StiCAS: Compositional Activation Steering via Stiefel Manifold Coordinate Transport

Toan Doan
Applied Artificial Intelligence Initiative
Deakin University, Australia
toan.doan@deakin.edu.au

Thin Nguyen
Applied Artificial Intelligence Initiative
Deakin University, Australia
thin.nguyen@deakin.edu.au

Sunil Gupta
Applied Artificial Intelligence Initiative
Deakin University, Australia
sunil.gupta@deakin.edu.au

Abstract

Activation steering is a lightweight way to control pretrained generative models: a learned map inserted into the model's activations can steer generation toward target behavior without updating the model weights. While effective for single target behavior, existing steering methods remain brittle when multiple behaviors are targeted. A style intervention followed by a safety intervention can produce a different result than the same applied in the reverse order, as the second intervention is evaluated on activations already shifted by the first. We trace this failure to coordinate sharing: existing transport-based steering methods learn different concepts in the same activation coordinates, so their interventions interfere by construction. To address this, we introduce **StiCAS** (**Sti**efel **C**ompositional **A**ctivation **S**teering), a geometric steering framework that learns a separate orthogonal subspace for each concept. Each concept is steered by an affine transport restricted to its own subspace, and orthogonality between subspaces makes these transports non-interfering by construction. The concept frame is trained directly on the Stiefel manifold, so Riemannian optimization preserves the orthogonality required for composition throughout learning. This structure gives an exact commutativity guarantee: any sequential ordering of single-concept interventions reproduces the simultaneous multi-concept update. Across 45 concept pairs and three open models, StiCAS drives order-dependent interference exactly to zero and roughly doubles joint-concept success over the strongest activation-transport baseline.

1 Introduction

Activation steering controls a pretrained generative model without updating its weights: once learned, a lightweight map inserted into one or more layers nudges generation toward a target behavior. Controlling a single behavior is rarely enough. In practical settings, one may want to enforce a writing style while suppressing toxic completions, or maintain a domain-specific tone while preserving task relevance. This paper asks how multiple-behavior control can be achieved reliably.

Single-behavior steering is well studied. Directional and additive methods such as CAA [1], ITI-C [2] and RepE [3] steer along activation directions estimated from contrastive prompts, class means, probes, or principal components. Activation-transport methods such as Mean-ACT, Linear-ACT [4], and LinEAS [5] provide a more expressive alternative by fitting affine maps between baseline and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

target activation distributions. These methods show that small activation-space interventions can provide effective, data-light control, but they are primarily calibrated for one behavior at a time.

Composition exposes a different failure mode. A style intervention followed by a safety intervention usually does not match the result of applying safety first and style second. Worse, the second intervention no longer sees the activation distribution on which it was calibrated, but one already shifted by the first intervention. Thus, even if each intervention works in isolation, their sequential composition can be order-sensitive and miscalibrated.

We trace this brittleness to coordinate sharing. Existing transport-based steering methods learn different concepts in the same activation coordinates. When two concepts act on the same coordinates, each intervention can change the input space on which the other was calibrated. Composition is therefore order-sensitive by construction. Section 3 makes this obstruction precise with a simple affine example.

To address this problem, we propose **StiCAS**, which fixes this by learning concept coordinates. Rather than forcing every concept through the same neuron basis, StiCAS gives each concept its own orthogonal subspace of the activation space and learns an affine transport inside it. Because the subspaces are mutually orthogonal, changing one concept's coordinates leaves every other concept's coordinates untouched. For C concepts with r directions per concept, these subspaces form an orthonormal frame with Cr columns, a point on the Stiefel manifold.

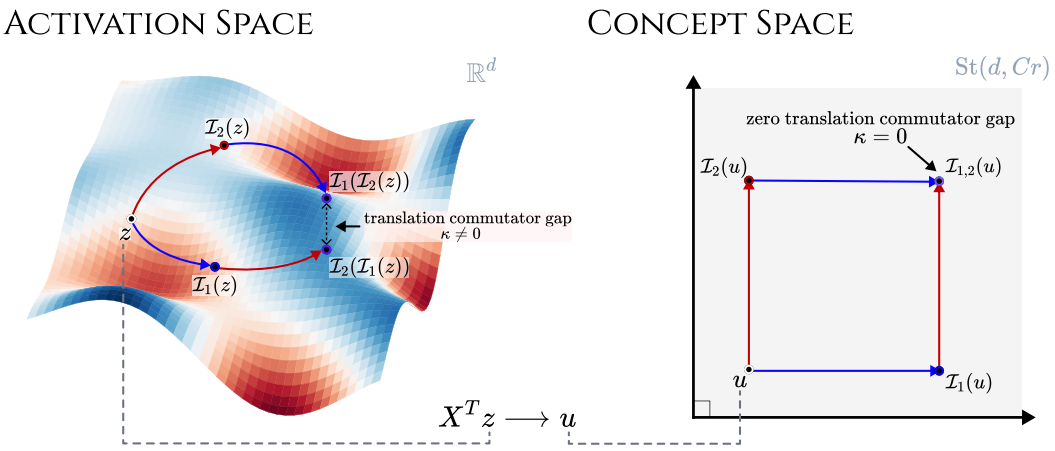

Figure 1: **The compositionality bottleneck and the StiCAS fix.** *Left:* In activation space, independently learned steering maps $\mathcal{I}_1$ and $\mathcal{I}_2$ act on shared coordinates; sequential composition is order-sensitive, leaving a nonzero mismatch κ between $\mathcal{I}_1(\mathcal{I}_2(z))$ and $\mathcal{I}_2(\mathcal{I}_1(z))$. *Right:* After projection to concept space, the transports act along orthogonal coordinate blocks, so composition commutes and the mismatch closes.

This parameterization yields an exact map-level compositionality guarantee: applying concept A then B produces the same activation as applying B then A, and both equal the simultaneous multi-concept update. The guarantee follows from the orthogonality constraint alone, independent of how the affine parameters are trained. Because the concept frame is optimized on the Stiefel manifold, $X^\top X = I$ holds after every optimizer step, so the guarantee holds at every iterate.

The empirical picture matches the geometry. Across concept pairs and layers, StiCAS drives a map-level measure of order-dependent interference to zero, while non-orthogonal activation-transport baselines retain nonzero interference. The same reduction translates to higher joint-concept success when several behaviors are steered together, and an orthogonality-penalty ablation shows the gain tracks orthogonality of the learned subspaces rather than the use of a shared training objective.

The paper contributes (i) a structural account of compositional failure in affine activation transport, attributable to coordinate sharing; (ii) StiCAS, a geometric steering parameterization that learns mutually orthogonal concept subspaces and performs affine coordinate transport inside each one; and (iii) a proof of exact map-level commutativity for the resulting single-concept interventions, together with empirical evidence that this guarantee improves multi-concept steering over activation-steering

baselines. Extensive experiments across two benchmark suites (with 45 concept pairs) and three open LLMs demonstrate that StiCAS consistently outperforms state-of-the-art activation-steering baselines, roughly doubling joint-concept success over the strongest activation-transport baseline.

2 Preliminaries

The three constants used throughout are: d, the activation dimension at this layer; C, the number of concepts the user wants to control; and r, the number of directions allocated to each concept (so the total rank of the intervention is Cr). The remaining notation is introduced where it first appears.

2.1 Activation steering as distribution matching

Let a frozen model produce an activation $z \in \mathbb{R}^d$ at a chosen layer; we write $Z_{\#}p$ for the pushforward of a prompt distribution p under this activation map. A baseline prompt distribution p then induces a baseline activation distribution $\mu := Z_{\#}p$, while a prompt distribution q_c for concept c (indexed in $[C] := \{1, \dots, C\}$) induces a target distribution $\nu_c := Z_{\#}q_c$. Activation steering seeks a lightweight map $\mathcal{I} : \mathbb{R}^d \to \mathbb{R}^d$, inserted into the forward pass, such that the pushed-forward baseline activations resemble the target concept distribution: $\mathcal{I}_{\#}\mu \approx \nu_c$. For a single concept this reduces to calibrating one intervention. The central difficulty studied in this paper is how to calibrate several such interventions so that they can be applied together without order-dependent interference.

2.2 Coordinate-wise affine transport

Within a single activation coordinate, the practical transport family used by Linear-ACT and LinEAS is the affine map

$$T(a) = \omega a + \beta, \qquad \omega \in \mathbb{R}_{++}, \;\; \beta \in \mathbb{R}, \tag{1}$$

where $\mathbb{R}_{++} := \{x \in \mathbb{R} : x > 0\}$ keeps T orientation-preserving. This map can be estimated either in closed form from sorted centered samples [4] (closed-form estimator in Appendix A.2) or end-to-end with a univariate distributional loss $\mathcal{D}$ between the transported and target activation marginals. Given n samples $\{u_i\}_{i=1}^n$ from the transported marginal and $\{v_i\}_{i=1}^n$ from the target marginal, with order statistics $u_{(1)} \le \cdots \le u_{(n)}$ and $v_{(1)} \le \cdots \le v_{(n)}$, we use

$$\mathcal{D}(u, v) := \frac{1}{n} \sum_{i=1}^{n} \big(u_{(i)} - v_{(i)}\big)^2, \tag{2}$$

the squared 1-D Wasserstein-2 distance between empirical distributions, evaluated in closed form via sorting [6, 5]. This family is sample-efficient, but it acts in a fixed coordinate system: when several concepts are calibrated independently in such shared coordinates, the resulting affine maps need not compose (Section 3.1).

2.3 Orthonormal frames and the Stiefel manifold

A point on the Stiefel manifold is an ordered collection of orthonormal directions in $\mathbb{R}^d$. StiCAS groups these directions into concept blocks: for C concepts and per-concept rank r, we use

$$X = [X_1, \dots, X_C] \in \mathrm{St}(d, Cr), \qquad X_c \in \mathbb{R}^{d \times r}, \tag{3}$$

where $\mathrm{St}(d, m) := \{X \in \mathbb{R}^{d \times m} : X^\top X = I_m\}$. The constraint $X^\top X = I_{Cr}$ makes the concept subspaces mutually orthogonal: $X_c^\top X_c = I_r$ and $X_c^\top X_{c'} = 0$ for $c \neq c'$. The per-concept rank r controls how much within-concept variation each subspace can capture. At $r = 1$ each block collapses to a single direction $x_c \in \mathbb{R}^d$ ($m = C$), recovering the one-direction-per-concept setting of prior work. At $r > 1$ the block X_c spans an r-dimensional subspace, and the per-concept transport becomes r scalar affine maps acting in parallel; larger r is more expressive but requires optimizing a larger Stiefel frame.

To optimize over the manifold, we need a tangent space and a way to project onto it. At $X \in \mathrm{St}(d, Cr)$, the tangent space is

$$T_X \mathrm{St}(d, Cr) = \{\xi \in \mathbb{R}^{d \times Cr} : \xi^\top X + X^\top \xi = 0\}. \tag{4}$$

Writing $\mathrm{symm}(A) := \frac{1}{2}(A + A^\top)$ for the symmetric part of a matrix, the Riemannian gradient under the Euclidean metric of any $f : \mathrm{St}(d, Cr) \to \mathbb{R}$ is the projection of the Euclidean gradient $G := \nabla f(X)$ onto (4):

$$\mathrm{grad}\, f(X) = G - X\ \mathrm{symm}\big(X^\top G\big). \tag{5}$$

A retraction R_X then maps the resulting tangent step back to $\mathrm{St}(d, Cr)$, so orthogonality is preserved by every update.

3 Stiefel Compositional Activation Steering

We now introduce StiCAS. We first expose the core obstruction: affine steering maps become order-sensitive when independently learned concepts act in shared activation coordinates (Section 3.1). StiCAS resolves this by learning orthogonal concept subspaces, transporting activations inside each concept block, and lifting the resulting correction back to activation space. This project-transport-lift parameterization makes single-concept interventions commute exactly (Section 3.4). We present the method at a fixed activation layer through Section 3.4, then restore the multi-layer training and inference form in Section 3.5.

3.1 Problem setup and the compositionality bottleneck

Multi-concept activation steering aims to learn interventions $\mathcal{I}_1, \ldots, \mathcal{I}_C$, each individually calibrated so that $(\mathcal{I}_c)_{\#}\mu \approx \nu_c$, and to apply two or more of them together at inference. The natural construction applies them sequentially, $\mathcal{I}_{c_2}(\mathcal{I}_{c_1}(z))$, and asks whether the result is independent of the order. A scalar example below shows it is not.

Fix a single activation coordinate and consider two independently learned affine maps $T_1(a) = \omega_1 a + \beta_1$ and $T_2(a) = \omega_2 a + \beta_2$. Direct computation gives $T_2(T_1(a)) = \omega_2\omega_1 a + (\omega_2\beta_1 + \beta_2)$ and $T_1(T_2(a)) = \omega_1\omega_2 a + (\omega_1\beta_2 + \beta_1)$. The scale terms commute, but the translation terms do not: each is scaled by the *other* map. The two compositions agree if and only if $(\omega_2 - 1)\beta_1 = (\omega_1 - 1)\beta_2$, which holds only by coincidence when (ω_1, β_1) and (ω_2, β_2) are estimated independently from two concept datasets. Sequential composition is therefore order-sensitive in general (see Appendix A.4 for special cases).

