Why Geometric Continuity Emerges in Deep Neural Networks: Residual Connections and Rotational Symmetry Breaking

Kyungwon Jeong[1] **Won-Gi Paeng**[1,2] **Honggyo Suh**[1]

[1]Hyntel [2]Neurometry

kwjeong@hyntel.net wgpaeng@neurometry.co.kr hgsuh@hyntel.net

Abstract

Weight matrices in deep networks exhibit geometric continuity—principal singular vectors of adjacent layers point in similar directions. While this property has been widely observed, its origin remains unexplained. Through experiments on toy MLPs and small transformers, we identify two mechanisms: residual connections create cross-layer gradient coherence that aligns weight updates across layers, and symmetry-breaking nonlinearities constrain all layers to a shared coordinate frame, preventing the rotation drift that would otherwise destabilize weight structure. Crucially, a nonlinear but rotation-equivariant activation reaches less than half of ReLU's continuity, isolating symmetry breaking—not nonlinearity alone—as the active ingredient. Activation and normalization play distinct roles: activation concentrates continuity in the leading singular direction, while normalization distributes it across multiple directions. In transformers, continuity is *projection-specific*: Q, K, Gate, and Up (which read from the residual stream) develop input-space ($\mathbf{v}_1$) continuity; O and Down (which write to it) develop output-space ($\mathbf{u}_1$) continuity; V alone, lacking an adjacent nonlinearity, develops only low continuity.

Code: `https://github.com/smallhours19/geometric-continuity`

1 Introduction

Recent work has revealed that weight matrices of adjacent layers in large language models share similar structure [1, 2]. We observe that this similarity extends to principal singular vectors, which point in similar directions across adjacent layers, forming smooth trajectories across depth—a property we term *geometric continuity*. Cross-layer similarity of this kind is what layer pruning [3, 4] and cross-layer parameter sharing [2] exploit; whether geometric continuity itself yields such gains remains an open question.

To quantify this, we decompose each layer's weight matrix via SVD and measure the cosine similarity of the leading right singular vectors ($\mathbf{v}_1$) between adjacent layers in pretrained Llama-3.1-8B [5] (Appendix A). We find that continuity is *space-specific*: Q, K, Gate, and Up projections show $\mathbf{v}_1$ (input-space) continuity, O and Down show $\mathbf{u}_1$ (output-space) continuity, while V develops only low continuity.

Despite these observations, a fundamental question remains unanswered: **why does geometric continuity emerge?** Is it an inevitable consequence of training neural networks, or does it require specific architectural or data conditions?

In this work, we investigate the origin of geometric continuity through ablation experiments on toy MLPs (MNIST) and small transformers (WikiText-103). Figure 1 previews the central result: pro-

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

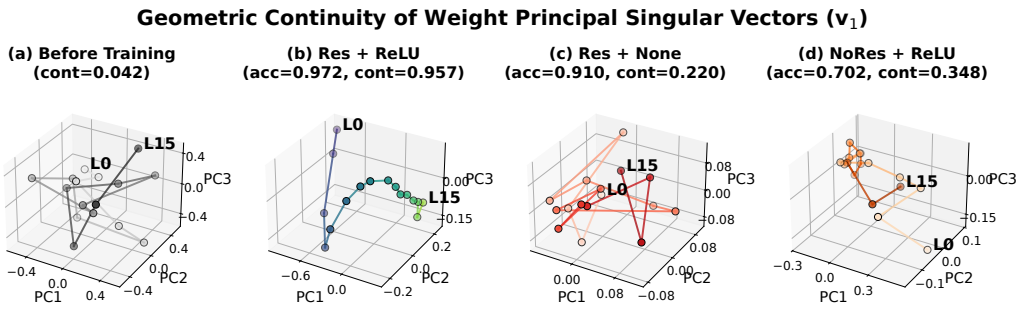


Figure 1: **Geometric continuity of weight $\mathbf{v}_1$ across layers.** 3D PCA of principal right singular vectors from a 16-layer MLP trained on MNIST. **(a)** Before training: random. **(b)** Res+ReLU: smooth trajectory. **(c)** Res+None: activation removed, low continuity. **(d)** NoRes+ReLU: residual removed, weak continuity. Single run; seed means in Table 1.

