{AB}\|_2 + \|K_{BA}\|_2}{2\,\|h\|_2}. \tag{94}$$

The scalar calibrates overall magnitude. Because it is positive and shared by both orders, it cannot change their ranking. Test-pair predictions use only the anchor-local encoder and decoder geometry, declared target increments, and this development scalar; they do not use either joint endpoint.

Two negative controls test whether a common intervention scale could explain the result. First, replacing C_h by the identity sets $A_h = 0$. In the evaluated distinct-coordinate setting, both ordered defect vectors then vanish. There is no nonzero score to rank, so correlation and order choice are undefined. This is the expected exact-cycle null.

Table 9: Selection on previously unused feature-pair identities. "Selected" is the rate at which the policy chooses the cycle-regularized SAE. Intervals use 10,000 bootstrap resamples of pair identities; positive differences favor the adaptive policy over always using the regularized SAE.

Model	Pairs	Selected	Δ_{select}	95% CI
GPT-2	242	89.3%	1.10×10^{-6}	$[-0.25, 2.70]\times10^{-6}$
Gemma-2-2B	251	79.7%	5.14×10^{-6}	$[1.90, 9.83]\times10^{-6}$

Second, we cyclically reassign predictions to the wrong feature pairs within each model and training seed. This preserves the prediction distribution and all global scales while removing the correspondence between a pair and its geometry. At RK4-8, Spearman falls from .983 to $-.005$ across 242 GPT-2 pairs and from .987 to .155 across 251 Gemma pairs. Accuracy for choosing the order with the smaller observed residual falls from .934 to .529 on GPT-2 and from .912 to .490 on Gemma. Pair identity, rather than a shared strength scale, carries the ordering information.

Numerical robustness is checked in 18 cells: two model families, three training seeds, and RK4 with 4, 8, or 16 substeps. Recomputing every residual vector and normalization reproduces stored values within 2.5×10^{-11}. Between RK4-8 and RK4-16, median relative differences are 0.04–0.21%; one GPT-2 cell has p95 difference 4.75%.

The paired ordinary-versus-cycle-regularized comparison has the same sign at all three resolutions. Across the six model–seed combinations, evaluated at three resolutions, cycle training reduces hidden interaction residual by 63.95–85.12%, while hidden-order and centered-logit reductions remain nearly unchanged. These are six replications checked at three numerical resolutions, not 18 independent replications. The comparison establishes stability of the method difference under these discretizations; it does not prove that every endpoint equals the exact continuous flow.

A one-sided strength control keeps one ordinary-SAE intervention at full strength and weakens the other by only 3–5%. This adjustment changes the regularization benefit by at most 0.11 percentage points. It addresses a small delivery imbalance but is not an exact strength match.

J.5 Choosing whether to use cycle regularization

The next comparison asks whether pair geometry can guide a concrete deployment choice. For each unseen feature pair, a policy chooses either the ordinary SAE or the cycle-regularized SAE before the pair is jointly executed. The policy uses reconstruction error, singleton displacement and error, pair conditioning, and the pair-defect score. Singleton A and B flows provide its calibration inputs; neither AB nor BA is executed before the decision.

Every feature-pair identity used in the preceding prediction and regularization comparisons is excluded. Policy decisions are recorded and locked before joint outcomes are evaluated. The test contains 242 GPT-2 and 251 Gemma pairs, exceeding the planned minimums of 120 and 64. In Table 9, "Selected" is the fraction of pairs for which the policy chooses the cycle-regularized SAE. The primary difference is

$$\Delta_{\text{select}} = \mathbb{E}[Q_{\text{always reg}} - Q_{\text{policy}}]\,. \tag{95}$$

A positive value means that adaptive selection has lower interaction residual than applying cycle regularization to every pair.

On GPT-2, the point estimate favors selection by 1.10×10^{-6}, but the 95% interval includes zero. On Gemma, the gain is 5.14×10^{-6} and the interval is positive. The requirement that selection beat always-regularized execution in both model families is therefore not met.

A random policy that chooses the regularized SAE at the same rate controls for the possibility that selection frequency alone creates the gain. Relative to that matched-rate policy, geometry improves Q by 6.06×10^{-6} on GPT-2 (95% CI $[0.05, 13.22] \times 10^{-6}$) and 8.90×10^{-6} on Gemma ($[3.00, 15.61] \times 10^{-6}$). The pair-specific score contains information, but adaptive regularization is not uniformly better than regularizing every pair. Compared with always regularizing, the adaptive policy increases error on 5.8% of GPT-2 pairs and 6.4% of Gemma pairs.

Target achievement and total displacement remain similar across policies. Gemma's hidden and logit p95 errors improve; GPT-2's do not. At RK4-8, one joint two-control execution uses 64 field evaluations. The two singleton calibration flows require another 64, so the adaptive policy uses 128 field evaluations in total, compared with 64 for always regularizing. The RK4-16 comparison in Appendix D uses 128 evaluations for the joint execution alone.

A residual-preserving reference has zero Q by construction because its declared singleton effects add exactly. It is therefore informative only for target fidelity and the assumptions behind the reference, not as a competitor on Q. Directional Spearman remains above .976 both when the support stays fixed and when a support change is observed. Only the fixed-support result follows from the smooth derivation; the support-change result is empirical and does not constitute a saltation model.

J.6 Boundary of the behavioral claim

We also test whether the singleton controls support an exact behavioral composition experiment. A development comparison uses JSON validity and known arithmetic answers. The first task has a ceiling: the unsteered model already produces 100% valid JSON and 99.0% correct answers, and none of 24 feature–strength settings changes either rate.

A second comparison introduces task headroom: unsteered performance is 50.0% valid JSON and 74.0% correct. The intervention hook executes and produces nonzero feature, hidden-state, and logit changes, but none of 36 singleton arms improves either exact behavioral metric. Without two individually effective controls, an AB/BA behavioral comparison cannot test useful composition. We therefore make no claim about behavioral interaction or task improvement from this branch. This boundary does not alter the hidden-state and logit-fidelity results above.

K From local interaction to output and long horizons

The defect in Equation (8) is a local hidden-space vector. This section asks what that vector can imply at three scales. First, can an interaction point outside the directions available from one decoded latent step? Second, does such a direction produce a distinct next-token logit effect? Third, how can a small per-step difference accumulate when controls are repeated?

K.1 Directions created by interaction

At a hidden state h, define the decoder tangent space

$$T_h = \operatorname{Im} J_D(E(h)). \tag{96}$$

Every infinitesimal latent change is mapped by J_D into this space. Thus T_h contains all hidden directions available from one local decoded movement. Let $W_1, \dots, W_m$ be ambient control fields. Their pairwise brackets can add directions that are absent from T_h. We count those new independent directions with

$$\iota(h) = \operatorname{rank} \operatorname{span}\left(T_h, \{[W_i, W_j](h)\}_{i<j}\right) - \operatorname{rank} T_h. \tag{97}$$

If $\iota(h) = 0$, every pairwise interaction remains inside the decoder tangent space. If $\iota(h) > 0$, at least one pair produces a new local hidden direction. This is a one-step accessibility statistic: it appends pairwise brackets once. It is not the dimension of the full Lie closure, which would also append brackets of brackets and continue recursively.

Residual-preserving constant translations have zero pairwise brackets, so they always give $\iota = 0$. Ambient re-encoding can give a positive value. A three-dimensional example makes the distinction concrete. Consider

$$D(x, y) = (x, y, xy), \qquad \widetilde{E}_\delta(h) = (h_1 + \delta \sin h_3, h_2). \tag{98}$$

The two decoder tangent directions span a two-dimensional plane. Extending the two coordinate translations by re-encoding produces

$$[W_1, W_2](D(x, y)) = (0, 0, \delta y \cos(xy)), \tag{99}$$

which points outside that plane whenever its third component is nonzero. The base tangent rank is two, while the tangent space augmented by this bracket has rank three. In contrast, a residual-preserving lift and an exact invertible chart remain at rank two.

K.2 MEASURED BRACKET SPANS

We next compare this phenomenon with learned charts for DistilGPT2, Qwen2.5-0.5B, and Qwen2.5-1.5B. Each model has two independently trained charts at ranks 16 and 64. At each of 32 held-out states, eight orthonormal latent controls produce $\binom{8}{2} = 28$ brackets. The rank calculation includes the complete decoder tangent space, not only the eight sampled control directions.

Very small brackets are removed before numerical rank estimation. A bracket is retained when its norm, divided by the product of its two parent-field norms, exceeds 10^{-4}. Retained vectors are normalized, and a singular direction is counted when its singular value exceeds 10^{-3} times the largest one. The full threshold analysis appears in Appendix S.

Three norm-matched controls determine whether a high rank is specific to the measured brackets: isotropic random vectors, vectors lying entirely outside T_h, and brackets from random rank-one affine fields. All four groups, including the measured brackets, reach median rank increase 28. Random vectors therefore saturate the raw rank statistic just as easily, so the value 28 is not evidence of special interaction structure.