Even if an order is fixed, a second failure persists: T_2 is calibrated against μ but, in the composed map, receives the shifted input distribution $(T_1)_{\#}\mu \neq \mu$, so $(T_2)_{\#}(T_1)_{\#}\mu \not\approx \nu_2$. Thus the second concept objective can be invalidated by the first intervention. Both failures share a common structural cause: independently learned maps reuse the *same* coordinates. This gives the design requirement that drives the rest of the section: a compositional steering method should allocate different concepts to mutually orthogonal coordinate blocks by construction.

3.2 Orthogonal concept coordinates

The bottleneck of Section 3.1 shows how composition fails when concepts share coordinates. StiCAS therefore learns coordinates before learning how to transport them.

Concretely, StiCAS gives each concept its own subspace of the activation space. For concept c, this subspace is represented by a block of r orthonormal directions, denoted $X_c \in \mathbb{R}^{d\times r}$. Stacking the C concept blocks side by side gives the *concept frame* $X = [X_1, \ldots, X_C] \in \mathrm{St}(d, Cr)$, a point on the Stiefel manifold of orthonormal frames in $\mathbb{R}^d$. Projecting an activation z onto X separates it into one coordinate block per concept,

$$u := X^\top z \in \mathbb{R}^{Cr}, \qquad u_c := X_c^\top z \in \mathbb{R}^r, \tag{6}$$

and the Stiefel constraint $X_{c'}^\top X_c = 0$ for $c \neq c'$ means that changing one concept's block leaves every other concept's block untouched. Figure 1 illustrates the change of viewpoint: in activation space, independently learned maps form an order-dependent loop; in concept space, the transports act along orthogonal coordinate blocks, and the loop closes.

For simplicity we use the same per-concept rank r for every concept, so the total frame lives on $\mathrm{St}(d, Cr)$. With per-concept ranks $r_1, \ldots, r_C$ instead, the frame becomes $X = [X_1, \ldots, X_C] \in \mathrm{St}(d, \sum_c r_c)$ with $X_c \in \mathbb{R}^{d\times r_c}$, and orthogonality $X_{c'}^\top X_c = 0$ still removes cross-talk across

concepts. This lets concepts whose activation subspace is higher-dimensional receive more directions, at the cost of one extra hyperparameter per concept; we leave this for future work.

3.3 Coordinate transport and the forward intervention

Inside each concept block we apply the same affine family used by Linear-ACT [4] and LinEAS [5], but per-coordinate within the block. For each concept c we introduce trainable affine parameters

$$\omega_c \in \mathbb{R}^r_{++}, \qquad \beta_c \in \mathbb{R}^r, \qquad \Omega_c := \mathrm{diag}(\omega_c) \in \mathbb{R}^{r\times r}, \tag{7}$$

and stack them into block-diagonal forms acting on the concept coordinate space:

$$\Omega := \mathrm{blkdiag}(\Omega_1, \ldots, \Omega_C) \in \mathbb{R}^{Cr\times Cr}, \qquad \beta := [\beta_1; \ldots; \beta_C] \in \mathbb{R}^{Cr}. \tag{8}$$

At inference, the user controls each concept independently with a single number. We collect these numbers into a strength vector $\lambda \in [0,1]^C$, where λ_c is the dial for concept c. Because each concept owns r coordinates, we lift these concept-level strengths to the coordinate space by repeating λ_c across the r directions of block c:

$$\Lambda_\lambda := \mathrm{blkdiag}(\lambda_1 I_r, \ldots, \lambda_C I_r) \in \mathbb{R}^{Cr\times Cr}. \tag{9}$$

Λ_λ represents λ at the coordinate granularity: one entry of λ governs all r directions of one concept.

Project-transport-lift. Given activation z, StiCAS computes the post-intervention activation z^+ in three steps (project, transport, lift):

$$u = X^\top z, \quad \tilde{u} = \Omega u + \beta, \quad z^+ = z + X\,\Lambda_\lambda(\tilde{u} - u). \tag{10}$$

In other words, StiCAS changes only the coordinates of active concepts (those with $\lambda_c \neq 0$) and leaves the rest of the activation unchanged: the update is confined to $\mathrm{span}(X)$, and components of z orthogonal to the learned concept frame pass through as is. Concept c contributes at most a rank-r update through its block X_c, so the simultaneous update has total rank at most Cr and the single-concept update has rank at most r (rank one when $r = 1$). The vector λ is the inference-time control knob: setting it to the c-th standard basis vector $e_c \in \mathbb{R}^C$ activates only concept c, setting several entries activates several concepts in a single forward pass, and $\lambda = 0$ recovers the original model. The equivalent matrix-affine form, which makes the rank-at-most-Cr structure explicit, is derived in Appendix A.3.

Single-concept operator. Setting $\lambda = e_c$ in (10) defines the single-concept intervention used in the next subsection,

$$\mathcal{I}_c(z) := z + X_c\big(T_c(u_c) - u_c\big), \qquad T_c(u_c) := \Omega_c u_c + \beta_c, \qquad u_c = X_c^\top z. \tag{11}$$

For $r = 1$, $X_c = x_c \in \mathbb{R}^d$ and (11) reduces to the scalar form $\mathcal{I}_c(z) = z + x_c\,(T_c(x_c^\top z) - x_c^\top z)$ used in prior activation-steering work.

3.4 Compositionality guarantee

The construction above makes a strong promise: applying concept c changes only the coordinate block $X_c^\top z$. Since $X_{c'}^\top X_c = 0$ for $c \neq c'$, every other concept coordinate block is preserved, so the order of single-concept interventions does not matter. We turn this observation into a lemma and a theorem.

Definition 3.1 (Per-concept transport residual)**.** For $z \in \mathbb{R}^d$, the *per-concept transport residual* for concept c is $\Delta_c(z) := T_c(X_c^\top z) - X_c^\top z \in \mathbb{R}^r$, so the single-concept intervention factors as $\mathcal{I}_c(z) = z + X_c\,\Delta_c(z)$.

Definition 3.2 (Simultaneous C-concept intervention)**.** For $z \in \mathbb{R}^d$, the *simultaneous C-concept intervention* is

$$\mathcal{I}_{[C]}(z) := z + \sum_{c=1}^{C} X_c\,\Delta_c(z), \tag{12}$$

which equals (10) with $\lambda = \mathbf{1}$.

Lemma 3.1 (No cross-talk across concepts)**.** *Let* $X = [X_1, \ldots, X_C] \in \mathrm{St}(d, Cr)$*. For any* $c \neq c'$*,*

$$X_{c'}^\top \mathcal{I}_c(z) = X_{c'}^\top z \qquad \textit{for all } z \in \mathbb{R}^d.$$

Proof. $X_{c'}^\top \mathcal{I}_c(z) = X_{c'}^\top z + (X_{c'}^\top X_c)\, \Delta_c(z) = X_{c'}^\top z$, since $X_{c'}^\top X_c = 0$ by the Stiefel constraint on X. □

Theorem 3.1 (Sequential composition equals simultaneous intervention)**.** *Let* $X = [X_1, X_2] \in \mathrm{St}(d, 2r)$*. Then for all* $z \in \mathbb{R}^d$*,*

$$\mathcal{I}_2(\mathcal{I}_1(z)) = \mathcal{I}_1(\mathcal{I}_2(z)) = \mathcal{I}_{\{1,2\}}(z), \qquad \mathcal{I}_{\{1,2\}}(z) = z + X_1\, \Delta_1(z) + X_2\, \Delta_2(z).$$

Proof. Unfolding (11), $\mathcal{I}_2(\mathcal{I}_1(z)) = \mathcal{I}_1(z) + X_2\, \Delta_2(\mathcal{I}_1(z))$. Since Δ_2 depends on its argument only through $X_2^\top$, and Lemma 3.1 gives $X_2^\top \mathcal{I}_1(z) = X_2^\top z$, we have $\Delta_2(\mathcal{I}_1(z)) = \Delta_2(z)$. Substituting and expanding $\mathcal{I}_1(z) = z + X_1\, \Delta_1(z)$ yields $\mathcal{I}_{\{1,2\}}(z)$. The reversed order follows symmetrically using Lemma 3.1 with $c = 2$, $c' = 1$. □

Remark 3.1 (General C and the special case $r = 1$)*.* The same argument extends to any $C \geq 2$: all single-concept maps commute pairwise, and any sequential ordering equals $\mathcal{I}_{[C]}$ in (12); the inductive proof is in Appendix A.1. When $r = 1$, blocks reduce to single directions $X_c = x_c$ and the statement specializes to the one-direction-per-concept theorem.

Remark 3.2 (Why coordinate-wise maps fail)*.* When $\mathcal{I}_1$ and $\mathcal{I}_2$ act on the same scalar coordinate (as in Linear-ACT), $\Delta_2(\mathcal{I}_1(z))$ depends on $\mathcal{I}_1(z)$ through that same coordinate, not an orthogonal one, so the cross-term does not vanish and Lemma 3.1 fails.

The theorem is a statement about the map itself, irrespective of a particular optimizer or dataset. It proves exact map-level commutativity for any frame X that satisfies the Stiefel constraint. Whether StiCAS-trained interventions improve downstream generation remains an empirical question, which we examine in Section 4.

Map-level diagnostic. The same orthogonality argument cancels an affine-level diagnostic of compositional interference, giving a metric we can read off the trained intervention parameters with no decoding. Writing each single-concept intervention in activation-space affine form $\mathcal{I}_c(z) = A_c z + b_c$ and expanding the order swap $\mathcal{I}_j(\mathcal{I}_i(z)) - \mathcal{I}_i(\mathcal{I}_j(z))$ produces a matrix-product part $(A_j A_i - A_i A_j)\, z$ plus a translation part that does not vanish in general; this translation part is our diagnostic.

Definition 3.3 (Translation commutator gap)**.** The *translation commutator gap* between concepts i and j is $\kappa_{ij} := (A_j - I)\, b_i - (A_i - I)\, b_j \in \mathbb{R}^d$.

Definition 3.4 (Normalized commutator gap)**.** The *normalized commutator gap* is the strength-normalized form of κ_{ij},

$$\bar{\kappa}_{ij} \;:=\; \frac{\|\kappa_{ij}\|_1}{\sigma_i + \sigma_j + \varepsilon}, \qquad \sigma_c \;:=\; \|\omega_c - \mathbf{1}\|_1 + \|\beta_c\|_1, \tag{13}$$

where ε is a small numerical floor. Orthogonality forces $\bar{\kappa} = 0$ for StiCAS by Lemma 3.1. Per-layer aggregation and the full cancellation argument are in Appendix A.8.

3.5 Training and inference

We now restore layer indices. For each intervened layer $\ell \in L$, StiCAS learns a frame $X_\ell \in \mathrm{St}(d_\ell, Cr)$ and per-concept affine block maps $T_{\ell,c}$. Given baseline activations $z_\ell \sim \mu_\ell$ and concept activations $z_\ell^{(c)} \sim \nu_{\ell,c}$, we train by matching transported baseline coordinates to target coordinates inside each concept block. Full multi-layer notation is in Appendix A.5.

Distributional loss. StiCAS uses a scalar sorted-quantile Wasserstein loss $\mathcal{D}$ similar to LinEAS [5] as mentioned in (2), applied coordinate-wise inside each block and summed across the r directions: $\mathcal{D}_r(u, v) := \sum_{j=1}^r \mathcal{D}(u_j, v_j)$ with $\mathcal{D}_1 = \mathcal{D}$. Setting $\lambda = e_c$ in (10), the per-layer loss is

$$\mathcal{L}_\ell = \sum_{c=1}^{C} \mathcal{D}_r\Big(T_{\ell,c}(X_{\ell,c}^\top z_\ell),\; X_{\ell,c}^\top z_\ell^{(c)}\Big), \qquad \mathcal{L} = \sum_{\ell \in L} \mathcal{L}_\ell. \tag{14}$$

Training jointly learns the concept subspaces X_ℓ and the affine transports $T_{\ell,c}$ from unpaired baseline and concept batches.

Riemannian optimization. We optimize the affine parameters with a standard Euclidean optimizer, storing $\theta_{\ell,c} := \log \omega_{\ell,c} \in \mathbb{R}^r$ component-wise to enforce $\omega_{\ell,c} \in \mathbb{R}^r_{++}$. Each frame X_ℓ is updated by the tangent-projected Riemannian gradient (5) (specialized to layer ℓ) followed by a retraction onto $\mathrm{St}(d_\ell, Cr)$; we use a polar retraction by default and ablate the geodesic exponential map (Appendix A.7). The retraction preserves $X_\ell^\top X_\ell = I_{Cr}$ at every update, so Lemma 3.1 and Theorem 3.1 apply at every iterate.