jected onto three principal components, the $\mathbf{v}_1$ of a 16-layer MLP's layers trace a smooth trajectory only when both a residual connection and a symmetry-breaking activation are present. Our key findings are:

1. **Two necessary conditions** in toy MLPs (Figure 1): residual connections provide cross-layer gradient coherence, and symmetry-breaking nonlinearities (activation, normalization) constrain all layers to a shared coordinate frame, preventing rotation drift that would otherwise destabilize weight structure. Activation and normalization play distinct roles: activation concentrates continuity in the leading singular direction, while normalization distributes it across multiple directions. Weight $\mathbf{v}_1$ emerges as the principal direction of each layer's cumulative gradient over training. A nonlinear but rotation-equivariant activation reaches less than half of ReLU's continuity (Section 4.3): symmetry breaking, not nonlinearity alone, is the active ingredient.
2. **Projection-specific continuity** in transformers: Q, K, Gate, and Up (which read from the residual stream) develop input-space ($\mathbf{v}_1$) continuity; O and Down (which write to it) develop output-space ($\mathbf{u}_1$) continuity; V alone, lacking an adjacent nonlinearity, develops only low continuity. Direct ablation confirms the Gate dependence: removing the MLP activation drops Gate $\mathbf{v}_1$ from 0.66 to 0.37 with only a modest perplexity change ($\sim 3\%$).

2 Related Work

Cross-layer weight and Jacobian structure. Several works have independently observed that deep networks exhibit cross-layer geometric structure. Min and Wang [1] propose DOCS, a metric based on column-wise cosine similarity distributions, revealing that adjacent layers form similarity clusters. Wang et al. [2] exploit shared SVD bases across layers for LLM compression. Most closely related to our work, Li and Papyan [6] discover that residual Jacobians exhibit top singular vector alignment across depth that vanishes when skip connections are removed—directly corresponding to our Condition 1 (residual connections). Their analysis focuses on Jacobians rather than weight matrices (which can diverge under nonlinearity) and identifies only the residual condition. Three elements are new here: our Res+None experiment shows that residual alone is insufficient for weight continuity, motivating the additional role of symmetry-breaking nonlinearity (Condition 2); we quantify that condition *continuously* through an equivariance error, rather than treating symmetry breaking as a discrete property of the activation (Section 4); and we identify a projection-specific read/write pattern in transformers, in which input- and output-space continuity separate according to where each matrix contacts the residual stream (Section 5). Razzhigaev et al. [7] show that adjacent transformer layer *outputs* are nearly linearly related (Procrustes similarity ~ 0.99), with the linear relationship weakening when residual connections are removed. Our work complements these observations by identifying *why* this structure emerges from a mechanistic perspective.

Implicit structural regularization. Ji and Telgarsky [8] prove that, in deep linear networks trained under gradient descent, the top rank-1 components of adjacent layers align: $|\mathbf{v}_{l+1}^\top \mathbf{u}_l| \to 1$ (signal-flow alignment between adjacent layers), providing a theoretical precedent for cross-layer

structural regularity. Marion et al. [9] show that deep residual networks are implicitly regularized toward neural ODE solutions, explaining smooth cross-layer transformations. Beaglehole et al. [10] establish that the left singular structure of weight matrices aligns with pre-activation tangent features (a gradient-derived quantity), a closely related phenomenon to our finding that weight $\mathbf{v}_1$ reflects accumulated gradient directions.

Symmetry and symmetry breaking. Saxe et al. [11] derive exact learning dynamics for deep linear networks under orthogonal invariance of adjacent weight products, providing a foundation for analyzing rotational degrees of freedom in such models. Godfrey et al. [12] formalize this by characterizing the *intertwiner group* for each activation type: linear networks admit the full general linear group as symmetry, while element-wise activations such as ReLU reduce it to a subgroup (permutations and positive rescaling) that preserves coordinate axes. Marcotte et al. [13] extend this analysis to residual architectures, proving that an isolated residual block preserves the same conservation laws as its non-residual counterpart, while no conservation law depends only on the two weight matrices adjacent to a skip connection—a result for nonlinear blocks, consistent with the frame pinning we find once a nonlinearity is present. Our finding that gradient-to-weight transfer is unstable in linear residual networks can be understood through this lens: a nonlinearity in the residual branch leaves no function-preserving relative rotation of adjacent layers' frames, removing the freedom that would otherwise allow weight frames to drift and enabling stable imprinting of gradient directions onto weights.