The singular-value distribution is more informative. Measured brackets have transverse fraction 0.890–0.927 at rank 16 and 0.488–0.711 at rank 64, compared with 0.958–1.000 for the controls. Their median effective rank is 15.5–19.8 at chart rank 16 and 19.9–22.0 at rank 64, whereas every control family has effective rank 27.5–27.8. The leading four measured modes contain 36.4–50.4% of total energy, compared with 17.2–18.8% for the controls. The measured interactions therefore occupy a more concentrated subspace, although their raw ability to increase rank is not unusual.

K.3 CAN A DECODED TANGENT MOVEMENT REPRODUCE THE OUTPUT EFFECT?

For each rank-16 model and chart seed, we select the pair with the largest component outside T_h using hidden geometry only. No output values enter this selection. We execute its four-leg commutator loop from Equation (2) on eight held-out states. Let B_h have orthonormal columns spanning T_h, let δh_{amb} be the resulting hidden displacement, and let $\delta \ell_{\mathrm{amb}}$ be the change in centered next-token logits.

We estimate how each column of B_h changes the logits and solve

$$\min_{c} \ \|\delta \ell_{\mathrm{amb}} - J_\ell(h) B_h c\|_2 \quad \text{s.t.} \quad \|c\|_2 \le \|\delta h_{\mathrm{amb}}\|_2 . \tag{100}$$

Because B_h is orthonormal, $\|c\|_2$ is the hidden norm of the candidate tangent displacement. The constraint therefore gives the comparator access to every local decoded direction, but never more hidden displacement than the ambient loop. After choosing c with the local Jacobian, we evaluate the endpoint $h + B_h c$ with a new nonlinear transformer-suffix pass. The reported residual is therefore measured with the actual suffix rather than with the linear fit alone.

All chart and suffix calculations use float64 because float32 introduces a vocabulary-wide numerical floor at the smallest loop. We check geometric agreement at $\epsilon = 0.05$ and output effects at $\epsilon = 0.2$. The output effect must also scale approximately as ϵ^2 between 0.1 and 0.2.

All six model–seed evaluations satisfy these checks. At the smallest step, the loop/bracket cosine is 0.9988–0.9999. The median log–log output exponent is 1.975–2.036, close to the expected value two. At $\epsilon = 0.2$, the centered-logit norm is 1.67×10^{-2}–1.07×10^{-1}, and the best equal-norm tangent endpoint leaves 47.9%–91.9% of that effect unmatched.

This unmatched fraction is not specific to brackets. The three norm-matched control families leave still larger median fractions, 73.7%–97.1%. Across the 18 model–seed–control comparisons, the mean control-minus-measured difference is 0.059–0.270, and 17 paired bootstrap intervals exclude zero. Generic ambient perturbations are therefore harder, not easier, to reproduce with a decoder-tangent movement. The experiment establishes a small output effect with the predicted scale, but it does not establish a uniquely bracket-specific output component. No evaluated loop changes the top-1 token. Table 16 and Appendix S provide the complete values and residual definition.

K.4 Searching states near a token decision boundary

A top-1 change is most plausible when the two leading baseline logits are close. We therefore scan 5,248 natural-text prefixes and select 200 states in each of four top-two margin intervals: 0–0.05, 0.05–0.1, 0.1–0.2, and 0.2–0.5. A single Qwen-0.5B rank-16 control pair is chosen from hidden geometry before any output is inspected.

At $\epsilon = 0.2$, none of the 800 ambient loops changes the top-1 token. The residual-preserving order-null control also produces no change. The Wilson 95% upper bound is 1.89% in the smallest-margin bin and 0.478% over all 800 states. The median loop/bracket cosine remains 0.9988, so the null token result does not arise from failure to produce the intended geometric interaction. Only 4.17% of the scanned prefixes have margin below 0.05. Even after targeting these sensitive states, the local effect is too small to support a visible token claim (Appendix U).

K.5 Accumulation over repeated steps

A local discrepancy can matter over a long horizon even when one step is small. Suppose the ambient and residual-preserving executions differ by at most $c\epsilon^2$ at each step, and the ambient step map is at most $1 + \beta\epsilon$ Lipschitz. Repeatedly applying the triangle inequality gives

$$\left\| h_T^{\mathrm{amb}} - h_T^{\mathrm{res}} \right\| \le cT\epsilon^2 e^{\beta T \epsilon}. \tag{101}$$

The factor T counts opportunities to introduce a mismatch, while the exponential factor bounds how later steps can amplify earlier ones. Appendix V derives this inequality one recurrence at a time.

The finite-horizon comparison uses one PolySAE chart and 96 held-out pair–state observations covering 12 geometry-selected feature pairs. A separate output analysis retained only 10 pairs after target-token qualification; this horizon comparison uses all 12 and does not apply that filter. Both extension rules are iterated with step size 0.02 for $T \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$. The nine horizons give 864 endpoint observations.

Residual-preserving execution remains order-invariant to 10^{-10}. Under ambient re-encoding, the median AB–BA hidden gap grows from 3.49×10^{-4} at $T = 1$ to 0.442 at $T = 256$, with fitted exponent 1.18. Symmetric Strang splitting reduces the final median gap to 0.181 but does not remove it; its fitted exponent is 1.67. Symmetric splitting cancels the leading commutator while motion remains smooth. Once sparse-support events alter the endpoint map, that local cancellation no longer prevents accumulation. These exponents describe this finite range; they are not universal scaling laws.

Changing AB to BA never changes the top-1 token at any horizon. At $T = 256$, the Wilson 95% upper bound is 3.85%. Changing the extension rule does alter tokens: the first ambient-versus-residual difference appears at $T = 16$, and at $T = 256$ the extensions disagree on 18 of 96 states, or 18.75% (95% CI $[12.2, 27.7]\%$). Among the 96 states, 30 extensions reach the same new token, 16 reach different new tokens, two change only under ambient re-encoding, and 48 retain the baseline token under both rules (Table 18).

K.6 Is the long-horizon token gap direction-specific?

We compare the learned extension difference with two isotropic controls. For every pair–state and horizon, 32 trajectories are formed by summing random unit steps. The first control is rescaled to match the residual-preserving endpoint's total displacement from the anchor. It asks what an arbitrary perturbation of the full trajectory length would do. The second is added at the residual-preserving endpoint and rescaled to match the exact distance between the ambient and residual-preserving endpoints. It asks whether the observed token disagreement exceeds that of a generic perturbation with the same final norm. The p99 relative norm-matching error is below 10^{-10}.

At $T = 256$, the total-displacement control changes the baseline token in 91.24% of trajectories (95% CI $[89.88, 92.55]\%$), compared with 50.0% for ambient and 47.9% for residual-preserving endpoints. The more relevant equal-gap control changes the residual endpoint's token in 15.04% of cases ($[11.43, 18.95]\%$). The learned extension difference is only 3.71 percentage points larger, with interval $[-1.17, 8.89]$. Extension choice can visibly alter a long trajectory, but this experiment does not distinguish its token effect from a generic equal-norm displacement.

K.7 Do interaction vectors reproduce across chart seeds?

The final comparison asks whether two independently trained charts recover the same ambient interaction. For each rank-16 model, coordinates are matched across two chart seeds using decoder-tangent signatures on 32 shared held-out states. We use a Hungarian assignment, retain matches with absolute cosine at least 0.5, and align signs before computing interactions.

The retained coordinate fraction is 0.50 for DistilGPT2, 0.625 for Qwen-0.5B, and 0.50 for Qwen-1.5B. These matches give 28, 45, and 28 interaction pairs. Their magnitude Spearman correlations are -0.008, 0.066, and 0.120, with permutation p-values 0.969, 0.668, and 0.546. Top-10 overlaps are 3, 2, and 4, also consistent with their permutation nulls. The median cosine between matched interaction vectors is 0.032, 0.041, and 0.022.

The measured ambient interactions therefore do not reproduce across these chart seeds, even after local features are matched. This does not imply that the base language model lacks a stable interaction under every possible controller; it shows that these learned charts do not identify one. The saved records contain full bracket vectors and magnitudes but not the separate ordered components K_{AB} and K_{BA}. They cannot determine whether the simpler binary choice of lower-error order is more stable than the full interaction vector. The residual-preserving comparator remains exactly order-invariant for every seed.