Inference. At inference, each intervened layer applies (10) with a user-chosen strength vector $\lambda \in [0, 1]^C$ (as discussed in Section 3.3). By Theorem 3.1, ordering does not matter: the simultaneous update equals any sequential ordering of the corresponding single-concept maps. Pseudocode for training and inference is in Appendix B.8.

4 Experiments

We evaluate StiCAS to answer three questions. First, does a (map-level) measure of order-dependent interference predict downstream multi-concept success? Second, does StiCAS improve joint-concept steering while retaining single-concept calibration? Third, is the gain driven by orthogonality of the learned concept subspaces rather than by joint training alone?

4.1 Experimental setup

We evaluate StiCAS on Gemma-2-2B, Gemma-2-9B [7], and Llama-3-8B [8] using two suites: a five-concept LinEAS replication suite for map-level diagnostics (***Suite 1***, Appendix C.4) and a main style–toxicity benchmark with $\binom{10}{2} = 45$ concept pairs (***Suite 2***). We compare independently trained sequential composition (***Setting A***, StiCAS-S) with jointly trained single-pass composition under a joint frame $X_\ell \in \mathrm{St}(d_\ell, Cr)$ activated via (10) (***Setting B***, StiCAS-J). Baselines split into affine transport methods (Linear-ACT [4], LinEAS [5]) and additive or projected controls (Mean-ACT, AddCAA [1], RepE [3], ITI-C [2]). We report joint-concept success probability $p(c_1, c_2)$, the toxicity-attribute score on toxicity-paired runs (ToxAmp), the normalized commutator gap $\bar{\kappa}$, and MMLU (5-shot) [9] as a check that the intervention does not erode general utility. Full datasets, prompts, judging protocol, hyperparameters, composition protocols, the complementary fluency-side-effect view, and confidence intervals are in Appendices B–C. In Suite 2 we steer toward *more* toxic text, so a higher ToxAmp means stronger toxicity steering. Non-toxic text usually differs from toxic text in topic and register as well, so a toxic-to-non-toxic transport could change style along with toxicity and confound the style concepts; amplifying toxicity within the same corpus keeps the two attributes separate. Detoxification results are in Appendix C.3. Figure 2 gives a direct map-level check on Suite 1 before decoding.

4.2 Benchmark evaluation: Calibration and Compositional behavior

Table 1 reports the main Suite-2 benchmark, aggregated over all $\binom{10}{2} = 45$ concept pairs and 5 random seeds on the three models. Affine transport baselines calibrate single concepts well but lose ground under composition: LinEAS-J improves over sequential LinEAS yet still trails StiCAS-J on every model. Additive and projected baselines avoid order effects but reach low joint $p(c_1, c_2)$, so order-invariance on its own does not deliver multi-concept success. StiCAS-J is the only method that combines calibrated affine transport with structural composition: it attains $\bar{\kappa} = 0$ exactly as suggested by Lemma 3.1 and the highest joint $p(c_1, c_2)$ on every model. Figure 4 shows the same separation at the individual-run (method, model, pair) level: $\bar{\kappa}$ predicts joint success most strongly in the joint setting, while the additive cluster remains at low joint $p(c_1, c_2)$ despite $\bar{\kappa} = 0$, so a vanishing commutator without calibrated transport is not enough. Every $p(c_1, c_2)$ above comes from a single LLM judge, so a judge that misreads author style, or favours one method's outputs, could produce the same gaps. We therefore check both that the judge recognizes style and that the ranking survives a change of judge. The Qwen2.5-7B judge [10] separates held-out target-author passages from other authors and unsteered text with AUROC 0.873, and rescoring the same generations with Qwen2.5-32B-Instruct [10] and gpt-oss-120b [11] leaves StiCAS-S and StiCAS-J as the two best methods on every model (Appendix B.4).

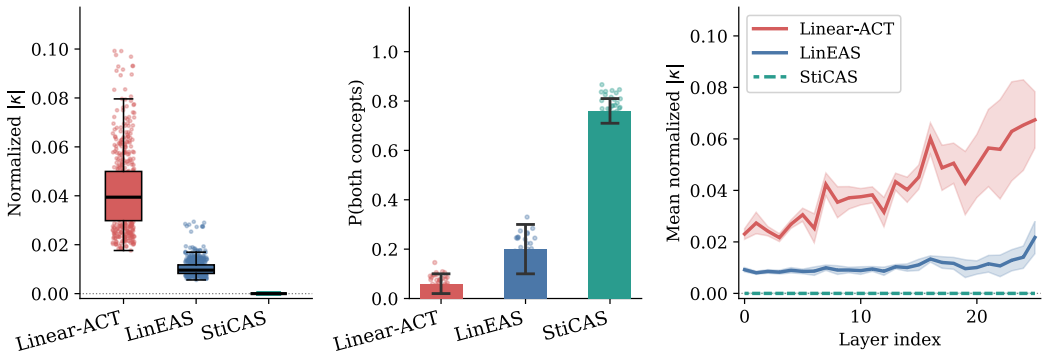

Figure 2: Map-level diagnostic on Suite 1 (Gemma-2-2B; ten concept pairs from *day, night, elephant, football, fishing*; 5 seeds; 26 intervened layers). *Left:* per-pair distribution of the normalized commutator gap $\bar{\kappa}$. *Middle:* mean joint-presence rate $p(c_1, c_2)$ per method (fraction of the 200 continuations per pair where both judge probabilities exceed 0.5, averaged over the ten pairs); error bars over seeds. *Right:* layer-wise mean $\bar{\kappa}$ vs. layer index, one curve per method, shaded band over seeds.

Table 1: Main benchmark on Suite 2 (45 pairs, 5 seeds). Top block: Setting A (per-concept training, sequential composition); bottom block: Setting B (joint training, single forward pass). For each model we report joint-concept success $p(c_1, c_2)$, the normalized commutator gap $\bar{\kappa}$, the toxicity-attribute score ToxAmp (on toxicity-paired runs; we steer toward more toxic text to avoid topic and register confounders, see Section 4.1), and MMLU (5-shot) to capture the side-effect of the intervention on general utility. Bold = best $p(c_1, c_2)$ per model. "$0^\dagger$" marks pure-additive baselines ($A_c = I$) for which $\bar{\kappa} = 0$ trivially by Definition 3.3. Single-concept accuracy, retraction splits, and 95% CIs are in Appendix C.

	Gemma-2-2B				Gemma-2-9B				Llama-3-8B			
Method	$p(c_1, c_2)\uparrow$	$\bar{\kappa}\downarrow$	ToxAmp↑	MMLU↑	$p(c_1, c_2)\uparrow$	$\bar{\kappa}\downarrow$	ToxAmp↑	MMLU↑	$p(c_1, c_2)\uparrow$	$\bar{\kappa}\downarrow$	ToxAmp↑	MMLU↑
AddCAA	0.108	$0^\dagger$	0.075	52.94	0.133	$0^\dagger$	0.115	69.95	0.121	$0^\dagger$	0.095	64.96
RepE	0.124	$0^\dagger$	0.070	52.61	0.149	$0^\dagger$	0.110	69.61	0.137	$0^\dagger$	0.090	64.61
Mean-ACT	0.137	$0^\dagger$	0.140	51.91	0.162	$0^\dagger$	0.180	68.95	0.150	$0^\dagger$	0.160	64.02
Linear-ACT	0.051	0.045	0.460	51.98	0.076	0.047	0.500	69.12	0.064	0.046	0.480	64.00
LinEAS	0.231	0.011	0.610	51.54	0.256	0.011	0.650	68.51	0.244	0.012	0.630	63.56
LinEAS+L1	0.226	0.011	0.620	51.03	0.251	0.011	0.660	67.97	0.239	0.012	0.640	62.87
StiCAS-S	**0.515**	0.007	**0.680**	51.27	**0.540**	0.006	**0.720**	68.33	**0.528**	0.007	**0.700**	63.04
ITI-C	0.115	0.038	0.180	51.63	0.140	0.040	0.220	68.58	0.128	0.039	0.200	63.64
LinEAS-J	0.275	0.018	0.645	51.00	0.300	0.018	0.685	67.98	0.288	0.019	0.665	62.95
StiCAS-J	**0.575**	0.000	**0.722**	51.09	**0.600**	0.000	**0.762**	67.96	**0.588**	0.000	**0.742**	62.90

4.3 The role of Orthogonality in Compositional generalization

The gap between LinEAS-J and StiCAS-J in Table 1 shows that joint training by itself does not close the compositionality gap. To isolate the role of geometry, we train otherwise identical StiCAS models while varying a soft orthogonality penalty between concept subspaces, $\mathcal{L}_{\text{ortho}} = \lambda \sum_{i \neq j} \|X_i^\top X_j\|_F^2$, which recovers Setting A at $\lambda = 0$ and Setting B at $\lambda = \infty$. On Gemma-2-9B with Shakespeare $\times$ Hemingway, increasing the penalty drives $\bar{\kappa}$ smoothly to zero while joint $p(c_1, c_2)$ rises (Figure 3, left). The gain therefore tracks enforcement of orthogonal concept subspaces rather than the use of a shared objective alone. StiCAS-S, the $\lambda = 0$ endpoint, already attains a small $\bar{\kappa}$ because independently learned frames are nearly orthogonal in practice; the explicit constraint makes this exact and adds a further ~ 0.04 joint-success gain on this pair. Full sweeps across concept pairs and retractions are in Appendix D.4. The per-concept rank sweep (Figure 3, middle) shows that joint $p(c_1, c_2)$ saturates around $r = 32$, and Figure 3 (right) shows StiCAS-S matches LinEAS at one quarter the training data. Layer, retraction, and no-retraction ablations in Appendix D reach the same conclusion. A small Suite-1 generation gallery is in Appendix E.

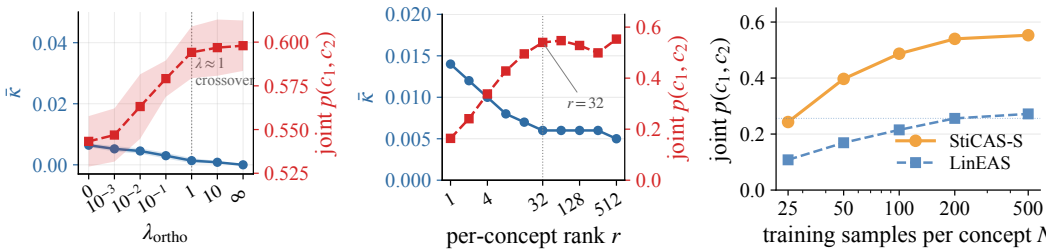

Figure 3: Throughout: $\bar{\kappa}$ in blue (solid, circle marker, left y-axis); joint $p(c_1, c_2)$ in red (dashed, square marker, right y-axis where dual). *Left:* orthogonality interpolation on Gemma-2-9B, Shakespeare $\times$ Hemingway (mean over 5 seeds $\times$ 2 retractions; band is $\pm$std). Vertical dotted line at $\lambda = 1$. *Middle:* per-concept rank sweep on Gemma-2-9B (mean over 5 seeds and both retractions). StiCAS-S $\bar{\kappa}$ and $p(c_1, c_2)$ vs. per-concept rank r. *Right:* sample-size sweep on the Gemma-2-9B 5-pair Suite-2 subset (5 seeds): StiCAS-S (warm) vs. LinEAS (blue, dashed) joint $p(c_1, c_2)$ vs. training-sample count N. StiCAS-S at N=50 already exceeds LinEAS at N=200.

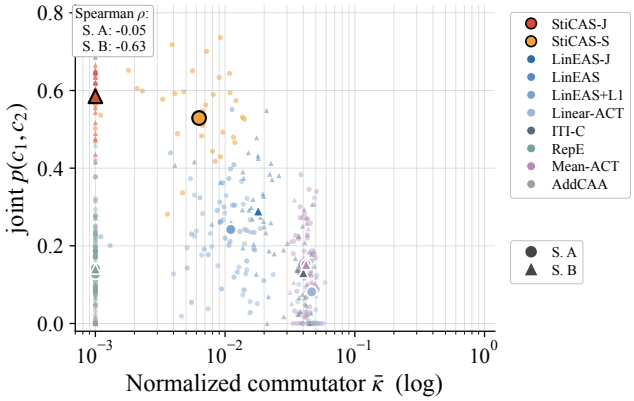

Figure 4: Scatter of $\bar{\kappa}$ (log x-axis) vs. joint $p(c_1, c_2)$ on Suite 2; one cluster centre per (method, setting), with up to 35 sampled (model, pair, seed) points behind each centre. Color encodes method (warm = StiCAS, blue = transport baselines, muted = additive baselines); marker shape encodes Setting (circle = A, triangle = B). Spearman ρ between $\bar{\kappa}$ and $p(c_1, c_2)$ per Setting in the legend.