Layer similarity and pruning. Gromov et al. [3] show that up to half of LLM layers can be pruned with minimal performance loss, and Men et al. [4] identify redundant layers via Block Influence scores. These results are consistent with geometric continuity—adjacent layers that share similar principal directions may perform similar transformations—although whether continuity itself makes layers prunable is an open question (Section 6).

3 Background

Singular value decomposition. Any weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$ admits the decomposition $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are orthogonal, and $\mathbf{\Sigma}$ is diagonal with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$. We call $\mathbf{v}_1$ (first column of $\mathbf{V}$) the *principal right singular vector* (input space) and $\mathbf{u}_1$ (first column of $\mathbf{U}$) the *principal left singular vector* (output space).

Geometric continuity. For a sequence of weight matrices $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}$ across layers, we define *geometric continuity* as the mean absolute cosine similarity between adjacent layers' principal singular vectors:

$$\text{Continuity}(\mathbf{v}_1) = \frac{1}{L-1}\sum_{l=1}^{L-1}\left|\cos\left(\mathbf{v}_1^{(l)},\ \mathbf{v}_1^{(l+1)}\right)\right| \tag{1}$$

The absolute value handles the intrinsic sign ambiguity of singular vectors: $(\mathbf{u}, \mathbf{v})$ and $(-\mathbf{u}, -\mathbf{v})$ represent the same SVD component. A value near 1 indicates that $\mathbf{v}_1$ evolves smoothly across layers; a value near $\sqrt{2/(\pi d)}$ (where d is the vector dimension) indicates random, unstructured directions.

Frame-sensitive versus invariant alignment. The signal-flow alignment of Ji and Telgarsky [8], $s_l = |\cos(\mathbf{u}_1^{(l)}, \mathbf{v}_1^{(l+1)})|$, is invariant to rotating the shared interface. We report it alongside, but take the frame-sensitive continuity as primary: pruning, sharing and merging act on the weights and need a common frame, and in nonlinear residual networks the frame is bound to the function (Appendix I).

σ^2-weighted continuity. The $\mathbf{v}_1$ metric ignores whether the leading direction is actually dominant. The σ^2-weighted variant (σ^2-WA) averages the alignment of all $K = \min(m, n)$ singular directions of $\mathbf{W} \in \mathbb{R}^{m\times n}$, weighted by their share of $\|\mathbf{W}\|_F^2 = \sum_k \sigma_k^2$ (anchored to layer l):

$$\text{Continuity}_{\sigma^2} = \frac{1}{L-1}\sum_{l=1}^{L-1}\sum_{k=1}^{K}\frac{\left(\sigma_k^{(l)}\right)^2}{\sum_j\left(\sigma_j^{(l)}\right)^2}\left|\cos\left(\mathbf{v}_k^{(l)},\ \mathbf{v}_k^{(l+1)}\right)\right| \tag{2}$$

It reduces to $\mathbf{v}_1$ continuity when σ_1 dominates and weights all directions equally when the spectrum is flat; Section 4.3 uses it to separate activation (structure concentrated in $\mathbf{v}_1$) from LayerNorm (structure spread over many directions).

Gauge symmetry and rotation orbits. Linear products of weight matrices possess a continuous symmetry: for any orthogonal $\mathbf{R}$, $(\mathbf{W}_l\mathbf{R}^\top)(\mathbf{R}\mathbf{W}_{l-1}) = \mathbf{W}_l\mathbf{W}_{l-1}$ leaves the product unchanged. The set of all weight pairs related by such rotations forms an *orbit* of the rotation group $SO(d)$ (the group of d-dimensional rotations). Within an orbit the combined function is identical, so the loss landscape is flat along orbit directions: nothing resists motion along them, and an optimizer that is not rotation-equivariant, such as Adam, can drift along them. Section 4.3 shows that a nonlinear activation breaks the gauge symmetry, resisting the drift.

4 What Causes Geometric Continuity?

We investigate the origin of geometric continuity through ablation experiments on a toy MLP, systematically isolating the contribution of each architectural and data factor.