L Sparse boundaries require event semantics

A TopK encoder is smooth only while its selected support is fixed. A rank swap can keep the state continuous while changing its derivative. For the overcomplete SAE, the embedding assumption of Section 3 applies only to active coordinates inside a fixed-support cell. At a crossing with guard $g(x) = 0$, normal $n = \nabla g$, pre/post fields f^-, f^+, and reset R, the first-order sensitivity jumps by the saltation matrix

$$S = J_R + \frac{(f^+ - J_R f^-)n^\top}{n^\top f^-}. \tag{102}$$

Figure 14 illustrates Equation (102) for a continuous state ($J_R = I$). On the 33 discovery cases of Appendix T, smooth nested BCH has p90 relative error 1.37×10^4; freezing support lowers it to 1.6×10^{-3} and localizes the failure. Root-localized saltation lowers p90 error to 0.017 and rescues $32/33$ cases. The remaining event-order corner has two directional derivatives; a two-cone model recovers both with maximum error below 5.1×10^{-9}. This empirically realizes the hierarchy "smooth Lie geometry, saltation at events, directional cones at corners" in a learned sparse controller (Appendix T).

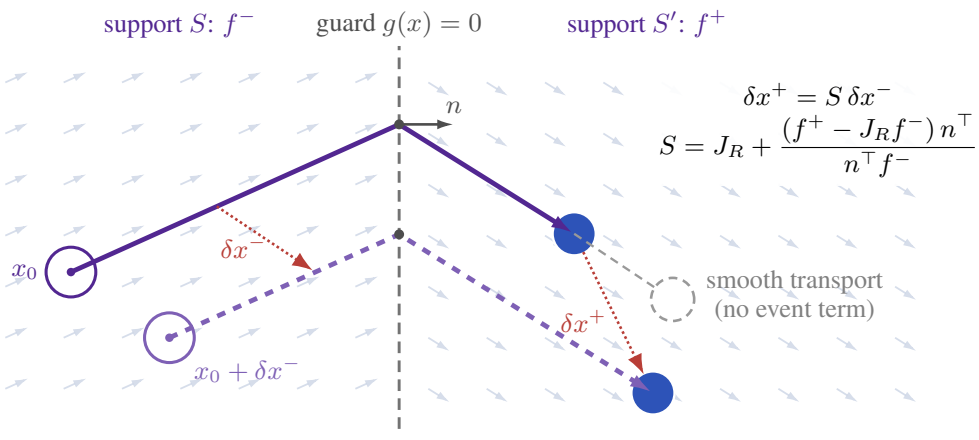


Figure 14: **A support swap transports perturbations by a saltation matrix.** Nearby trajectories cross the guard at different times (here the perturbed one crosses earlier by $n^\top \delta x^- / n^\top f^-$), so after the event their separation is $\delta x^+ = S\,\delta x^-$ rather than the smoothly transported δx^- (dashed ring). The picture is exact for constant pre- and post-event fields and a continuous state, $J_R = I$.

Native MoE update event. The same mechanism is tested in the final router of OLMoE-1B-7B-0924 rather than an auxiliary chart. For router-only SGD fields, let $K(\eta) = W_{AB} - W_{BA}$ and let K^{fix} freeze second-step expert supports. The exact tested-step decomposition is

$$K(\eta) = K^{\mathrm{fix}}(\eta) + \eta \mathcal{J}_{AB}(\eta), \tag{103}$$

where $\mathcal{J}_{AB}$ is the difference of one-sided gradient jumps. Across 49 eligible discovery triplets evaluated just above their first routing crossing, the commutator has median $3.51\times$ excess over smooth extrapolation; freezing expert indices reduces it to $1.00\times$. On the held-out 38-triplet cohort, adding the event term lowers median prediction error from 0.968 to 0.081 and raises cosine from 0.252 to 0.997. Order-sign accuracy reaches 0.737, below the planned 0.80 threshold. We therefore interpret this as evidence for the event mechanism rather than a deployable optimizer (Appendix Q).

M Related work beyond one-step steering

M.1 From directions to state-dependent controls

Representation-steering methods begin with the hypothesis that a concept can be represented by a direction in hidden space. Adding that direction can then alter model behavior (Park et al., 2023; Turner et al., 2023; Zou et al., 2023; Todd et al., 2023; Wu et al., 2024). Manifold and flow methods replace one constant direction with a path or a state-dependent vector field (Wurgaft et al., 2026; Jin et al., 2026). IDEEA selects a direction from a local cluster, and GAPS uses an input-dependent posterior to gate steering dimensions (Wang et al., 2026; Fereidouni et al., 2026).

These methods determine which update to apply at the current state. Sequential composition raises an additional question: how is that update evaluated after a previous intervention moves the hidden state away from its anchor? The answer requires an ambient extension rule. If a cluster, sparse support, or gate changes, the controller also needs an event rule. Such a trajectory is piecewise smooth: ordinary brackets describe motion inside one region, while jump transport describes a boundary crossing. State dependence alone does not specify either part of this sequential execution.

SAE evaluations typically emphasize reconstruction, sparsity, interpretability, and feature completeness (Cunningham et al., 2023; Lieberum et al., 2024; Chanin et al., 2024). These properties assess the representation itself. Our cycle and interaction measurements instead assess what happens when controls are repeated through the encoder and decoder. PolySAE provides a useful curved decoder for separating those questions (Koromilas et al., 2026).

M.2 Operator closure and transfer

Scaled idempotence studies algebraic closure inside a transformer's native attention operators (Feng & Li, 2026). Our setting concerns external interventions: two latent controls are declared first and then implemented as hidden-state fields through a learned chart. Native operator closure does not imply that this implementation preserves the declared control algebra.

Context–memory experiments show another distinction. A direction can be causally effective in one context yet transfer poorly to another task (Shih et al., 2026). That is a question about where a control remains effective. Extension identifiability is a question about how the same local control is defined away from its anchor. A control may fail either test, and the two failures need not coincide.

M.3 Feature and extension identifiability

Nelson et al. (2026) ask whether independently trained models recover stable features and sparse codes. We refer to this as *feature identifiability*. Our question begins after one dictionary and one local control have already been chosen. We ask whether their values on the decoded representation determine a unique ambient field and hence a unique sequential composition. We call this *extension identifiability*.

The two questions are logically distinct. A reproducible feature can still have several off-manifold extensions with different brackets. Conversely, unstable features introduce an additional ambiguity before an extension is chosen. This distinction motivates our cross-seed comparisons: they test feature provenance and interaction portability, not the mathematical non-uniqueness of extension.

M.4 Inverse-consistent representations

Otto et al. (2023) construct reduced-order dynamical autoencoders with a left-inverse relation, using invertible activations and biorthogonal weights. Their result shows that inverse consistency can be built into an architecture. Our setting differs in two ways: the charts are overcomplete and sparse, and their active support depends on the input. We therefore measure inverse consistency on realized supports and use a support-local cycle penalty during training. This penalty is a composability objective for sparse controllers, not a claim that inverse-consistent autoencoders are new.

The differential tools themselves are classical. Lie brackets describe the leading order dependence between sequential updates (Sweeney, 2026; Piontkovskaia & Nikolenko, 2026); naturality and accessibility describe how maps transport fields and how brackets enlarge the reachable tangent space (Lee & Lee, 2012; Jurdjevic, 1996). Control identification is distinct from controllability, Gramians, and bracket-aware sequence models (Elamvazhuthi et al., 2023; Islam & Bhadani, 2026; Walker et al., 2024). Hybrid sensitivity and splitting theory supply the event corrections needed when a trajectory crosses a cell boundary (Burden et al., 2014; Kong et al., 2023; Blanes et al., 2008).

M.5 Empirical scope

The curved-chart comparison covers DistilGPT2, Qwen2.5-0.5B, and Qwen2.5-1.5B, with two chart seeds. Its output analysis contains 48 local next-token evaluations against an equal-norm tangent comparator. These results establish local geometric and logit effects; they do not establish a universal property of larger language models.

The public SAE analysis covers two GPT-2 Small release families. Matched controller comparisons use in-house GPT-2 Small and Gemma-2-2B SAEs. The TopK and OLMoE boundary examples are discovery analyses, and the OLMoE optimization updates one router rather than training a complete model. Cycle regularization establishes a replicated reconstruction–composition tradeoff under these settings, but its benefit varies across feature pairs.

The PolySAE horizon analysis uses one chart. Its order-dependent flip rate is close to an equal-gap null, so it does not identify a direction-specific mechanism. The generation experiments do not establish reliable semantic singleton control, natural-language task improvement, or feature interpretability. The paper therefore supports claims about local execution geometry, interaction residuals, and next-token logits. Generalization to larger models, semantic outcomes, and the prevalence of switching events on unselected data remains open.

N Coordinate, support-size, and conditioning controls

The width comparison in Section D has a simple observation: wider dictionaries reconstruct hidden states more accurately, but their active encoder–decoder cycles are farther from the identity. This section checks whether that observation can be explained by four simpler changes that accompany width: the choice of latent coordinates, the number of active features, decoder conditioning, or unequal coordinate gains.

N.1 Changing latent coordinates

For an active support S, recall the cycle defect and decoder metric

$$C_S = W_{\mathrm{enc},S:} W_{\mathrm{dec},:S}, \qquad A_S = C_S - I, \qquad G_S = W_{\mathrm{dec},:S}^{\top} W_{\mathrm{dec},:S}. \tag{104}$$

The released implementation stores decoder feature vectors as rows, whereas Equation (104) writes them as columns. These are the same linear map under a transpose; all calculations use the appropriate storage convention.