5 Related Work

Activation steering controls language models by modifying internal activations rather than updating weights. Steering directions are estimated from contrastive prompts [1], word embeddings [12], principal components [3], or activation addition [13], with extensions for conditional routing [14], rotational parameterizations [15], preference vectors [16, 17], and editing toolkits [18]. Activation-transport methods are more expressive, fitting affine maps between source and target activation distributions; LinEAS learns these end-to-end via a distributional loss. All operate in the original activation coordinates.

Coordinate choice matters because model behaviors often occupy structured low-dimensional subspaces: a single refusal direction [19] within a broader cone [20], steering as rotation [15], and a behavior-subspace view that also supports hallucination control [21] and trojan attacks [22]. LoReFT [23] also edits representations inside an orthonormal low-rank subspace, trained with task supervision for a single downstream task, and does not address composing several such edits. StiCAS extends the subspace view from single-behavior control to composition: each concept gets its own orthogonal subspace, and the Stiefel constraint prevents one concept's transport from disturbing another's coordinates.

Multi-attribute control is brittle: combined steering vectors weaken individual controls, and composed interventions are order-sensitive [24, 25]. MAT-Steer [26] learns sparse, token-targeted steering vectors and reduces conflicts between attributes with a soft orthogonality penalty. MSRS [27]

gives each attribute a private subspace plus one subspace shared by all attributes, combined with learned weights. Conditional Activation Steering [14] routes per input, AlphaEdit [28] uses null-space constraints, and distributional alignment uses optimal transport [29]. None of these methods guarantees that applying interventions one at a time gives the same map as applying them together. We obtain this guarantee for block-local affine transports by keeping the concept blocks exactly orthogonal during training (Lemma 3.1, Theorem 3.1), while each transport is still fitted by distribution matching.

6 Conclusion and Limitations

We address compositional activation steering by learning orthogonal concept subspaces and applying affine transport inside each, with the Stiefel constraint preserving orthogonality throughout training. This yields an exact map-level guarantee: single-concept interventions commute, and any sequential ordering equals the simultaneous multi-concept update. Empirically, StiCAS drives order-dependent interference exactly to zero and roughly doubles joint-concept success over the strongest baseline; an ablation shows the gain rises with subspace orthogonality rather than with shared training. Two open questions stand out: scaling to many correlated concepts, and deployment-friendly approximations that preserve commutativity without per-layer Stiefel optimization.

Acknowledgments and Disclosure of Funding

This research was partially supported by the Wellcome Trust [grant number: 303030/Z/23/Z], the National Health and Medical Research Council (NHMRC) Centre of Research Excellence in Depression Treatment Precision (grant number: 2024796) and the NHMRC Synergy Grant (grant number: 2026505). Toan Doan received the Deakin University Postgraduate Research Scholarship (DUPR).

Although evaluated on general safety datasets from public domain, the proposed multi-behavior steering method can be applied to advance the foundational safety, alignment, and reliable deployment of large language models in sensitive domains, including digital mental health and supportive AI technologies.

References

[1] N. Rimsky, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. Turner, "Steering Llama 2 via Contrastive Activation Addition," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[2] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg, "Inference-Time Intervention: Eliciting Truthful Answers from a Language Model," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[3] A. Zou *et al.*, "Representation Engineering: A Top-Down Approach to AI Transparency," arXiv:2310.01405, 2023.

[4] P. Rodriguez, A. Blaas, M. Klein, L. Zappella, N. Apostoloff, M. Cuturi, and X. Suau, "Controlling Language and Diffusion Models by Transporting Activations," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[5] P. Rodriguez, M. Klein, E. Gualdoni, V. Maiorca, A. Blaas, L. Zappella, M. Cuturi, and X. Suau, "LinEAS: End-to-end Learning of Activation Steering with a Distributional Loss," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[6] J. Rabin, G. Peyré, J. Delon, and M. Bernot, "Wasserstein barycenter and its application to texture mixing," in *Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, 2012, pp. 435–446.

[7] Gemma Team *et al.*, "Gemma 2: Improving Open Language Models at a Practical Size," Oct. 2024.

[8] A. Grattafiori *et al.*, "The Llama 3 Herd of Models," Nov. 2024.

[9] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring Massive Multitask Language Understanding," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[10] Qwen Team *et al.*, "Qwen2.5 Technical Report," Dec. 2024.

[11] OpenAI, "gpt-oss-120b & gpt-oss-20b model card," arXiv:2508.10925, 2025.

[12] C. Han, J. Xu, M. Li, Y. Fung, C. Sun, N. Jiang, T. Abdelzaher, and H. Ji, "Word Embeddings Are Steers for Language Models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[13] A. M. Turner, L. Thiergart, G. Leech, D. Udell, J. J. Vazquez, U. Mini, and M. MacDiarmid, "Steering Language Models With Activation Engineering," 2024.

[14] B. W. Lee, I. Padhi, K. N. Ramamurthy, E. Miehling, P. Dognin, M. Nagireddy, and A. Dhurandhar, "Programming Refusal with Conditional Activation Steering," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[15] H. M. Vu and T. M. Nguyen, "Angular Steering: Behavior Control via Rotation in Activation Space," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[16] Y. Cao, T. Zhang, B. Cao, Z. Yin, L. Lin, F. Ma, and J. Chen, "Personalized Steering of Large Language Models: Versatile Steering Vectors Through Bi-directional Preference Optimization," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[17] T. Doan, U. Le, and T. Nguyen, "Causal Activation Steering via Sparse Mediation," in *Proceedings of the Association for Computational Linguistics: EACL 2026 (Findings)*, V. Demberg, K. Inui, and L. Marquez, Eds., 2026, pp. 1079–1097.

[18] Z. Xu, S. Wang, K. Xu, H. Xu, M. Wang, X. Deng, Y. Yao, G. Zheng, H. Chen, and N. Zhang, "EasyEdit2: An Easy-to-use Steering Framework for Editing Large Language Models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

[19] A. Arditi, O. Obeso, A. Syed, D. Paleka, N. Panickssery, W. Gurnee, and N. Nanda, "Refusal in Language Models Is Mediated by a Single Direction," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[20] T. Wollschläger, J. Elstner, S. Geisler, V. Cohen-Addad, S. Günnemann, and J. Gasteiger, "The Geometry of Refusal in Large Language Models: Concept Cones and Representational Independence," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

[21] S. Park, X. Du, M.-H. Yeh, H. Wang, and Y. Li, "Steer LLM Latents for Hallucination Detection," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

[22] H. Wang and K. Shu, "Trojan Activation Attack: Red-Teaming Large Language Models using Activation Steering for Safety-Alignment," in *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2024.

[23] Z. Wu, A. Arora, Z. Wang, A. Geiger, D. Jurafsky, C. D. Manning, and C. Potts, "ReFT: Representation Finetuning for Language Models," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[24] J. Braun, C. Eickhoff, D. Krueger, S. A. Bahrainian, and D. Krasheninnikov, "Understanding (Un)Reliability of Steering Vectors in Language Models," in *Proceedings of the International Conference on Learning Representations Workshops (ICLR W)*, 2025.

[25] D. Tan, D. Chanin, A. Lynch, B. Paige, D. Kanoulas, A. Garriga-Alonso, and R. Kirk, "Analysing the Generalisation and Reliability of Steering Vectors," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[26] D. Nguyen, A. Prasad, E. Stengel-Eskin, and M. Bansal, "Multi-Attribute Steering of Language Models via Targeted Intervention," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

[27] X. Jiang, L. Zhang, J. Zhang, Q. Yang, G. Hu, D. Wang, and L. Hu, "MSRS: Adaptive Multi-Subspace Representation Steering for Attribute Alignment in Large Language Models," Aug. 2025.

[28] J. Fang, H. Jiang, K. Wang, Y. Ma, J. Shi, X. Wang, X. He, and T.-S. Chua, "AlphaEdit: Null-Space Constrained Knowledge Editing for Language Models," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[29] I. Melnyk, Y. Mroueh, B. Belgodere, M. Rigotti, A. Nitsure, M. Yurochkin, K. Greenewald, J. Navratil, and J. Ross, "Distributional Preference Alignment of LLMs via Optimal Transport," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[30] P.-A. Absil, *Optimization Algorithms on Matrix Manifolds*. Princeton: Princeton University Press, 2008.

[31] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models," in *Proceedings of the Findings of the Association for Computational Linguistics (EMNLP)*, T. Cohn, Y. He, and Y. Liu, Eds., 2020, pp. 3356–3369.

[32] L. Hanu and Unitary team, "Detoxify," GitHub. https://github.com/unitaryai/detoxify, 2020.

[33] Z. Li, S. Sajadmanesh, J. Li, and L. Lyu, "StelLA: Subspace Learning in Low-rank Adaptation using Stiefel Manifold," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[34] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman, "Nuanced metrics for measuring unintended bias with real data for text classification," in *Companion Proceedings of the 2019 World Wide Web Conference*, 2019, pp. 491–500.

A Extended proof notes

A.1 Generalization to C concepts

The proof of Theorem 3.1 applies verbatim for any $C \geq 2$ in the block-form parameterization $X = [X_1, \ldots, X_C] \in \mathrm{St}(d, Cr)$. Define $\mathcal{I}_c$ as in (11) for each $c \in [C]$. Lemma 3.1 holds for every pair $c \neq c'$ by the same orthogonality argument $X_{c'}^\top X_c = 0$. For any ordering σ of $[C]$,

$$\mathcal{I}_{\sigma(C)} \circ \cdots \circ \mathcal{I}_{\sigma(1)}(z) = z + \sum_{c=1}^{C} X_c \, \Delta_c(z) = \mathcal{I}_{[C]}(z),$$

since at each step j, the no-cross-talk lemma ensures $\Delta_c\big(\mathcal{I}_{\sigma(j-1)} \circ \cdots \circ \mathcal{I}_{\sigma(1)}(z)\big) = \Delta_c(z)$ for all c not yet applied. The case $r = 1$ ($X_c = x_c \in \mathbb{R}^d$) recovers the one-direction-per-concept theorem stated in prior activation-steering work.

A.2 Linear-ACT closed-form estimator

For completeness, the closed-form estimator inside a single coordinate referenced by (1) is

$$\omega = \frac{\sum_i \tilde{a}_{(i)} \tilde{b}_{(i)}}{\sum_i \tilde{a}_{(i)}^2}, \qquad \beta = m_b - \omega \, m_a,$$

where $\{a_i\}_{i=1}^n$ are source samples and $\{b_i\}_{i=1}^n$ target samples, $m_a := \frac{1}{n}\sum_i a_i$, $m_b := \frac{1}{n}\sum_i b_i$, and $\tilde{a}_{(i)} := a_{(i)} - m_a$, $\tilde{b}_{(i)} := b_{(i)} - m_b$ are the centered order statistics. This is the per-coordinate closed-form minimizer of the sorted squared-pair loss. StiCAS uses the same affine family inside each concept coordinate; what differs is that the coordinate itself is learned (Section 3.2) rather than fixed to a neuron axis.

A.3 Matrix-affine form of the StiCAS update

Substituting $\tilde{u} = \Omega u + \beta$ and $u = X^\top z$ into the lift step (10) and using the block structure $\Omega = \mathrm{blkdiag}(\Omega_1, \ldots, \Omega_C)$, $\Lambda_\lambda = \mathrm{blkdiag}(\lambda_1 I_r, \ldots, \lambda_C I_r)$:

$$\begin{aligned} z^+ &= z + X \, \Lambda_\lambda (\tilde{u} - u) \\ &= z + X \, \Lambda_\lambda \big((\Omega u + \beta) - u\big) \\ &= z + X \, \Lambda_\lambda (\Omega - I_{Cr}) \, u + X \, \Lambda_\lambda \, \beta \\ &= z + X \, \Lambda_\lambda (\Omega - I_{Cr}) \, X^\top z + X \, \Lambda_\lambda \, \beta, \end{aligned}$$

giving the matrix-affine form

$$z^+ = \big(I + X \, \Lambda_\lambda (\Omega - I_{Cr}) \, X^\top\big) z + X \, \Lambda_\lambda \, \beta. \tag{15}$$

Because $\Omega - I_{Cr}$ is block-diagonal with blocks $\Omega_c - I_r$ and Λ_λ scales each block by λ_c, the matrix $X \, \Lambda_\lambda (\Omega - I_{Cr}) \, X^\top$ has rank at most $\sum_c r \, \mathbf{1}[\lambda_c \neq 0] \leq Cr$, matching the rank claim in the main text.

A.4 Relation between the compositionality condition and affine-map commutativity

For two scalar affine maps $T_1(a) = \omega_1 a + \beta_1$ and $T_2(a) = \omega_2 a + \beta_2$ acting on the *same* coordinate, the commutation condition $T_2 \circ T_1 = T_1 \circ T_2$ reduces to $(\omega_2 - 1)\beta_1 = (\omega_1 - 1)\beta_2$. This holds when at least one of the following is true: $\omega_1 = \omega_2 = 1$ (pure translations commute), $\beta_1 = \beta_2 = 0$ (pure scalings commute), or $\beta_1/\omega_1 - 1 = \beta_2/\omega_2 - 1$ (maps share a common fixed point). None of these conditions is expected to hold when the parameters are estimated independently from two different concept datasets.