4.1 Experimental Setup

We use a residual MLP with D=256 hidden dimensions and L=16 layers (depth sweep in Appendix F). Each layer computes $\mathbf{h}_{l+1} = \mathbf{h}_l + \text{act}(\mathbf{W}_l\mathbf{h}_l)$, where $\mathbf{W}_l \in \mathbb{R}^{256\times256}$ has no bias and is initialized with PyTorch's default Kaiming uniform scheme (bound $= 1/\sqrt{256} \approx 0.0625$, std ≈ 0.036). An embedding layer maps inputs from $\mathbb{R}^{784}$ to $\mathbb{R}^{256}$, and a linear head maps to 10 classes. We train on MNIST [14] (60K train / 10K test, standard split) with Adam (lr $= 10^{-3}$, batch size 128) for 20 epochs. Unless otherwise noted, all results report means over 3 seeds $(42, 123, 777)$; the baseline Res+ReLU achieves 97.6% test accuracy (not tuned for accuracy; Appendix G).

We measure $\mathbf{v}_1$, $\mathbf{u}_1$, and σ^2-weighted continuity (Section 3; $\mathbf{v}_2$ in Appendix G) for both weight and gradient matrices across the 16 layers. Each configuration trains on a single NVIDIA A100 or H200 GPU in under 10 minutes.

Experiments. Sections 4.2–4.4 analyze four families of experiments on this testbed: (i) a *symmetry-class ablation* (Table 1; all 21 configurations in Appendix G) of the residual connection, activation and LayerNorm, with controls separating nonlinearity from $SO(d)$ symmetry breaking (fixed linear breakers Diag and Perm; an exactly $SO(d)$-equivariant nonlinearity, EqNL); (ii) a *small-initialization* run of Res+None and Res+ReLU (σ=0.0001, 50 epochs) in which the first epoch's $\Delta\mathbf{W}$ dominates $\mathbf{W}^{(0)}$, separating the formation of continuity from its retention (Figure 2); (iii) two *interventions* on the flat directions: projecting them out of every update, and injecting rotations into trained networks (Appendix I); (iv) *optimizer and initialization-scale* ablations (SGD with momentum; Adam β_2=0 with shrinking initialization; Appendix F).

4.2 Condition 1: Residual Connections

Residual connections create gradient coherence from initialization. In a residual network, the gradient with respect to layer l's input satisfies:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{h}_l} = \frac{\partial\mathcal{L}}{\partial\mathbf{h}_{l+1}}\left(\mathbf{I} + \frac{\partial f_l}{\partial\mathbf{h}_l}\right) \tag{3}$$

The identity term $\mathbf{I}$ ensures that adjacent layers receive similar error signals. We verify this empirically with Res+ReLU: gradient $\mathbf{v}_1$ cross-layer continuity is 0.95 *at the first backward pass*, before any weight update, while weight $\mathbf{v}_1$ continuity starts at 0.04 (random) and gradually increases to 0.96 during training (Figure 12 in Appendix E).

Without residual, gradient coherence is lost. Removing residual connections ($\mathbf{h}_{l+1} = \text{act}(\mathbf{W}_l\mathbf{h}_l)$) eliminates gradient coherence: gradient $\mathbf{v}_1$ continuity drops to 0.05–0.07 (random level; Table 10) by the end of training, and weight $\mathbf{v}_1$ continuity reaches only 0.04–0.36 (Table 1), far below the 0.96 obtained with residual connections (NoRes+ReLU's higher $\mathbf{u}_1$, 0.65, suggests a forward-signal pathway we leave to future work).

Table 1: Ablation results (MNIST, 16-layer MLP, 3 seeds, mean), grouped by symmetry class and ordered by equivariance error eq-err within each group (definitions in Sections 3 and 4.3). $\bar{\mathbf{G}}\mathbf{v}_1$ = continuity of the cumulative gradient; Δ_{GW} = coherence lost between cumulative gradient and final weight (both defined in Section 4.3). †the radial control of an earlier version of this work, numerically linear at the operating scale; ‡a genuinely nonlinear $SO(d)$-equivariant control (both explained in the text). **Bold**/underline = notably high/low. All 21 configurations and additional metrics in Appendix G.