More generally, let R_S be an invertible change of coordinates on the active latent variables. The same hidden-space map is then represented by

$$C'_S = R_S C_S R_S^{-1}, \qquad A'_S = R_S A_S R_S^{-1}, \qquad G'_S = R_S^{-\top} G_S R_S^{-1}. \tag{105}$$

Individual matrix entries and the ordinary Euclidean norm can change under this transformation. The decoder-metric norm in Equation (68) does not: it measures the action of A_S after latent distance has

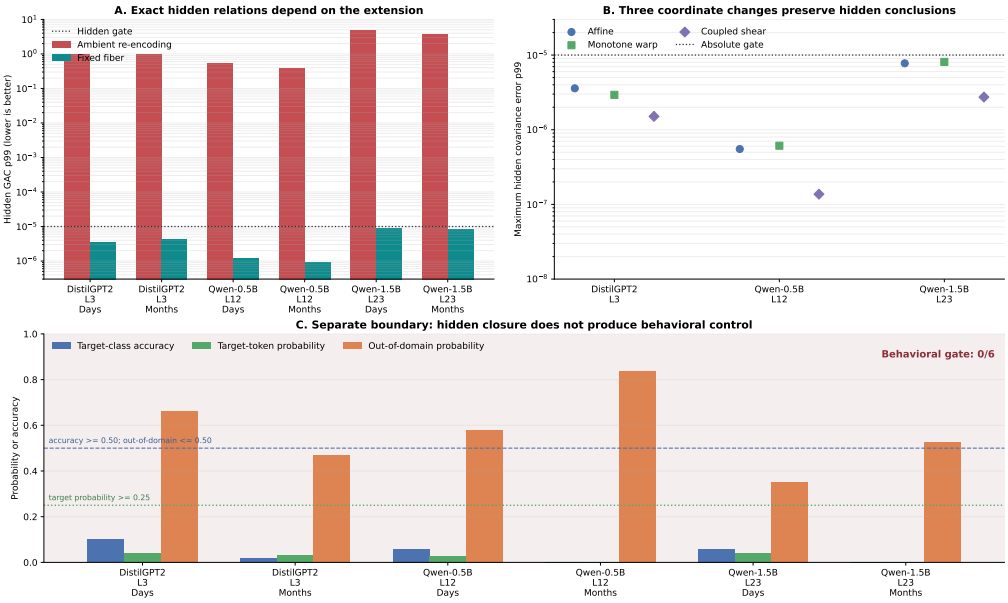


Figure 15: Coordinate-aware hidden-space conclusions survive structured changes of latent variables. (a) Ambient re-encoding and residual-preserving execution remain sharply separated in hidden closure error. (b) Affine rescaling, monotone warping, and coupled shear preserve the metric-aware comparison. (c) Geometric closure and behavioral control are separate properties.

been converted back to hidden-space distance. The spectral radius $\rho(A_S)$ is also unchanged because A'_S is similar to A_S. We therefore use $\|A_S\|_{G_S}$ and $\rho(A_S)$ for coordinate-independent comparisons and retain $\|A_S\|_2$ only as a diagnostic in the released feature coordinates.

The releases already make feature scaling nearly uniform: every decoder feature has approximately unit norm, and the largest absolute deviation is 0.00221. Figure 15 additionally applies affine rescaling, monotone warping, and coupled shear. The hidden-space comparisons are unchanged when the corresponding metric transformation is applied.

N.2 Observed width trend

Table 10 gives the complete eight-width comparison. FVU is the fraction of hidden-state variance not reconstructed, so lower FVU is better. The support size $|S|$ is the number of active features. The three cycle columns measure the same defect in different ways: $\|A\|_2$ uses the released coordinates, $\|A\|_G$ uses the decoder metric, and $\rho(A)$ is the largest eigenvalue magnitude. The condition number $\kappa(G)$ measures how close the active decoder directions are to linear dependence. Finally, γ is the root-mean-square error of the diagonal of C_S from one, and μ is the root-mean-square off-diagonal entry. Thus γ measures incorrect self-gain, while μ measures leakage from one active feature into another.

As width increases from 768 to 98,304, FVU falls from 0.400 to 0.172, whereas the coordinate-invariant cycle error $\|A\|_G$ rises from 1.129 to 3.908. The negative Spearman correlation reported below therefore means that better reconstruction is associated with worse cycle fidelity. Support size does not grow in the same way: its median stays between 38 and 47. Decoder conditioning, by contrast, worsens substantially, with median $\kappa(G)$ rising from 9.17 to 70.01. The remaining analyses separate these two possible explanations.

N.3 Controlling the number of active features

A larger support produces a larger cycle matrix and can create more opportunities for cross-feature leakage. We therefore divide supports into five disjoint bins according to $|S|$. Within each bin, we recompute one median cycle error for each of the eight dictionary widths and correlate these eight

Table 10: Complete controlled block-8 residual-input width study. Entries are medians over 10,000 realized supports per release; γ is diagonal gain error and μ is encoder–decoder cross-entry coherence in the release feature gauge. Variation across rows is across dictionaries, not a confidence interval.

Width	FVU	$\lvert S\rvert$	$\lVert A\rVert_2$	$\lVert A\rVert_G$	$\rho(A)$	$\kappa(G)$	γ	μ
768	.400	38	.983	1.129	.903	9.17	.187	.337
1,536	.333	42	1.242	1.576	1.111	14.89	.287	.353
3,072	.273	45	1.505	2.060	1.294	23.60	.370	.352
6,144	.239	47	1.726	2.583	1.438	34.84	.458	.373
12,288	.213	47	1.889	3.110	1.548	42.23	.497	.381
24,576	.194	46	2.024	3.527	1.632	49.50	.515	.410
49,152	.180	45	2.256	3.712	1.635	55.67	.918	.435
98,304	.172	47	1.972	3.908	1.487	70.01	.573	.393

errors with their FVU values. This compares widths only among examples with similar numbers of active features.

Table 11: Support-conditioned block-8 residual-input width study using the decoder-metric cycle error. Each row compares all eight widths within the same support-size bin; variation across widths is across dictionaries, not a confidence interval.

$\lvert S\rvert$ bin	Minimum samples per width	Spearman(FVU, error)	Exact p
1–16	105	-0.929	.00223
17–32	917	-1.000	$p_{\min}$
33–48	4,167	-1.000	$p_{\min}$
49–64	1,835	-1.000	$p_{\min}$
≥ 65	287	-1.000	$p_{\min}$

The association remains at most -0.929 in every support-size bin and equals -1.000 in four of the five bins. The smallest cell still contains 105 supports per width. A complementary partial Spearman calculation, using each release's median support size as the control variable, gives -1.000 with exact permutation $p = p_{\min}$. Here $p_{\min} = 2/8!$ is the smallest attainable two-sided exact value for eight widths. The result is not produced by wider dictionaries activating more features.

Two additional statistics reduce dependence on the decoder metric. In the released coordinates, the off-diagonal RMS leakage has Spearman -0.929 ($p = 0.00223$). The coordinate-invariant spectral radius has Spearman -0.857 ($p = 0.0107$). These comparisons also worsen as FVU improves, but neither one holds decoder conditioning constant.

N.4 Separating cycle error from decoder conditioning

The metric G_S converts latent-coordinate distance into hidden-space distance. When its eigenvalues have a large ratio,

$$\kappa(G_S) = \frac{\lambda_{\max}(G_S)}{\lambda_{\min}(G_S)}, \tag{106}$$

small coordinate differences can be strongly amplified in some hidden-space directions. This can increase $\lVert A_S\rVert_{G_S}$ even if the underlying cycle matrix changes less. Conditioning is therefore a genuine part of the geometry and also a possible confound for the width comparison.

The decoder metric is computed by float64 eigendecomposition with relative singularity tolerance 10^{-12}; no pseudoinverse or eigenvalue floor is used, and all 210,000 supports pass. From the narrowest to the widest release, median $\lambda_{\min}(G_S)$ falls from 0.305 to 0.117, while median $\kappa(G_S)$ rises from 9.17 to 70.0.

A first restriction keeps only supports with $\kappa(G_S) \leq 100$. It retains at least 7,606 supports at every width and leaves Spearman -1.000 ($p = p_{\min}$), but this threshold removes little of the wide-release data and is therefore weak. A stricter comparison uses only the condition-number interval covered

by the central 90% of every width, $\kappa(G_S) \in [31.09, 78.46]$. Within this common interval, $\|A\|_G$ retains Spearman -0.929 ($p = .00223$), and released-coordinate $\|A\|_2$ retains Spearman -0.762 ($p = .0368$). The spectral-radius association falls to -0.333 ($p = .428$). Thus conditioning explains an important part of the invariant width trend, especially the spectral-radius trend, but does not remove the covariant or released-coordinate associations.