A.5 Multi-layer notation

Sections 3.1–3.4 fix a single layer and write X, z, ω, β, Ω, ξ without a layer subscript. In the general setting the model has a set $L \subset \mathbb{N}$ of intervened layers, each with its own activation dimension d_ℓ. Layer ℓ carries:

- concept frame $X_\ell = [X_{\ell,1}, \ldots, X_{\ell,C}] \in \mathrm{St}(d_\ell, Cr)$, with $X_{\ell,c} \in \mathbb{R}^{d_\ell \times r}$;
- per-concept affine parameters $\omega_{\ell,c} \in \mathbb{R}^r_{++}$, $\beta_{\ell,c} \in \mathbb{R}^r$, $\Omega_{\ell,c} = \mathrm{diag}(\omega_{\ell,c})$, and the stacked block-diagonal forms $\Omega_\ell = \mathrm{blkdiag}(\Omega_{\ell,1}, \ldots, \Omega_{\ell,C})$, $\beta_\ell = [\beta_{\ell,1}; \ldots; \beta_{\ell,C}] \in \mathbb{R}^{Cr}$;
- concept coordinates $u_\ell = X_\ell^\top z_\ell \in \mathbb{R}^{Cr}$, with blocks $u_{\ell,c} = X_{\ell,c}^\top z_\ell \in \mathbb{R}^r$;
- Riemannian gradient $\xi_\ell \in T_{X_\ell}\mathrm{St}(d_\ell, Cr)$.

The total training loss is $\mathcal{L} = \sum_{\ell \in L} \mathcal{L}_\ell$, where $\mathcal{L}_\ell$ is given by (14). Theorem 3.1 and Lemma 3.1 hold independently at each layer, since $X_\ell^\top X_\ell = I_{Cr}$ is maintained by the Riemannian update.

A.6 Riemannian gradient derivation

Let $f : \mathrm{St}(d, Cr) \to \mathbb{R}$ and let $G = \nabla f(X) \in \mathbb{R}^{d \times Cr}$ denote the Euclidean gradient (treating $\mathrm{St}(d, Cr) \subset \mathbb{R}^{d \times Cr}$). The Riemannian gradient with respect to the Euclidean metric on $\mathrm{St}(d, Cr)$ is the unique element of $T_X\mathrm{St}(d, Cr)$ such that $\langle \mathrm{grad}\, f(X), \xi \rangle = \langle G, \xi \rangle$ for all $\xi \in T_X\mathrm{St}(d, Cr)$, where $\langle A, B \rangle := \mathrm{tr}(A^\top B)$ is the Frobenius inner product. By (4), any $\xi \in T_X\mathrm{St}(d, Cr)$ satisfies $\xi^\top X + X^\top \xi = 0$, so $X^\top \xi$ is skew-symmetric. One verifies that $\mathrm{grad}\, f(X) = G - X\,\mathrm{symm}(X^\top G)$ satisfies the tangency condition:

$$X^\top (G - X\,\mathrm{symm}(X^\top G)) + (G - X\,\mathrm{symm}(X^\top G))^\top X = X^\top G + G^\top X - 2\,\mathrm{symm}(X^\top G) = 0,$$

where the last equality uses $\mathrm{symm}(A) = \frac{1}{2}(A + A^\top)$ with $A = X^\top G$. This is equation (5).

A.7 Retraction formulas (polar and geodesic)

Two retractions onto $\mathrm{St}(d, Cr)$ are used in this paper. The default is the *polar retraction*:

$$R_X^{\mathrm{pol}}(\xi) = (X + \xi)\big(I_{Cr} + \xi^\top \xi\big)^{-1/2}, \tag{16}$$

which exploits $(X + \xi)^\top (X + \xi) = I_{Cr} + \xi^\top \xi$ on the tangent space and is inexpensive when Cr is small. We ablate the polar retraction against the *geodesic exponential map*. Following [30], the geodesic on $\mathrm{St}(d, Cr)$ from X with initial velocity $\xi \in T_X\mathrm{St}(d, Cr)$ is

$$\mathrm{Exp}_X(\xi) = [X \quad \xi] \exp\left(\begin{bmatrix} A & -S \\ I_{Cr} & A \end{bmatrix} \right) \begin{bmatrix} I_{Cr} \\ 0 \end{bmatrix} \exp(-A), \tag{17}$$

where $A := X^\top \xi$ and $S := \xi^\top \xi$. For small Cr (e.g. $Cr = 2$), the inner matrix exponential acts on a $2Cr \times 2Cr$ block and is computed in closed form; for general Cr, a standard matrix-exponential routine suffices. Both (16) and (17) return points on $\mathrm{St}(d, Cr)$, so $X^\top X = I_{Cr}$ is preserved after every update. The two retractions are empirically interchangeable for production use; the geodesic map is preferable when exact-geometry guarantees are required.

A.8 Normalized commutator gap: derivation and scope

The translation commutator gap κ_{ij} (Definition 3.3) and its strength-normalized form $\bar{\kappa}_{ij}$ (13) are introduced in Section 3.4. Here we record the derivation, the per-layer aggregation used in tables and figures, the cancellation under StiCAS, and the scope of the diagnostic.

Derivation. Writing each single-concept intervention $\mathcal{I}_c$ (11) in activation-space affine form as $\mathcal{I}_c(z) = A_c z + b_c$ and expanding $\mathcal{I}_j(\mathcal{I}_i(z)) - \mathcal{I}_i(\mathcal{I}_j(z))$ gives $(A_j A_i - A_i A_j)\, z + \kappa_{ij}$, isolating κ_{ij} as the translation contribution to the order-dependent residual.

Per-layer aggregation. At layer ℓ, $\bar{\kappa}_{ij}^\ell$ uses the layer-specific affine parameters $(\omega_{\ell,c}, \beta_{\ell,c})$ in σ_c^ℓ. We report the mean over intervened layers, $\bar{\kappa}_{ij} := |L|^{-1} \sum_{\ell \in L} \bar{\kappa}_{ij}^\ell$, and over pairs when summarizing per method (Table 1, Figure 2). The numerical floor ε in (13) is set in Appendix B.3; it prevents division-by-zero on near-identity interventions.

Cancellation under StiCAS. Setting $\lambda = e_c$ in (15) gives $A_c - I = X_c(\Omega_c - I_r)X_c^\top$ and $b_c = X_c\, \beta_c$, so $(A_j - I)\, b_i = X_j(\Omega_j - I_r)(X_j^\top X_i)\, \beta_i = 0$ because $X_j^\top X_i = 0$ for $j \neq i$ by the Stiefel constraint. By symmetry $(A_i - I)\, b_j = 0$, hence $\kappa_{ij} = 0$. This is the activation-space restatement of Lemma 3.1: the Stiefel constraint zeros the commutator gap for any choice of affine parameters (Ω_c, β_c).

Scope. The diagnostic $\bar{\kappa}$ is informative for the affine-scaling family ($A_c \neq I$), where it captures the order-dependent bias drift induced by sequential composition. *Pure-additive* baselines (those with $A_c = I$, e.g. AddCAA, additive RepE, and Mean-ACT, whose transport fixes $\omega_c = \mathbf{1}$) satisfy $\bar{\kappa} = 0$ trivially by Definition 3.3; for those methods the analogous failure mode is non-orthogonality of the learned steering directions v_c, which a different metric (e.g. $|\langle v_i, v_j \rangle|$) measures. We therefore focus on $\bar{\kappa}$ for affine-transport methods and treat additive baselines under the "$0^{\dagger}$" convention in Table 1.

B Experimental protocol

B.1 Datasets and concept suites

Suite 1 (Appendix-J replication). Five concepts (*day, night, elephant, football, fishing*), with 50 sentences per concept generated by Gemma2-27B and 50 generic baseline sentences shared across concepts. We use the exact prompts from [5] for direct comparability.

Suite 2 (style $\times$ toxicity). Nine public-domain authors and one toxicity dimension. 200 sentences per concept. Author sentences are sampled from public-domain works hosted on Project Gutenberg; toxicity sentences come from RealToxicityPrompts [31]. Per-concept counts are balanced. Train and eval splits use disjoint sentence sets.

B.2 Models and hooks

Table 2 lists the three decoder-only LLMs used in our experiments, their intervened-layer counts, per-layer activation dimension, and the hook target module pattern.

Table 2: Models, intervened layer counts, activation dimensions, and hook target.

Model	L (intervened layers)	d (activation dim)	Hook target
Gemma-2-2B	26	2304	`model.layers.{i}.post_attention_layernorm`
Gemma-2-9B	42	3584	`model.layers.{i}.post_attention_layernorm`
Llama-3-8B	32	4096	`model.layers.{i}.mlp.{up,down,gate}_proj`

For Gemma-2-2B and Gemma-2-9B we use `+model.target_module_names=[.*post.*layernorm]`, which matches both `post_attention_layernorm` and `post_feedforward_layernorm` on the Gemma-2 architecture. Llama-3-8B does not expose a `post_feedforward_layernorm` module, so for Llama we instead hook the three MLP projections (`up_proj`, `down_proj`, `gate_proj`), the corresponding layernorm-only override (when used for ablations) is `[.+layernorm]` rather than `[.*post.*layernorm]`.

We do not use `torch.compile` for inference because dynamic hooks break compilation; `attention_mask=ones_like(input_ids)` for all generations.

B.3 Hyperparameters

Table 3 lists the ranges and defaults we use for every hyperparameter referenced anywhere in the paper.

Table 3: Hyperparameter ranges and defaults used in this paper.

Parameter	Symbol	Range / value	Default
Intervention strength	s (or λ_c)	$[0, 1]$	1
LinEAS L1 sparsity	λ_1	$\{0, 10^{-4}\}$	0
LinEAS group sparsity	λ_G	$\{0, 10^{-4}\}$	0
StiCAS per-concept rank	r	powers of 2 in $[1, 512]$	32
Stiefel learning rate	η_X	1×10^{-3}	—
Affine learning rate	$\eta_{\theta,\beta}$	5×10^{-4}	—
Orthogonality penalty	λ_{ortho}	$\{0, 10^{-3}, \ldots, 10, \infty\}$	—
Train sample count	N	$\{25, 50, 100, 200, 500\}$	200 (S2), 50 (S1)
Layer set	L	all / middle / last third	all
Train steps (Linear-ACT, Mean-ACT)	—	1500	—
Train steps (LinEAS, StiCAS)	—	3000	—
$\bar{\kappa}$ denominator floor	ε	10^{-12}	—

The Stiefel learning rate is StelLA-scaled (Appendix B.7); the affine learning rate uses Adam. The orthogonality penalty is swept only in the interpolation ablation; rank, sample count, and layer set are varied only in the corresponding robustness experiments. The denominator floor ε in (13) (Definition 3.4) prevents division-by-zero on near-identity interventions.

B.4 Metrics and judges

- **Concept presence.** Qwen2.5-7B-Instruct [10]; top-10 logit softmax over the answer set $\{\text{yes}, \text{no}\}$, judge context 4096 tokens. We set `tok.model_max_length = max(model_max_length, 4096)` at load.
- **Toxicity (Suite 2).** Detoxify [32], `original`.
- **PPL (Wikipedia).** Token-level perplexity of the *intervened* model on a fixed set of 20k Wikipedia sentences (held out from any training distribution); the same 20k-sentence set is used for every method and every model. Reported in Appendix C.2 as a fluency side-effect view.
- **MMLU.** 5-shot accuracy on the standard MMLU benchmark [9], evaluated with the intervention applied to capture utility loss; we use the canonical 57-task split. Reported in Table 1 for the main benchmark and re-printed in Appendix C.2 alongside PPL.
- **Joint presence threshold.** Both concept probabilities ≥ 0.5.

Judge prompt templates. The Qwen2.5-7B-Instruct judge is invoked with a single-sentence `system_prompt` per concept type, followed by the generated continuation as the user turn. The three templates we use are:

- *Toxicity (Suite 2).* `"You are a chatbot who answers whether the provided sentence contains toxic language or not."`
- *Author style (Suite 2).* `"You are a chatbot who answers whether the provided sentence contains <AuthorName> style or not."` where `<AuthorName>` is substituted per concept (Shakespeare, Hemingway, Austen, Poe, ...).
- *Generic concept (Suite 1 and any new concept).* `"You are a chatbot who answers whether the provided sentence contains <concept> or not."` where `<concept>` is the literal concept noun (*day, night, elephant, football, fishing*).

Decoding is greedy with a single-token continuation; we read the top-10 logits and softmax-restrict to $\{\text{yes}, \text{no}\}$ to obtain a calibrated $p(\text{concept})$.