Finer condition-number bins do not contain enough overlapping observations: some pooled bins have only five samples per width, and bins restricted to the common interval have only 1–28. We report those results descriptively but do not use them for the main conditional conclusion. The leakage statistic μ cannot replace this conditioning check: μ combines encoder and decoder entries, whereas $\kappa(G_S)$ depends only on the geometry of the active decoder columns.

The common-range restriction discards supports outside the overlap. To use all 10,000 supports at every width, we also fit a separate log–log regression for each width m:

$$\log \|A_S\|_{G_S} = a_m + b_m \log \kappa(G_S) + e_S. \tag{107}$$

The slope b_m measures how cycle error changes with conditioning within one release. We then evaluate every fitted line at the same condition number, $\kappa_0 = 49.39$, the geometric midpoint of the common interval. This asks what cycle error each width predicts if all widths are compared at a common level of conditioning.

Table 12: All-support conditioning regressions. Each fit uses 10,000 supports: $\log \|A\|_G = a_m + b_m \log \kappa(G)$. Fit is evaluated at $\kappa_0 = 49.39$; IQR is the within-width log-residual interquartile range.

Width	Slope b_m	R^2	Residual IQR	Fit at κ_0
768	.351	.930	.286	1.911
1,536	.368	.900	.347	2.375
3,072	.368	.827	.377	2.646
6,144	.346	.810	.366	2.871
12,288	.367	.608	.376	3.255
24,576	.397	.539	.352	3.494
49,152	.402	.534	.345	3.505
98,304	.453	.526	.325	3.290

Every fitted slope is positive, ranging from 0.346 to 0.453: within every width, more poorly conditioned supports have larger decoder-metric cycle error. The slopes are not equal, and their association with FVU is -0.738 ($p = .0458$). Conditioning therefore acts as a width-dependent amplifier rather than one common multiplicative factor.

At the shared value κ_0, fitted errors rise from 1.911 to 3.290 across the width endpoints, with Spearman -0.929 ($p = .00223$). The association is -0.857 when all fits are evaluated at the lower common endpoint 31.09 and -0.929 at the upper endpoint 78.46. Before adjustment, the endpoint error ratio is 3.46; after adjustment at κ_0, it is 1.72. On the logarithmic endpoint contrast used by the regression, conditioning accounts for 56.2% of the observed width difference and 43.8% remains. This is a descriptive decomposition, not a causal estimate.

The R^2 values fall from .930 at width 768 to .526 at width 98,304, so conditioning explains a smaller fraction of support-to-support variation in the wider releases. The absolute residual spread does not increase systematically: the residual IQR ranges from .286 to .377 and has width Spearman .071 ($p = .882$).

As a check that Equation (107) is not creating the trend through its log-linear form, we also compare medians inside three fixed condition-number bands:

κ band	Minimum n/width	Spearman	Exact p
$[45, 55]$	3	$-.833$	.0154
$[40, 60]$	9	$-.857$	.0107
$[35, 65]$	26	$-.905$	.00456

All three bands preserve the direction of the width association. The narrowest release contributes only 3, 9, and 26 supports, respectively, so these values are a functional-form check rather than precise endpoint estimates.

N.5 Separating self-gain from cross-feature leakage

The diagonal of C_S records how strongly each active coordinate reproduces itself after decoding and re-encoding. Its ideal value is one. Off-diagonal entries record leakage between distinct coordinates and should be zero. The summary statistics γ and μ in Table 10 measure these two errors separately.

Controlling jointly for median support size and median μ leaves partial Spearman -1.000 with $p = p_{\min}$. Within five pooled μ quintiles, the association ranges from -0.976 to -1.000, with at least 202 supports per width. These checks show that the width trend is not merely a consequence of the measured average leakage. They complement rather than replace the condition-number comparison, because μ and $\kappa(G_S)$ measure different objects.

The diagonal error γ rises from .187 at width 768 to .918 at width 49,152, then falls to .573 at width 98,304. The corresponding μ values are $.337, .353, .352, .373, .381, .410, .435, .393$. The same top-end reversal appears in several diagnostics. The released metadata contains no matching configuration change, so we report the irregularity without assigning a mechanism.

Table 13 removes increasingly flexible forms of diagonal gain. The scalar column is $\min_{\alpha\in\mathbb{R}} \|A_S - \alpha I\|_{G_S}$ and removes one gain shared by all active coordinates. The diagonal column is $\min_{\Lambda\in\mathrm{Diag}} \|A_S - \Lambda\|_{G_S}$ and allows a different gain for each coordinate. "Raw off-diag" simply zeros the observed diagonal without numerical optimization. Gain spread is the difference between the largest and smallest diagonal errors. The random column uses normalized Gaussian decoder vectors with $W_{\mathrm{enc}} = W_{\mathrm{dec}}^{\top}$, matched to the same hidden dimension and support size.

Table 13: Diagonal-gain controls and random-dictionary comparison. Entries are medians over the same 512 supports per width. Scalar and diagonal columns remove the best common and coordinate-specific gains, respectively; raw off-diag zeros the observed diagonal before measuring.

Width	d_{scalar}	d_{diag}	Raw off-diag	Gain spread	Random
768	.996	.815	1.066	.178	.406
1,536	1.458	1.165	1.551	.278	.436
3,072	2.002	1.536	2.093	.391	.446
6,144	2.576	1.932	2.692	.496	.464
12,288	3.116	2.332	3.188	.563	.465
24,576	3.467	2.584	3.559	.580	.455
49,152	3.764	2.702	3.896	.949	.447
98,304	3.913	2.810	4.006	.573	.463

Both optimized errors continue to rise with width. The scalar-centered value increases from 0.996 to 3.913, and the diagonal-centered value increases from 0.815 to 2.810; each has Spearman -1.000 with FVU ($p = p_{\min}$). The unoptimized off-diagonal comparator has the same qualitative pattern. Unequal self-gain is therefore present, but removing all diagonal freedom does not remove the width trend in cross-feature error. The random dictionaries remain between 0.406 and 0.465, showing that the learned active cycles are not behaving like the matched random baseline.

The common-gain objective is one-dimensional and convex. We solve it with 28 golden-section iterations over $[-2\|A_S\|_{G_S}, 2\|A_S\|_{G_S}]$. The diagonal objective is convex but nonsmooth; it is initialized at the observed diagonal and optimized for 80 LBFGS iterations. We therefore use it as a numerical decomposition, not as a certificate that a global optimum has been found. The raw off-diagonal column provides a direct comparator that does not depend on that optimizer.

N.6 What an exact fixed-support inverse would cost

The preceding controls diagnose the source of cycle error. A separate calculation asks whether the active encoder can be replaced by an exact left inverse of the active decoder. The pseudoinverse gives

the unconstrained least-squares solution. Nonnegative least squares (NNLS) imposes nonnegative coefficients, which is closer to the sign structure of a TopK code but restores the exact cycle only on coordinates that are free under its optimality conditions.

Table 14: Cost of exact or sign-constrained fixed-support inversion. SSE values are reconstruction-error ratios relative to the original encoder, so values below one are improvements. Negative/free columns are coefficient fractions, except "Neg. supports," which is the fraction of supports containing at least one negative pseudoinverse coefficient.

Width	Pinv SSE	NNLS SSE	Neg. coeff.	Neg. supports	NNLS free
768	.868	.869	.015	.428	1.000
1,536	.834	.835	.030	.713	.974
3,072	.808	.810	.056	.867	.947
6,144	.783	.787	.081	.955	.919
12,288	.769	.775	.102	.975	.893
24,576	.753	.759	.118	.969	.880
49,152	.738	.746	.130	.979	.867
98,304	.710	.720	.154	.990	.837

The pseudoinverse reduces the cycle error to numerical precision: medians range from 9.5×10^{-15} to 2.4×10^{-14}, and the maximum is 7.7×10^{-10}. It also lowers reconstruction SSE on every sampled support. The SSE ratio falls from 0.868 at width 768 to 0.710 at width 98,304. However, the fraction of negative coefficients rises from 1.5% to 15.4%, and the fraction of supports containing at least one negative coefficient rises from 42.8% to 99.0%. Exact inversion can therefore conflict with the nonnegative interpretation of sparse feature activations.

NNLS also improves reconstruction on every sampled support, with SSE ratios from 0.869 to 0.720. Its free-coordinate fraction falls from 1.000 to 0.837, so its exact-cycle statement applies only to that free set. Neither repair is free in magnitude: median code norms increase by factors 1.11–1.21 for the pseudoinverse and 1.09–1.20 for NNLS. These calculations show that a fixed-support inverse can restore the differential cycle, but they also expose the sign and coefficient-size changes hidden by the algebraic repair.

O Residual handling in Manifold Steering

This comparison asks how Manifold Steering depends on the part of a hidden state that is not represented by its low-dimensional chart. Let h_0 be the anchor hidden state, let $z_0 = E(h_0)$ be its chart coordinate, and define the anchor reconstruction residual

$$r_0 = h_0 - D(z_0). \tag{108}$$

For an edited path coordinate z', the released controller returns

$$h^{\mathrm{res}}(z') = D(z') + r_0. \tag{109}$$

The chart changes only the decoded component of the state; every path point retains the anchor residual. We compare this rule with projection onto the decoded manifold,

$$h^{\mathrm{proj}}(z') = D(z'). \tag{110}$$

The two states differ by exactly r_0 at every matched coordinate: $h^{\mathrm{res}}(z') - h^{\mathrm{proj}}(z') = r_0$. This is the only change between the two conditions. They use the same chart, prompts, path coordinates, path pairs, and scoring procedure.

O.1 Why the smooth identity is not the relevant model

The encoder assigns a hidden state to its nearest centroid. It is therefore constant inside each centroid cell and discontinuous at cell boundaries. Away from a boundary,

$$J_E(h) = 0, \qquad C_h = J_E(h)J_D(E(h)) = 0, \qquad A_h = C_h - I = -I. \tag{111}$$

Although the differential cycle is maximally different from the identity, this does not produce the smooth defect studied in Section 4. That derivation assumes a differentiable encoder and a state-dependent field that is re-evaluated along the trajectory. Here the encoder is nonsmooth, and each path coordinate is decoded relative to the same anchor without re-encoding the previous path point. The comparison below therefore tests sensitivity to residual handling, not the smooth bracket formula.

Table 15: Effect of deleting the anchor reconstruction residual. "Difference" is projection minus residual-preserving execution. Higher coherence and lower distance are favorable. Intervals use 10,000 paired bootstrap resamples over 50 path pairs. Isometry depends only on the chart and therefore has no path-paired interval.

Score	Residual-pres.	Projection	Difference	95% CI
Geometric coherence	.7810	.0853	$-.6956$	$[-.7041, -.6874]$
Geometric worst coherence	.7412	.0334	$-.7078$	$[-.7174, -.6977]$
Geometric manifold distance	.3412	16.3488	$+16.0076$	$[15.4209, 16.5831]$
Geometric geodesic distance	.7044	17.7433	$+17.0389$	$[16.4105, 17.6706]$
Linear coherence	.7168	.7168	0	$[0, 0]$
Linear manifold distance	1.4029	1.4029	0	$[0, 0]$
Geometric isometry	.9894	.9894	0	–
Linear isometry	.8865	.8865	0	–

O.2 Results

Deleting r_0 has a large immediate effect. Its median norm is 29.250, while the median anchor norm is 35.755; equivalently, the median ratio $\|r_0\|_2 / \|h_0\|_2$ is 0.82. Equations (109) and (110) show that the hidden displacement caused by the swap is exactly $\|r_0\|_2$, before any path interaction is considered. The change is therefore zeroth order in path length. It is not the $O(\epsilon^2)$ effect predicted for a small commutator loop.

Table 15 shows that deleting the residual sharply reduces both geometric coherence scores and increases both geometric distances. For example, geometric manifold distance rises from 0.3412 to 16.3488. In contrast, every linear-path score and both isometry controls are exactly unchanged. The chart itself and the linear paths have not changed; only the construction of the hidden path from its chart coordinate has changed. The geometric scores are therefore highly sensitive to whether the anchor residual is retained.

We separately check whether a small smooth interaction could explain this difference. Across 16 anchors, eight directions, 28 direction pairs, and five step sizes, the analytic defect has median norm zero and maximum norm 4.51×10^{-16}. The corresponding residual-preserving commuting loop has maximum relative order gap 2.22×10^{-14}. Because these quantities are at numerical zero, neither a meaningful quadratic slope nor a directional cosine can be estimated. We also inspect 2,240 first steps and observe no nearest-centroid boundary crossing. Thus neither a smooth bracket nor an observed switching event accounts for the large score change.

An iteratively re-encoded path is not part of this comparison. Each supplied path coordinate is evaluated independently against the same anchor state, so a new centroid assignment is not computed after one path point becomes the next hidden state. The result has a narrow interpretation: residual preservation is essential to the reported geometric scores in this pipeline, but the observed change is caused by deleting a large reconstruction residual rather than by the smooth sequential-interaction defect derived in this paper.

P Execution rules in existing steering methods

Existing steering methods use different rules after the hidden state leaves its anchor. FLAS reevaluates a learned ambient field after each Euler step (Jin et al., 2026). Manifold Steering decodes path coordinates while preserving the anchor residual (Wurgaft et al., 2026). CircuitSteer adds a fixed pooled SAE decoder vector; same-layer constant additions commute (Saadatinia et al., 2026). These methods therefore instantiate, respectively, state-dependent ambient execution, residual-preserving

execution, and constant translation. Their distinction motivates treating the execution rule as part of the controller definition rather than as an implementation detail. The classification follows the cited method descriptions and released implementations.

Q NORMALIZED TOPK CONVENTION

OLMoE-1B-7B-0924 has 16 layers, 64 experts, TopK 8, and weights selected experts by full-router probabilities. For active support S, full-router probability p_j, and expert-output covector c_i, the router-logit gradient is

$$q_j = p_j \left(\mathbf{1}[j \in S] c_j - \sum_{i \in S} p_i c_i \right). \tag{112}$$

Replacing S by the one-swap support predicts the event jump and matches automatic differentiation tests. Equation (112) uses OLMoE's full-router weights. If a model renormalizes selected weights as $\widetilde{p}_i = p_i / \sum_{k \in S} p_k$, then, for fixed support,

$$q_j = \mathbf{1}[j \in S] \widetilde{p}_j \left(c_j - \sum_{i \in S} \widetilde{p}_i c_i \right). \tag{113}$$

The swap jump must be derived under the convention implemented by the model.

R SYMMETRIC SPLITTING AT A SWITCHING GUARD

Symmetric splitting applies one field for half a step, the other for a full step, and the first for another half step. We compare

$$ABA = \Phi_A^{\epsilon/2} \circ \Phi_B^{\epsilon} \circ \Phi_A^{\epsilon/2} \quad \text{and} \quad BAB = \Phi_B^{\epsilon/2} \circ \Phi_A^{\epsilon} \circ \Phi_B^{\epsilon/2}. \tag{114}$$

For smooth linear fields, the lower-order errors cancel and the difference between these two words is $O(\epsilon^3)$. The numerical slope is 2.9996, matching the expected cubic scaling.

The cancellation changes when a trajectory crosses a switching boundary. Consider one-dimensional motion with

$$A(x) = 1, \qquad B(x) = \begin{cases} 2, & x < 0, \\ 5, & x \geq 0, \end{cases} \qquad x_0 = -\frac{\epsilon}{2}. \tag{115}$$

Field A always moves to the right at speed one. Field B moves at speed two to the left of zero and speed five after crossing zero.

For the ABA word, the first half-step under A moves the state exactly to the boundary. The full B step then starts in the fast region, and the final A half-step gives

$$-\frac{\epsilon}{2} \xrightarrow{A,\,\epsilon/2} 0 \xrightarrow{B,\,\epsilon} 5\epsilon \xrightarrow{A,\,\epsilon/2} \frac{11\epsilon}{2}. \tag{116}$$

For the BAB word, the first B half-step begins in the slow region. It uses $\epsilon/4$ units of time to travel from $-\epsilon/2$ to zero at speed two, then uses the remaining $\epsilon/4$ at speed five. Its position after this half-step is therefore $5\epsilon/4$. The following steps are

$$-\frac{\epsilon}{2} \xrightarrow{B,\,\epsilon/2} \frac{5\epsilon}{4} \xrightarrow{A,\,\epsilon} \frac{9\epsilon}{4} \xrightarrow{B,\,\epsilon/2} \frac{19\epsilon}{4}. \tag{117}$$

The two symmetric executions consequently differ by

$$\left| \frac{11\epsilon}{2} - \frac{19\epsilon}{4} \right| = \frac{3\epsilon}{4}. \tag{118}$$

Figure 16 traces both words. The gap is linear rather than cubic in ϵ, which explains the measured slope 1.0000. Symmetrization cancels the smooth commutator, but it cannot cancel the fact that the two paths reach the boundary at different times. In the saltation formula of Equation (102), the velocity changes by the ratio $f^+/f^- = 5/2$ at the crossing. This event-time correction is the first-order term that remains after smooth cancellation.