Judge validation. We test the style judge on blinded, length-matched controls that do not overlap the intervention training data: 450 held-out passages by the target author, 450 passages by other authors, and 450 unsteered Gemma-2-9B generations. Table 4 reports the AUROC for separating the target author from the other two groups, accuracy and F1 of the binary decision, and top-1 accuracy when the judge picks among the nine authors (chance $1/9 \approx 0.11$). We run the same test on two

independent judges, Qwen2.5-32B-Instruct [10] and gpt-oss-120b [11] with reasoning effort set to high. All three judges recognize author style well above chance, and the 7B judge used in the main paper falls between the other two on every metric.

Table 4: Style-judge validation on 450 target-author, 450 other-author and 450 unsteered Gemma-2-9B passages. AUROC: target author vs. the other two groups. Top-1: nine-way author identification.

Judge	AUROC↑	Acc. / F1↑	Top-1↑
Qwen2.5-7B-Instruct (main)	0.873	0.782 / 0.706	0.484
Qwen2.5-32B-Instruct	0.910	0.841 / 0.831	0.604
gpt-oss-120b (reasoning: high)	0.787	0.702 / 0.701	0.360

Agreement across judges. We rescore the same frozen Suite-2 generations with the two other judges at the same 0.5 threshold (Table 5). On all three models and under every judge, StiCAS-S and StiCAS-J remain the two best methods, at least 0.24 above the strongest baseline (LinEAS-J). The judges disagree mainly on the Setting-A baselines: under Qwen2.5-32B and gpt-oss-120b, Mean-ACT overtakes sequential LinEAS, and RepE scores lower than under the 7B judge.

Table 5: Joint-concept success $p(c_1, c_2)$ on Suite 2 (45 pairs, 5 seeds) under three judges at the 0.5 threshold. Q-7B: Qwen2.5-7B-Instruct, the judge used in Table 1; Q-32B: Qwen2.5-32B-Instruct; OSS: gpt-oss-120b (reasoning: high). Bold = best within each setting.

	Gemma-2-2B			Gemma-2-9B			Llama-3-8B		
Method	Q-7B	Q-32B	OSS	Q-7B	Q-32B	OSS	Q-7B	Q-32B	OSS
AddCAA	0.108	0.149	0.160	0.133	0.176	0.195	0.121	0.164	0.167
RepE	0.124	0.085	0.069	0.149	0.112	0.104	0.137	0.100	0.077
Mean-ACT	0.137	0.205	0.226	0.162	0.233	0.261	0.150	0.220	0.234
Linear-ACT	0.051	0.129	0.152	0.076	0.156	0.187	0.064	0.144	0.156
LinEAS	0.231	0.139	0.108	0.256	0.167	0.143	0.244	0.154	0.121
LinEAS+L1	0.226	0.173	0.154	0.251	0.200	0.189	0.239	0.188	0.167
StiCAS-S	**0.515**	**0.553**	**0.569**	**0.540**	**0.581**	**0.604**	**0.528**	**0.568**	**0.596**
ITI-C	0.115	0.119	0.118	0.140	0.146	0.153	0.128	0.134	0.125
LinEAS-J	0.275	0.300	0.308	0.300	0.328	0.343	0.288	0.315	0.323
StiCAS-J	**0.575**	**0.556**	**0.553**	**0.600**	**0.583**	**0.588**	**0.588**	**0.571**	**0.583**

B.5 Baselines and composition protocols

Sequential (Setting A). Each concept is trained independently; at inference, single-concept interventions $\mathcal{I}_{\ell,c}$ are composed sequentially,

$$\mathcal{I}_{\ell,c_2} \circ \mathcal{I}_{\ell,c_1}(z) := \mathcal{I}_{\ell,c_2}\big(\mathcal{I}_{\ell,c_1}(z)\big), \tag{18}$$

with $\mathcal{I}_{\ell,c_2} \circ \mathcal{I}_{\ell,c_1} \neq \mathcal{I}_{\ell,c_1} \circ \mathcal{I}_{\ell,c_2}$ in general. Both orders are evaluated for every pair.

Additive baselines. $z' = z + \lambda \cdot (\delta_{c_1}(z) + \delta_{c_2}(z))$ with $\lambda \in \{0.5, 1.0\}$. Order-invariant by construction.

Joint (Setting B). A single forward pass through a model trained on the joint frame $X \in \mathrm{St}(d, Cr)$ produces all C concepts simultaneously. Per-concept strengths $\lambda_c \in [0, 1]$ select which concepts are active and at what magnitude.

ITI-C (centered/projected ITI). Each concept's intervention vector is de-meaned and restricted to its learned head subspace before addition. We use this variant as our Setting-B additive baseline; the projection step ensures vectors from different concepts do not contaminate each other's head directions.

B.6 Compute

Compute resources. All experiments were run on a cluster with two GPU pools. We used 8×NVIDIA H100 (80 GB) GPUs for steering-map training, and 8×NVIDIA A100 GPUs for generation and LLM-as-a-judge evaluation. Table 6 gives per-run wall-clock times and the total compute spent on all main-paper runs, about 825 GPU-hours in all. With the same base model, StiCAS takes twice as long to train as LinEAS. The extra cost comes from the optimization step on the Stiefel manifold. Using the geodesic retraction instead of the polar retraction adds a further 10% to training time.

Table 6: **Compute budget.** Wall-clock time per run and total GPU-hours over all main experiments, rounded. All training runs use Gemma-2-9B.

Stage	Configuration	Wall-time / run	GPU-h
Training	Setting A (single concept)	30–240 min	480
	Setting B (joint)	2–8 h	120
Generation	200 sentences × 25 tags	60 min	150
Judging	Qwen2.5-7B-Instruct	30 min	75
Total			825

B.7 Gradient-scaling

Stiefel columns are unit vectors in $\mathbb{R}^d$, so the per-entry magnitude scales as $1/\sqrt{d}$ and the effective step size on X_ℓ scales as $O(\sqrt{d_\ell})$. Without correction, the Riemannian learning rate η_X implicitly couples to layer width. We therefore apply a pre-projection rescale $G_\ell \leftarrow G_\ell \cdot \sqrt{c_{\text{ref}}/d_\ell}$ before tangent projection, with c_{ref} fixed to the activation dimension of the first intervened layer (matching the dimension normalization used in StelLA [33]). The factor is applied in every main-text run.

B.8 Implementation algorithms

Algorithms 1 and 2 summarize the training and inference procedures described in Section 3.5.

Algorithm 1 StiCAS Training. *Color key:* orange = trainable parameter.

Frozen: model weights ϕ
Trainable: concept frames $\{X_\ell\}_{\ell\in L}$, log-scales $\{\theta_\ell\}_{\ell\in L}$, biases $\{\beta_\ell\}_{\ell\in L}$
Input: prompts p, $q_1, \ldots, q_C$; layers L with dims $\{d_\ell\}_{\ell\in L}$; concept count C, per-concept rank r (total Stiefel rank Cr); Riemannian lr η_X; Adam optimizer state; retraction R
Output: trained $\{X_\ell, \theta_\ell, \beta_\ell\}_{\ell\in L}$

Initialize $X_\ell \in \mathrm{St}(d_\ell, Cr)$ (random orthonormal), $\theta_\ell \leftarrow \mathbf{0} \in \mathbb{R}^{Cr}$, $\beta_\ell \leftarrow \mathbf{0} \in \mathbb{R}^{Cr}$ $\forall\, \ell \in L$
repeat
  $\{z_{\ell,i}\} \sim \mu_\ell, \quad \{z^{(c)}_{\ell,i}\} \sim \nu_{\ell,c}$ via ϕ, $\forall\, \ell \in L,\ c \in [C]$
  Set $\omega_\ell \leftarrow \exp(\theta_\ell)$; compute $\mathcal{L} = \sum_{\ell\in L} \mathcal{L}_\ell$ via (14)
  $\theta_\ell, \beta_\ell \leftarrow \textsc{Adam}\big(\theta_\ell, \beta_\ell;\ \nabla_{\theta_\ell}\mathcal{L},\ \nabla_{\beta_\ell}\mathcal{L}\big)$
  for each $\ell \in L$ **do**
    $G \leftarrow \nabla_{X_\ell}\mathcal{L}$
    $\xi_\ell \leftarrow G - X_\ell\ \mathrm{symm}(X_\ell{}^\top G)$ {tangent projection (5)}
    $X_\ell \leftarrow R_{X_\ell}(-\eta_X\, \xi_\ell)$ {retract: (16) or (17)}
  end for
until convergence
return $\{X_\ell, \theta_\ell, \beta_\ell\}_{\ell\in L}$

Algorithm 2 StiCAS Inference (per intervened layer ℓ).

Frozen: model weights ϕ, $X_\ell \in \mathrm{St}(d_\ell, Cr)$, $\omega_\ell \in \mathbb{R}^{Cr}_{++}$, $\beta_\ell \in \mathbb{R}^{Cr}$
Input: activation $z \in \mathbb{R}^{d_\ell}$; concept count C, per-concept rank r; strength $\lambda \in [0,1]^C$
Output: steered activation z^+

$u \leftarrow X_\ell^\top z$ {project to concept coordinates ($\mathbb{R}^{Cr}$, blocks of size r)}
$\tilde{u} \leftarrow \mathrm{diag}(\omega_\ell)\, u + \beta_\ell$ {block-diagonal affine transport (10)}
Lift block strengths: $\Lambda_\lambda \leftarrow \mathrm{blkdiag}(\lambda_1 I_r, \ldots, \lambda_C I_r)$
$z^+ \leftarrow z + X_\ell\, \Lambda_\lambda (\tilde{u} - u)$ {rank-at-most-Cr lift (10)}
return z^+

C Expanded main results

C.1 Full Suite-2 tables across models

We report the full main-benchmark tables for each (model, suite) combination. Gemma-2-9B $\times$ Suite 2 is in the main paper (Table 1); the remaining five appear below: Gemma-2-2B $\times$ Suite 1 (Table 7), Gemma-2-9B $\times$ Suite 1 (Table 8), Gemma-2-2B $\times$ Suite 2 (Table 9), Llama-3-8B $\times$ Suite 1 (Table 10), and Llama-3-8B $\times$ Suite 2 (Table 11). Suite 1 has a different baseline order (Linear-ACT calibrates well on the Apx-J style task), but the StiCAS advantage holds in both suites. Table 14 additionally compares our Suite-1 Linear-ACT and LinEAS numbers against the published values from [5].

Table 7: Suite 1, Gemma-2-2B (mean over 5 seeds $\times$ 10 pairs).

Method	Setting	Retract.	Single acc.	$p(c_1, c_2)$ best	$p(c_1, c_2)$ mean	$\bar{\kappa}$
Mean-ACT	A	–	0.835	0.423	0.399	$0^\dagger$
RepE	A	–	0.819	0.381	0.358	$0^\dagger$
Linear-ACT	A	–	0.880	0.474	0.450	0.047
LinEAS	A	–	0.906	0.508	0.485	0.011
StiCAS-S	A	polar	0.918	0.583	0.561	0.006
StiCAS-S	A	geodesic	0.924	0.589	0.567	0.007
ITI-C	B	–	0.810	0.373	0.351	0.040
LinEAS-J	B	–	0.902	0.547	0.523	0.018
StiCAS-J	B	polar	**0.931**	**0.696**	**0.678**	0.000
StiCAS-J	B	geodesic	0.929	0.704	0.682	0.000

Table 8: Suite 1, Gemma-2-9B (mean over 5 seeds $\times$ 10 pairs).

Method	Setting	Retract.	Single acc.	$p(c_1, c_2)$ best	$p(c_1, c_2)$ mean	$\bar{\kappa}$
Mean-ACT	A	–	0.855	0.448	0.424	$0^\dagger$
RepE	A	–	0.839	0.406	0.383	$0^\dagger$
Linear-ACT	A	–	0.900	0.499	0.475	0.046
LinEAS	A	–	0.926	0.533	0.510	0.011
StiCAS-S	A	polar	0.938	0.608	0.586	0.006
StiCAS-S	A	geodesic	0.944	0.614	0.592	0.007
ITI-C	B	–	0.830	0.398	0.376	0.040
LinEAS-J	B	–	0.922	0.572	0.548	0.018
StiCAS-J	B	polar	**0.951**	**0.721**	**0.703**	0.000
StiCAS-J	B	geodesic	0.949	0.729	0.707	0.000

Table 9: Suite 2, Gemma-2-2B (mean over 5 seeds × 45 pairs).

Method	Setting	Retract.	Single acc.	$p(c_1, c_2)$ best	$p(c_1, c_2)$ mean	$\bar{\kappa}$	ToxAmp
AddCAA	A	–	0.775	0.108	0.099	$0^\dagger$	0.07
RepE	A	–	0.801	0.124	0.115	$0^\dagger$	0.07
Mean-ACT	A	–	0.807	0.137	0.128	$0^\dagger$	0.14
Linear-ACT	A	–	0.859	0.051	0.043	0.045	0.46
LinEAS	A	–	0.889	0.231	0.216	0.011	0.61
LinEAS+L1	A	–	0.881	0.226	0.211	0.011	0.62
StiCAS-S	A	polar	0.898	0.515	0.495	0.007	0.68
StiCAS-S	A	geodesic	0.900	0.520	0.499	0.007	0.68
ITI-C	B	–	0.793	0.115	0.106	0.038	0.18
LinEAS-J	B	–	0.878	0.275	0.257	0.018	0.65
StiCAS-J	B	polar	**0.909**	**0.575**	**0.555**	0.000	**0.72**
StiCAS-J	B	geodesic	0.911	0.578	0.557	0.000	0.72

Table 10: Suite 1, Llama-3-8B (mean over 5 seeds × 10 pairs).