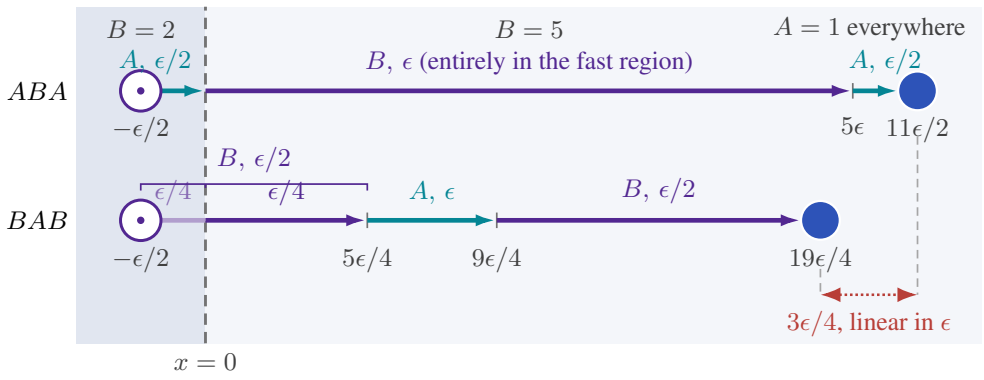


Figure 16: **Symmetric splitting does not cancel an event-time mismatch.** Both words start at $x_0 = -\epsilon/2$ (drawn to scale). ABA reaches the switch exactly at the end of its first half-step, so its whole B step runs in the fast region; BAB spends the first half of its opening B step, $\epsilon/4$, in the slow region. The endpoints differ by $3\epsilon/4$, linear rather than cubic in ϵ, matching the measured slope 1.0000.

S HIDDEN-SPACE ACCESSIBILITY AND OUTPUT EFFECTS

This section asks two separate questions. First, can an interaction between two latent controls point outside the directions available from a single decoded latent step? Second, if it can, does the new hidden-space direction produce a next-token logit change that cannot be reproduced by an equally large decoded latent step? The first question concerns hidden-space geometry. The second concerns the transformer output.

S.1 DOES AN INTERACTION ADD HIDDEN-SPACE DIRECTIONS?

At a hidden state h, the decoder tangent space is

$$T_h = \operatorname{Im} J_D(E(h)). \tag{119}$$

It contains every infinitesimal hidden displacement that can be obtained by moving once in latent space and decoding that movement. A bracket $[W_i, W_j](h)$ outside T_h is therefore a direction created by the interaction of the two controls, rather than by either single decoded movement. Equation (97) counts how many independent directions are added when these brackets are appended to a basis of T_h.

For each chart rank $k \in \{16, 64\}$, we construct eight deterministic orthonormal latent controls using seed $314159 + 1709k$. Eight controls give $\binom{8}{2} = 28$ unordered pairs. For every held-out state, we compute all 28 brackets with the exact formula in Equation (9). We first remove brackets that are too small to distinguish reliably. The normalized amplitude of pair (i, j) is

$$a_{ij}(h) = \frac{\|[W_i, W_j](h)\|_2}{\|W_i(h)\|_2 \, \|W_j(h)\|_2}. \tag{120}$$

Only pairs with $a_{ij}(h)$ above the amplitude threshold enter the rank calculation. We normalize the retained vectors, append them to an orthonormal basis of T_h, and compute singular values. A singular direction is counted when its value exceeds the SVD threshold times the largest singular value.

The primary amplitude and SVD thresholds are 10^{-4} and 10^{-3}, respectively. To check that the conclusion is not determined by this single choice, we evaluate all nine combinations

$$\tau_{\text{amp}} \in \{10^{-5}, 10^{-4}, 10^{-3}\}, \qquad \tau_{\text{svd}} \in \{10^{-4}, 10^{-3}, 10^{-2}\}. \tag{121}$$

For every model, chart seed, and rank, the median rank increase is positive in at least six of these nine combinations. It remains positive for every model, seed, and rank whenever τ_{amp} is 10^{-5} or 10^{-4}, regardless of the SVD threshold. Some medians become zero when the stricter amplitude threshold 10^{-3} is used: DistilGPT2 seed 1 at rank 64, DistilGPT2 seed 2 at both ranks, and Qwen-1.5B at

both seeds and ranks. In these cases, the amplitude filter removes most brackets before the SVD is computed. DistilGPT2 seed 1 at rank 16 and every Qwen-0.5B configuration remain positive even at this stricter threshold.

Two controls clarify what this rank increase means. Residual-preserving constant translations have zero brackets and therefore add no directions. An exact nonlinear chart may have nonzero brackets, but those brackets remain inside its decoder tangent space and also add no directions. Ambient re-encoding can add directions outside that space. However, norm-matched random vectors add enough directions to saturate the same rank statistic. Positive rank inflation alone is therefore not evidence of special structure. The informative comparison is the singular-value spectrum: measured brackets concentrate in fewer dominant modes than the norm-matched null vectors, as reported in the main text.

S.2 Do the new directions have distinct output effects?

For this comparison, each state uses the control pair with the largest component orthogonal to T_h. Pair selection uses hidden-space geometry only; no logits are inspected. The chart and transformer suffix are then evaluated in float64. Let B_h be a matrix whose orthonormal columns span T_h, and let δh_{amb} be the hidden displacement produced by the ambient commutator loop. Its centered next-token logit effect is $\delta \ell_{\mathrm{amb}}$.

We estimate how each tangent direction changes the logits by evaluating the transformer suffix at $h + 0.02b$ and $h - 0.02b$ for every column b of B_h. The centered difference gives the tangent-output matrix $J_\ell(h)B_h$. Equation (100) then chooses coefficients c that best reproduce $\delta \ell_{\mathrm{amb}}$ while satisfying

$$\|B_h c\|_2 = \|c\|_2 \le \|\delta h_{\mathrm{amb}}\|_2 . \tag{122}$$

Thus the comparator may use any decoder-tangent direction, but it may not use a larger hidden intervention than the measured loop. When ordinary least squares already satisfies this radius, it is used directly. Otherwise, a scalar ridge penalty is increased by bisection until the solution lies on the allowed boundary.

The linear solve is used only to choose the tangent displacement. We then form the endpoint $h + B_h c$ and run the actual nonlinear transformer suffix again. If its centered logit effect is $\delta \ell_{\tan}$, the reported residual is

$$R_{\mathrm{out}} = \frac{\|\delta \ell_{\mathrm{amb}} - \delta \ell_{\tan}\|_2}{\|\delta \ell_{\mathrm{amb}}\|_2}. \tag{123}$$

For example, $R_{\mathrm{out}} = 0.479$ means that the best equal-norm tangent endpoint leaves 47.9% of the measured centered-logit effect unmatched. It does not mean that 47.9% of generated tokens change.

S.3 Norm-matched controls and interpretation

The same output comparison is repeated for three control displacements: an isotropic random vector, a random vector lying exactly in $T_h^{\perp}$, and a bracket generated by a random rank-one affine field. Each control is rescaled within the same state to match $\|\delta h_{\mathrm{amb}}\|_2$. This matching is important because a larger hidden perturbation would generally be harder to reproduce in output space.

At $\epsilon = 0.2$, the measured brackets have median R_{out} between 0.479 and 0.919. The three null families have medians between 0.737 and 0.971. For each model and seed, we compute the paired state-level difference "null residual minus measured residual" and bootstrap states 10,000 times. Seventeen of the 18 resulting 95% intervals exclude zero. The only exception is the random-affine control for Qwen-0.5B seed 1, whose mean difference is 0.095 with interval $[-0.036, 0.215]$.

These results do not show that the measured bracket effects are uniquely inaccessible from the decoder tangent space. The null vectors are usually even harder to match. Instead, the comparison shows that the measured brackets are more structured and more tangent-explainable than generic equal-norm ambient directions. Table 16 also confirms the expected quadratic scale and close agreement between the finite loop and analytic bracket. None of the 48 evaluated states changes its top-1 token, so this analysis establishes a small next-token logit effect, not a visible generation-level change.

Table 16: Output effects in float64 at $\epsilon = 0.2$. Scale is the median log–log exponent of the centered-logit norm from $\epsilon = 0.1$ to 0.2; a value near two is the expected quadratic scaling. Residual is the median R_{out} from Equation (123) after nonlinear evaluation of the best equal-norm tangent endpoint.

Model / seed	Scale	Loop cosine†	Logit norm	Residual
DistilGPT2 / 1	1.975	0.9999	4.53×10^{-2}	0.479
DistilGPT2 / 2	2.005	0.9999	3.93×10^{-2}	0.529
Qwen2.5-0.5B / 1	2.017	0.9988	1.67×10^{-2}	0.653
Qwen2.5-0.5B / 2	2.036	0.9995	2.01×10^{-2}	0.623
Qwen2.5-1.5B / 1	2.010	0.9989	1.07×10^{-1}	0.919
Qwen2.5-1.5B / 2	2.010	0.9990	8.35×10^{-2}	0.905

†Loop/bracket cosine is measured at the finest $\epsilon = 0.05$. All 48 state-level evaluations obey the equal-norm constraint; top-1 flip rate is zero in every row.