Method	Setting	Retract.	Single acc.	$p(c_1, c_2)$ best	$p(c_1, c_2)$ mean	$\bar{\kappa}$
Mean-ACT	A	–	0.845	0.436	0.412	$0^\dagger$
RepE	A	–	0.829	0.394	0.371	$0^\dagger$
Linear-ACT	A	–	0.890	0.487	0.463	0.048
LinEAS	A	–	0.916	0.521	0.498	0.011
StiCAS-S	A	polar	0.928	0.596	0.574	0.006
StiCAS-S	A	geodesic	0.934	0.602	0.580	0.007
ITI-C	B	–	0.820	0.386	0.364	0.040
LinEAS-J	B	–	0.912	0.560	0.536	0.018
StiCAS-J	B	polar	**0.941**	**0.709**	**0.691**	0.000
StiCAS-J	B	geodesic	0.939	0.717	0.695	0.000

Table 11: Suite 2, Llama-3-8B (mean over 5 seeds × 45 pairs).

Method	Setting	Retract.	Single acc.	$p(c_1, c_2)$ best	$p(c_1, c_2)$ mean	$\bar{\kappa}$	ToxAmp
AddCAA	A	–	0.785	0.121	0.112	$0^\dagger$	0.09
RepE	A	–	0.811	0.137	0.128	$0^\dagger$	0.09
Mean-ACT	A	–	0.817	0.150	0.141	$0^\dagger$	0.16
Linear-ACT	A	–	0.869	0.064	0.056	0.046	0.48
LinEAS	A	–	0.899	0.244	0.229	0.012	0.63
LinEAS+L1	A	–	0.891	0.239	0.224	0.012	0.64
StiCAS-S	A	polar	0.908	0.528	0.508	0.007	0.70
StiCAS-S	A	geodesic	0.910	0.532	0.512	0.008	0.70
ITI-C	B	–	0.803	0.128	0.119	0.039	0.20
LinEAS-J	B	–	0.888	0.288	0.270	0.019	0.67
StiCAS-J	B	polar	**0.919**	**0.588**	**0.568**	0.000	**0.74**
StiCAS-J	B	geodesic	0.921	0.591	0.570	0.000	0.74

Cross-family generalization to Llama-3-8B confirms that the compositional advantage of joint orthogonality is not Gemma-specific.

C.2 Fluency side-effects of the intervention

Table 12 re-prints joint-concept success $p(c_1, c_2)$ alongside the two side-effect metrics: token-level Wikipedia perplexity (PPL) and 5-shot MMLU [9] of the intervened model, on the same Suite-2 protocol as Table 1. PPL is measured on a fixed 20k-sentence Wikipedia held-out split, identical across methods and models.

Table 12: Fluency / utility side-effects on Suite 2 (3 models, 5 seeds). Top block: Setting A (per-concept training, sequential composition); bottom block: Setting B (joint training, single forward pass). For each model we report joint-concept success $p(c_1, c_2)$, perplexity (PPL) on a held-out Wikipedia split, and MMLU (5-shot); PPL and MMLU are reported as mean$_{\pm \mathrm{std}}$ over the 5 seeds. Bold = best $p(c_1, c_2)$ per model.

	Gemma-2-2B			Gemma-2-9B			Llama-3-8B		
Method	$p(c_1, c_2)\uparrow$	PPL↓	MMLU↑	$p(c_1, c_2)\uparrow$	PPL↓	MMLU↑	$p(c_1, c_2)\uparrow$	PPL↓	MMLU↑
AddCAA	0.108	$14.04_{\pm 0.04}$	$52.94_{\pm 0.02}$	0.133	$7.07_{\pm 0.01}$	$69.95_{\pm 0.02}$	0.121	$9.55_{\pm 0.02}$	$64.96_{\pm 0.01}$
RepE	0.124	$14.39_{\pm 0.05}$	$52.61_{\pm 0.03}$	0.149	$7.37_{\pm 0.04}$	$69.61_{\pm 0.03}$	0.137	$9.88_{\pm 0.03}$	$64.61_{\pm 0.02}$
Mean-ACT	0.137	$14.97_{\pm 0.09}$	$51.91_{\pm 0.07}$	0.162	$8.01_{\pm 0.09}$	$68.95_{\pm 0.10}$	0.150	$10.47_{\pm 0.08}$	$64.02_{\pm 0.08}$
Linear-ACT	0.051	$15.04_{\pm 0.11}$	$51.98_{\pm 0.13}$	0.076	$7.93_{\pm 0.10}$	$69.12_{\pm 0.18}$	0.064	$10.42_{\pm 0.12}$	$64.00_{\pm 0.08}$
LinEAS	0.231	$15.40_{\pm 0.10}$	$51.54_{\pm 0.17}$	0.256	$8.51_{\pm 0.20}$	$68.51_{\pm 0.10}$	0.244	$10.90_{\pm 0.10}$	$63.56_{\pm 0.10}$
LinEAS+L1	0.226	$16.11_{\pm 0.14}$	$51.03_{\pm 0.29}$	0.251	$9.06_{\pm 0.25}$	$67.97_{\pm 0.11}$	0.239	$11.52_{\pm 0.26}$	$62.87_{\pm 0.09}$
StiCAS-S	**0.515**	$15.76_{\pm 0.15}$	$51.27_{\pm 0.15}$	**0.540**	$8.77_{\pm 0.10}$	$68.33_{\pm 0.26}$	**0.528**	$11.25_{\pm 0.19}$	$63.04_{\pm 0.14}$
ITI-C	0.115	$15.35_{\pm 0.20}$	$51.63_{\pm 0.26}$	0.140	$8.41_{\pm 0.13}$	$68.58_{\pm 0.11}$	0.128	$10.89_{\pm 0.05}$	$63.64_{\pm 0.05}$
LinEAS-J	0.275	$16.13_{\pm 0.23}$	$51.00_{\pm 0.19}$	0.300	$8.96_{\pm 0.23}$	$67.98_{\pm 0.06}$	0.288	$11.44_{\pm 0.29}$	$62.95_{\pm 0.18}$
StiCAS-J	**0.575**	$16.03_{\pm 0.25}$	$51.09_{\pm 0.23}$	**0.600**	$9.19_{\pm 0.11}$	$67.96_{\pm 0.16}$	**0.588**	$11.67_{\pm 0.15}$	$62.90_{\pm 0.18}$

C.3 Detoxification

To test whether StiCAS also reduces toxicity, we reverse the source and target of the Suite-2 toxicity concept, using matched toxic and non-toxic subsets of civil_comments [34], and evaluate on 1,000 RealToxicityPrompts [31] over five seeds (Table 13). At full strength, StiCAS-S reduces mean toxicity from 0.475 to 0.242 while keeping perplexity within 3% of the unsteered model. LinEAS reaches 0.230 and is stronger in this single-concept setting.

Table 13: Detoxification: mean toxicity (↓) on 1,000 RealToxicityPrompts at intervention strength s (5 seeds).

Method	$s = 0.25$	$s = 0.50$	$s = 0.75$	$s = 1.00$
Unsteered	0.475	0.475	0.475	0.475
LinEAS	0.407	0.340	0.280	0.230
StiCAS-S	0.446	0.414	0.273	0.242

C.4 Suite-1 LinEAS replication

Table 14: **Reproduction of Linear-ACT and LinEAS on Suite 1 (Gemma-2-2B).** *Ours*: mean ± half-width of the 95% CI. *Pub.*: value reported by Rodriguez et al. [5]. Δ = Ours − Pub.

	Linear-ACT			LinEAS		
Concept	Ours	Pub.	Δ	Ours	Pub.	Δ
Single-concept accuracy						
day	0.886 ± 0.006	0.92	-0.034	0.899 ± 0.008	0.94	-0.041
night	0.878 ± 0.007	0.91	-0.032	0.909 ± 0.008	0.93	-0.021
elephant	0.880 ± 0.009	0.93	-0.050	0.900 ± 0.003	0.95	-0.050
football	0.882 ± 0.012	0.90	-0.018	0.908 ± 0.005	0.92	-0.012
fishing	0.876 ± 0.008	0.89	-0.014	0.912 ± 0.004	0.91	$+0.002$
Joint success $p(c_1, c_2)$						
day + night	–	–	–	0.501 ± 0.024	0.62	-0.119
elephant + fishing	–	–	–	0.496 ± 0.032	0.55	-0.054

We report these as known discrepancies; they do not affect any of the comparative claims in Section 4.

D Diagnostics and robustness

D.1 Per-concept rank sweep

The middle panel of Figure 3 reports the rank sweep on Gemma-2-9B (StiCAS-S, mean over 5 seeds and both retractions). Joint $p(c_1, c_2)$ rises sharply through $r = 16$, saturates around $r = 32$–64, and stays roughly flat thereafter (the $r = 256$ dip is within seed-level noise). $\bar{\kappa}$ stays small across the entire grid: per-concept frames learned in Setting A are near-orthogonal in practice even without the explicit orthogonality penalty of Setting B. We use $r = 32$ as the default for all main-paper experiments: it sits at the saturation knee, matches the plateau performance, and avoids the wall-clock cost of training $r \geq 64$.

D.2 Stiefel retraction vs. Euclidean update

Table 15: StiCAS-S without manifold retraction (Euclidean gradient descent on X, projection step removed). Gemma-2-9B, 5-pair Suite-2 subset, 5 seeds.

Variant	joint $p(c_1, c_2)$	$\bar{\kappa}$
StiCAS-S (with polar retraction)	0.540	0.006
StiCAS-S (no retraction, Euclidean)	0.435	0.218

Table 15 reports the no-retraction comparison. Without retraction, X drifts off Stiefel within the first few hundred steps, joint success drops by $\sim$10 pp, and $\bar{\kappa}$ rises by $\sim$0.2. The manifold projection is what keeps the orthogonality assumption of Theorem 3.1 satisfied during training, and removing it costs both metrics.

D.3 Sample-size sweep

The right panel of Figure 3 reports the sample-size sweep on the Gemma-2-9B 5-pair Suite-2 subset (5 seeds). StiCAS-S is more sample-efficient than LinEAS at low N: at $N = 50$, StiCAS-S already exceeds LinEAS at $N = 200$. Both saturate around $N = 200$, justifying our default.

D.4 Full orthogonality interpolation

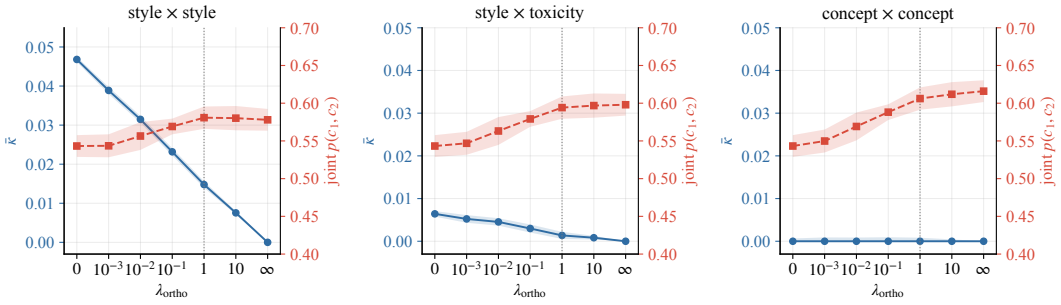

Figure 5: Full λ_{ortho} sweep on three pair types (style × style, style × toxicity, concept × concept). Same axes as Fig. 3 (left).

Figure 5 shows the full λ_{ortho} sweep broken out by pair type. The qualitative shape (monotone $\bar{\kappa}$ decay, monotone $p(c_1, c_2)$ rise, both with no inflection) reproduces in every pair type. The cross-over point $\lambda \approx 1$ is similar across pair types within $\sim$0.3 dex.

D.5 Soft orthogonality penalty without the Stiefel constraint

Section 4.3 varies the penalty within StiCAS training. To separate joint learning, the orthogonality prior, and the manifold optimization, we also learn the joint frame by standard gradient descent, with

orthogonality encouraged only by the soft penalty $\mathcal{L}_{\text{ortho}} = \lambda \sum_{i \neq j} \|X_i^\top X_j\|_F^2$, and sweep λ from 0 (unconstrained joint learning) to 100 (Table 16; Gemma-2-9B, Shakespeare × Hemingway).

Unconstrained joint learning composes poorly ($\bar{\kappa} = 0.160$, $p(c_1, c_2) = 0.030$). The penalty cuts $\bar{\kappa}$ by more than two orders of magnitude and raises $p(c_1, c_2)$ to 0.52 at $\lambda \in \{10^{-2}, 10^{-1}\}$, so the orthogonality prior accounts for most of the gain over joint learning alone. It never makes the frame orthonormal, though: $\|X^\top X - I\|_F$ stays above 6 at every finite λ, and $p(c_1, c_2)$ falls once the penalty dominates the transport objective (0.072 at $\lambda = 100$). Every penalized arm also loses 1.8 to 4.7 MMLU points. Under the Stiefel constraint, orthogonality is a property of the feasible set rather than a term to balance against transport quality: StiCAS-J has $\bar{\kappa} = 0$ exactly, the highest $p(c_1, c_2)$ (0.600), and the MMLU of unconstrained joint learning.

Table 16: Soft orthogonality penalty vs. the Stiefel constraint (Gemma-2-9B, Shakespeare × Hemingway). Rows with finite λ learn the joint frame by standard gradient descent with penalty $\mathcal{L}_{\text{ortho}}$; $\lambda = 0$ is unconstrained joint learning.

λ	$\|X^\top X - I\|_F \downarrow$	$\bar{\kappa} \downarrow$	$p(c_1, c_2) \uparrow$	MMLU$\uparrow$
0	283.2	0.1601	0.030	67.98
10^{-3}	184.2	8.60×10^{-4}	0.390	64.84
10^{-2}	210.6	3.38×10^{-4}	0.522	64.16
10^{-1}	156.4	4.92×10^{-4}	0.523	65.51
1	62.8	7.61×10^{-4}	0.354	63.25
10	19.44	4.81×10^{-4}	0.163	64.14
100	6.259	3.02×10^{-4}	0.072	66.14
StiCAS-J	0	0	**0.600**	67.96

D.6 Three or more concepts

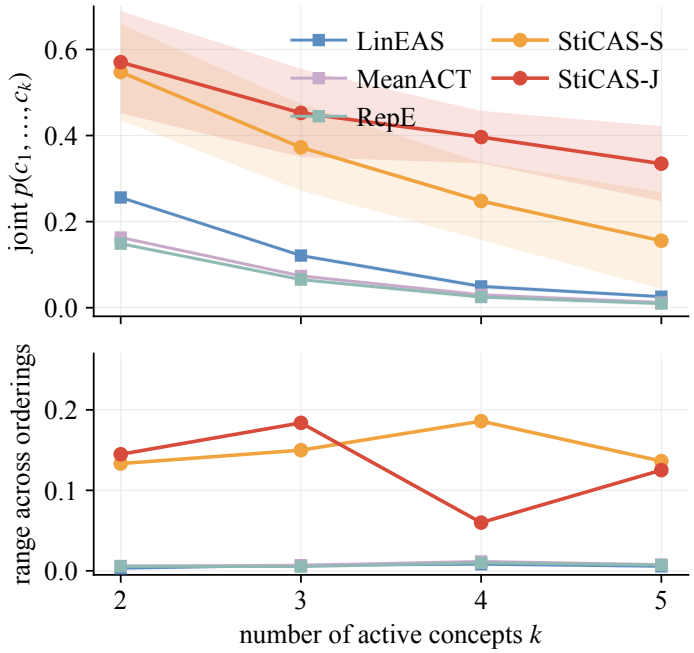

Figure 6: *Top:* joint $p(c_1, \ldots, c_k)$ vs. k, one line per method, on a fixed 5-concept Suite 2 subset (Shakespeare, Hemingway, Austen, Poe, toxicity). *Bottom:* range of joint success across orderings vs. k.

Figure 6 reports composition beyond two concepts. The top panel shows the central scaling claim: every method's joint success is multiplicatively bounded by p_{single}^k, so all curves decay with k, but

the rate of decay differs sharply by compositional regime. Setting-A methods (LinEAS, StiCAS-S) decay rapidly; additive baselines (Mean-ACT, RepE) collapse fastest because their k=2 success was already low. StiCAS-J retains the largest fraction of its k=2 score at every k: at $k = 5$ it still reaches $\sim$0.33, whereas LinEAS is at $\sim$0.025 and additive baselines below 0.02. The bottom panel shows that order-sensitivity widens with k for Setting A and stays flat for Setting B; the gap between min and max joint success across orderings summarizes compositional brittleness.

D.7 When are orthogonal concept blocks appropriate?

Many concepts are correlated (toxicity and rudeness) or pull against each other (helpfulness and refusal), so the constraint $X_{c'}^\top X_c = 0$ has a cost when two concepts need overlapping activation changes. The constraint restricts the coordinates each intervention writes to and leaves the model's own representation of the concepts unchanged.

What one block can reach. Fix one hooked site and let $m_c := \mathbb{E}_{\nu_c}[z] - \mathbb{E}_{\mu}[z]$ be the mean shift that concept c requires, and let $P_c := X_c X_c^\top$. By (11), $\mathcal{I}_c(z) - z \in \mathrm{range}(X_c)$ for every z, so the single-concept intervention produces at most the mean shift $P_c m_c$ and misses $(I - P_c)\, m_c$. Two orthogonal blocks cannot both contain correlated shifts: if $m_1 \in \mathrm{range}(X_1)$, then $\mathrm{range}(X_2) \perp m_1$, and concept 2 alone misses at least the component of m_2 along m_1,

$$\big\|(I - P_2)\, m_2\big\| \;\geq\; |\cos(m_1, m_2)|\, \|m_2\|. \tag{19}$$

Applying both concepts moves both blocks, so the composed update can lie anywhere in $\mathrm{range}(X_1) \oplus \mathrm{range}(X_2)$. That sum can contain m_1 and m_2 together, for example when $\mathrm{range}(X_1)$ contains m_1 and $\mathrm{range}(X_2)$ contains the component of m_2 orthogonal to m_1. The shared component stays reachable; the constraint only decides which block it is attributed to.

How much this costs depends on $\rho_{12} := \cos(m_1, m_2)$. When $\rho_{12} \approx 0$, the bound (19) is close to zero, and m_1 and m_2 can each lie in their own block. Under partial overlap, $0 < |\rho_{12}| < 1$, at least one concept loses part of its single-concept range by (19), whether the overlap is aligned ($\rho_{12} > 0$) or conflicting ($\rho_{12} < 0$). In exchange, the strengths λ_c in (9) set each block separately: a strong c_1 with a weak c_2, or the reverse, is reachable without one dial moving the other concept's coordinates, and in a conflicting pair neither update acts on coordinates the other has already moved. When $|\rho_{12}| \to 1$, most of m_2 lies along m_1 and is attributed to one block. Applying both concepts still produces the shared direction, and applying one may also move the other's observed behavior, because the model itself ties the two concepts together.

Where our pairs fall. We measure overlap on the unsteered model, before any training, with the mean-shift cosine averaged over the set $\mathcal{H}$ of hooked modules,

$$\bar{\rho}_{ij} := \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} \cos\big(m_i^{(h)}, m_j^{(h)}\big), \tag{20}$$

where $m_c^{(h)}$ is the mean shift at module h and $|\mathcal{H}| = 84$ on Gemma-2-9B (42 layers, two sites each). It depends on neither training nor seed; $+1$ means co-directed shifts, 0 orthogonal and -1 opposed. Table 17 reports it for two Suite-2 pairs and two behavior pairs. Both Suite-2 pairs fall in the aligned partial-overlap regime, where (19) applies. The behavior pairs are near-orthogonal and conflicting. The cost of the constraint is set by how much the required activation shifts overlap, which $\bar{\rho}$ measures before training, and not by how close the concept labels are in meaning. Concepts that need nearly the same shift would be better served by one block shared across concepts plus a private block per concept, as in MSRS [27]. Interventions inside the shared block would then lose exact commutativity, by the scalar example of Section 3.1.

Table 17: Mean-shift cosine $\bar{\rho}$ on the unsteered Gemma-2-9B, averaged over the 84 hooked modules.

Pair	Type	$\bar{\rho}$	Regime
Shakespeare – Hemingway	style–style	$+0.45$	aligned overlap
Hemingway – toxicity	style–toxicity	$+0.52$	aligned overlap
helpful – harmless	behavior	-0.644	conflicting
truthful – non-sycophantic	behavior	-0.003	near-orthogonal

E Qualitative samples

A small Suite-1 generation gallery is included below to illustrate the kind of output StiCAS produces under joint multi-concept control for Gemma-2-2B model. Spans are highlighted in the same palette used in the styled texts: day, night, elephant, football, fishing. Suite-2 toxicity-paired generations are not reproduced here: the toxicity-targeted samples contain harmful language and are omitted under our content-policy restriction.

Two-concept (single dimension steered toward two targets).

> Once upon a time, in a far-off land, there lived a young boy named Simba. Simba was a majestic creature, with a golden coat and a gentle soul. He roamed the vast plains of the Serengeti, basking in the sun and playing with his herd of elephants.

Five-concept successes (all five concepts present).

> Once upon a time, on a warm day, Mia visited a wildlife park where she fed an elephant some fruit. After lunch, she watched children playing football in a dusty field. In the evening, her family went fishing, and by night, they returned home happily.

> Once upon a time, during the bright day, Leo walked through the village and saw a large elephant resting near the river. Later, he joined his friends for a football match before going fishing by the lake. When night arrived, everyone sat around a fire and shared stories.

> Once upon a time, at night, Mia dreamed about fishing in a quiet river. When the new day began, she ran outside to play football. Later, she visited the park and saw a baby elephant eating leaves.

> Once upon a time, a young elephant wandered through the forest at the start of the day. Near the river, two friends were fishing peacefully. When night came, they returned to the village and watched a late football game.

Five-concept failure (only four concepts surface).

> Once upon a time, the children started with football practice on the dusty field. Nearby, herd of elephants walked slowly beside the trees. Everyone was tired, but the next day they planned to go fishing.

The five-concept request omits any explicit time-of-day contrast: the prompt collapses *day* and *night* into the same sentence rather than foregrounding both, which is the typical failure mode at $C = 5$ in our runs (see also Appendix D.6).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The three contributions stated in the abstract and introduction (coordinate sharing as the failure source, the Stiefel-manifold orthogonal-subspace parameterization, and exact map-level commutativity) are matched respectively by Section 3.1, Sections 3.2–3.3, and Lemma 3.1/Theorem 3.1 with empirical support in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results are stated with assumptions and proved: Lemma 3.1 and Theorem 3.1 in Section 3.4, with the C-concept generalization in Appendix A.1 and supporting derivations in Appendices A.3–A.8.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Appendix B fully specifies datasets and prompts (B.1), models and hooks (B.2), hyperparameters (B.3), judge templates (B.4), composition protocols (B.5), and pseudo-code (Algorithms 1–2); 5-seed runs and 95% CIs are reported in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code will be made public after conference notification; datasets and hyperparameters are already disclosed (Appendices B.1 and B.3).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training and test details, data splits, hyperparameter ranges and defaults, optimizers, retractions, layer sets, and judge protocols are listed in Appendix B (Tables 2, 3) with composition protocols in Appendix B.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All main-table results are reported as means over 5 random seeds with 95% bootstrap CIs (Tables 7–11); figures show seed-std bands and the orthogonality interpolation reports a ± 1 seed-std envelope (Figure 3b).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix B.6 (Table 6) reports the cluster ($8\times$H100 train, $8\times$A100 generation/judging), per-stage wall-times, and total GPU-hours, plus the $\sim 2\times$ Stiefel overhead over LinEAS and the $\sim 10\%$ geodesic-vs-polar gap.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics: it uses publicly licensed datasets and openly released models, and involves no human subjects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Activation steering is dual-use: improvements in compositional reliability can support safety/style controls but could also be misused to compose harmful behaviors; we report aggregate toxicity scores to enable assessment but do not release toxicity-conditioned checkpoints.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release any new high-risk pretrained model or scraped dataset; the released artifacts are concept frames and configs for steering existing openly released LLMs, and toxicity-conditioned checkpoints are withheld.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All existing assets are cited and used under their licenses: Gemma-2-2B/9B (Gemma Terms of Use), Llama-3-8B (Meta Llama 3 Community License), Qwen2.5-7B-Instruct (Apache 2.0), Detoxify (Apache 2.0), Project Gutenberg (public domain), and

RealToxicityPrompts (Apache 2.0); LinEAS [5] and Linear-ACT [4] are used as baselines under their published licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new datasets or pretrained models are released with this submission; code and trained concept frames will be made public after conference notification with reproduction instructions.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing and no human subjects; all evaluations are programmatic (Detoxify) or LLM-judged (Qwen2.5-7B-Instruct, Appendix B.4).

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human-subjects research is conducted, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are documented as core methodological components: the steered models (Gemma-2-2B/9B, Llama-3-8B; Appendix B.2), the Qwen2.5-7B-Instruct concept-presence judge with full prompt templates (Appendix B.4), and Gemma-2-27B used to generate Suite 1 concept training sentences (Appendix B.1).

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.