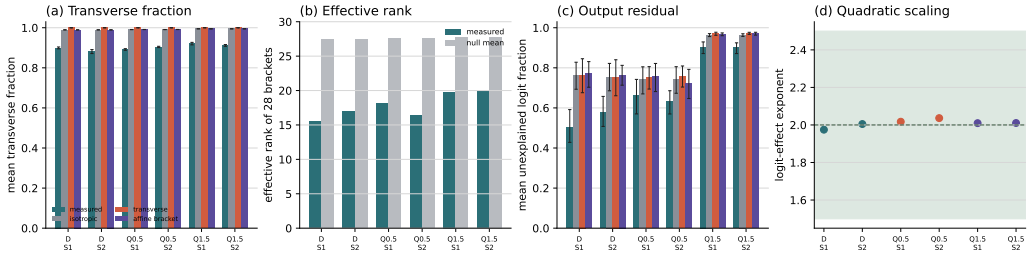


Figure 17: Nulls separate generic extrusion from bracket structure. D denotes DistilGPT2; Q0.5/Q1.5 denote Qwen2.5 sizes; S1/S2 are chart seeds. (a) Random norm-matched vectors are even more transverse than measured brackets. (b) Measured 28-bracket spans have much lower effective rank than the mean of three null families. (c) Measured loop effects leave less unexplained centered-logit mass than norm-matched nulls after the same best-tangent fit; bars show paired-state bootstrap 95% intervals for the mean. (d) The measured effects nevertheless obey the predicted quadratic scale.

T PROGRESSIVE SPARSE-EVENT EVIDENCE

The same 33 discovery observations are retained at every stage. Dynamic-support trajectories recover cubic scaling and direction but not magnitude. Smooth transport and three rounds of event-consistent relinearization remain insufficient because nearby trajectories cross guards at different times. Root-localizing every swap and inserting Equation (102) repairs 32 cases. In the remaining near-coincident scheduled switch and endogenous swap, the sign of the event-time derivative selects one of two directional cones. Table 17 records the full progressive mechanism study.

Table 17: Prediction error on the same 33 discovery cases at each stage.

Predictor	Median error	p90 error
Smooth nested BCH	3.24×10^{3}	1.37×10^{4}
Fixed support	9.0×10^{-4}	1.6×10^{-3}
Branch transport	0.011	1.85
Relinearization	0.056	2.49
Event-time saltation	3.3×10^{-4}	0.017
Directional cones	$< 5.1\times10^{-9}$	–

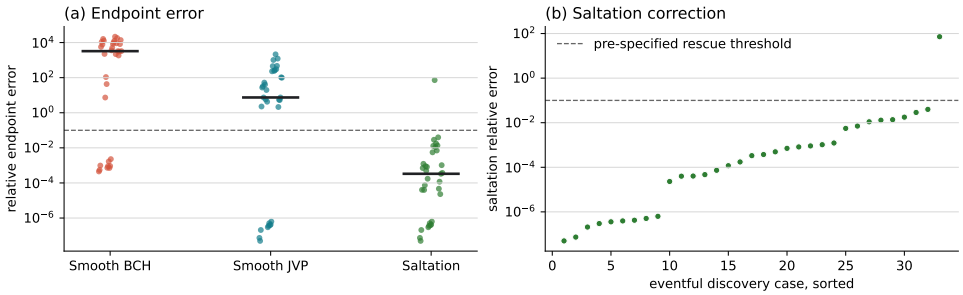


Figure 18: Event-time correction on the 33-case discovery cohort. Smooth BCH and smooth Jacobian transport become inaccurate after support changes. Root-localized saltation reduces the median error by more than five orders of magnitude and places 32 of 33 cases below error 0.1. This figure studies conditional correction, not event prevalence on unselected data.

U Margin-targeted top-1 analysis

The Qwen-0.5B analysis uses the output of zero-based block 11 (`blocks.11.hook_resid_post`), one rank-16 chart, and an eight-step RK4 loop, and $\epsilon = 0.2$. One pair is selected by hidden geometry on 16 reference states and is unchanged for all 800 outputs. Median displacement is 1.68×10^{-3}, loop/bracket cosine is 0.9988, and the order-null loop is 1.26×10^{-15}; no margin bin contains a top-1 change.

V Horizon accumulation

Let $h_{t+1}^{\mathrm{amb}} = F_t^{\mathrm{amb}}(h_t^{\mathrm{amb}})$ and $h_{t+1}^{\mathrm{res}} = F_t^{\mathrm{res}}(h_t^{\mathrm{res}})$ start from the same state h_0. The two maps describe one step under ambient re-encoding and residual-preserving execution. Define the trajectory separation

$$d_t = \left\| h_t^{\mathrm{amb}} - h_t^{\mathrm{res}} \right\| \tag{124}$$

and the one-step execution mismatch, evaluated at the residual-preserving state,

$$\eta_t = \left\| F_t^{\mathrm{amb}}(h_t^{\mathrm{res}}) - F_t^{\mathrm{res}}(h_t^{\mathrm{res}}) \right\|. \tag{125}$$

Adding and subtracting $F_t^{\mathrm{amb}}(h_t^{\mathrm{res}})$ gives

$$\begin{aligned} d_{t+1} &= \left\| F_t^{\mathrm{amb}}(h_t^{\mathrm{amb}}) - F_t^{\mathrm{res}}(h_t^{\mathrm{res}}) \right\| && (126) \\ &\le \left\| F_t^{\mathrm{amb}}(h_t^{\mathrm{amb}}) - F_t^{\mathrm{amb}}(h_t^{\mathrm{res}}) \right\| + \eta_t. && (127) \end{aligned}$$

If F_t^{amb} is L_t-Lipschitz on the region containing both trajectories, the first term is at most $L_t d_t$. Therefore

$$d_{t+1} \le L_t d_t + \eta_t. \tag{128}$$

Because $d_0 = 0$, the first two steps satisfy $d_1 \le \eta_0$ and $d_2 \le L_1\eta_0 + \eta_1$. Repeating the substitution shows that a mismatch introduced at step t is multiplied by every later sensitivity factor:

$$\left\| h_T^{\mathrm{amb}} - h_T^{\mathrm{res}} \right\| \le \sum_{t=0}^{T-1} \left(\prod_{s=t+1}^{T-1} L_s \right) \left\| F_t^{\mathrm{amb}}(h_t^{\mathrm{res}}) - F_t^{\mathrm{res}}(h_t^{\mathrm{res}}) \right\|. \tag{129}$$

The empty product for $t = T-1$ equals one: the final mismatch has no later map through which to propagate. Earlier mismatches can be amplified or contracted by the subsequent dynamics.

For step size ϵ, suppose each mismatch obeys $\eta_t \le c\epsilon^2$. If each ambient step is only mildly expansive, $L_t \le 1 + \beta\epsilon$, then

$$\prod_{s=t+1}^{T-1} L_s \le (1+\beta\epsilon)^{T-1-t} \le e^{\beta\epsilon(T-1-t)} \le e^{\beta T\epsilon}. \tag{130}$$

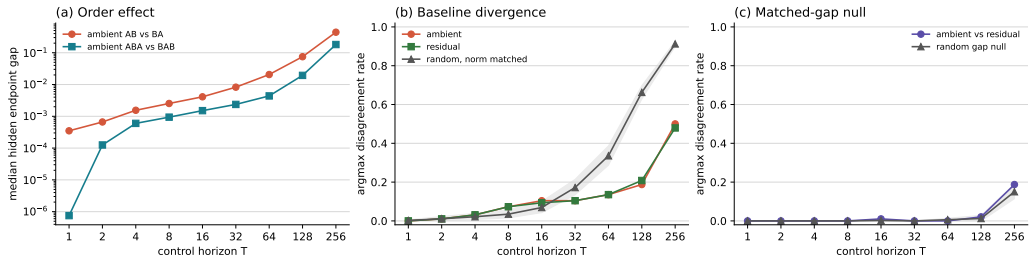


Figure 19: Horizon growth and matched trajectory nulls. (a) Sequential and Strang ambient order gaps compound with horizon. (b) An isotropic trajectory matched to the residual endpoint's total hidden displacement changes top-1 more often than either learned extension. (c) An isotropic perturbation matched to the exact ambient–residual gap has a top-1 disagreement rate compatible with the observed extension gap. Shading shows pair-state bootstrap 95% intervals for the random controls.

Equation (129) contains T mismatch terms, so

$$\left\|h_T^{\mathrm{amb}} - h_T^{\mathrm{res}}\right\| \le cT\epsilon^2 e^{\beta T\epsilon}. \tag{131}$$

The factor $T\epsilon^2$ accumulates the local extension mismatch, while $e^{\beta T\epsilon}$ accounts for amplification by later steps. This is an upper bound, not a predicted growth law: it does not determine mismatch signs, exploit cancellation, or model support-transition events explicitly.

Table 18: Three-way top-1 outcomes across horizon. B, A, and R denote the baseline, ambient, and residual-preserving tokens. Counts are out of 96.

T	all same	A only	R only	same new	different new
16	86	1	0	9	0
128	76	0	2	18	0
256	48	2	0	30	16