Configuration	eq-err	Acc	$\bar{\mathbf{G}}\,\mathbf{v}_1$	W $\mathbf{v}_1$	W σ^2-WA	Δ_{GW}	s_l	W $\mathbf{u}_1$
*Equivariant (*eq-err ≈ 0*)*								
Res+None	0	.909	.851	.257	.061	+.594	.261	.260
Res+Radial†	0	.913	.851	.192	.059	+.660	.180	.191
Res+EqNL‡	0	.957	.480	.435	**.311**	+.045	.354	.548
Non-equivariant but linear								
Res+Diag	.188	.913	.857	.163	.057	+.694	.164	.179
Res+Perm	1.99	.914	.840	.138	.057	+.702	.049	.048
Non-equivariant nonlinear								
Res+Leaky(.9)	.006	.932	.845	.265	.069	+.580	.075	.272
Res+Leaky(.5)	.271	.971	.914	.532	.089	+.381	.350	.549
Res+Tanh	.635	.970	.915	.846	.283	+.069	.151	.058
Res+Leaky(.1)	8.76	.970	.956	.911	.163	+.045	.389	.929
Res+ReLU	193	.976	.961	**.959**	.249	+.003	.564	.937
Res+GELU	206	.975	.977	**.964**	.249	+.012	.535	.955
Res+SiLU	247	.976	.977	**.965**	.243	+.012	.570	.963
Normalization								
Res+None+LN	0	.924	.733	.665	**.315**	+.068	.741	.650
Res+ReLU+LN	15	.979	.899	.892	.248	+.007	.301	.912
No residual								
NoRes+None	0	.706	.050	.041	.049	+.009	.312	.055
NoRes+ReLU	4.36	.727	.106	.360	.088	−.255	.590	.647
NoRes+GELU	11	.939	.162	.225	.083	−.063	.312	.637

4.3 Condition 2: Symmetry Breaking (Activation and LayerNorm)

The critical experiment: Res+None. The most revealing experiment removes the activation function while keeping residual connections (Res+None: $\mathbf{h}_{l+1} = \mathbf{h}_l + \mathbf{W}_l\mathbf{h}_l$). This configuration achieves cumulative-gradient $\mathbf{v}_1$ continuity 0.851 (high—residual still provides coherence) but weight $\mathbf{v}_1$ continuity only 0.257, far below Res+ReLU's 0.959 (Table 1). The gap is even starker in σ^2-weighted alignment: the per-step gradient's σ^2-WA reaches 0.928 across all singular vector directions (Table 10), yet the weight's is only 0.059 (Table 11)—rotational degeneracy blocks transfer not just in the dominant direction but across the entire spectrum. This demonstrates that **gradient coherence alone is insufficient for sustained weight continuity**—a nonlinearity is needed to stabilize the transfer.

Rotational symmetry breaking. In deep *non-residual* linear networks, adjacent layers admit an exact rotational degeneracy: $\mathbf{W}_l\mathbf{W}_{l-1} = (\mathbf{W}_l\mathbf{R}^\top)(\mathbf{R}\mathbf{W}_{l-1})$ for any orthogonal $\mathbf{R}$, so individual SVDs are underdetermined and flat directions appear in the loss landscape [11]. Residual connections obstruct *insertion*-type moves—placing an $\mathbf{R}$ between adjacent factors—because the identity terms couple $\mathbf{R}$ to the product: $(\mathbf{I}+\mathbf{W}_l\mathbf{R}^\top)(\mathbf{I}+\mathbf{R}\mathbf{W}_{l-1}) \neq (\mathbf{I}+\mathbf{W}_l)(\mathbf{I}+\mathbf{W}_{l-1})$ in general. They do *not*, however, obstruct *conjugation*-type directions, which transform the identity term along with the rest: writing $\mathbf{T}_l = \mathbf{I} + \mathbf{W}_l$, the substitution $\mathbf{T}_l \mapsto \mathbf{M}_l\mathbf{T}_l\mathbf{M}_{l-1}^{-1}$ leaves the network function exactly unchanged for any invertible $\mathbf{M}_1, \ldots, \mathbf{M}_{L-1}$ (with $\mathbf{M}_0 = \mathbf{M}_L = \mathbf{I}$). With a linear activation the exactly-flat manifold therefore retains dimension $(L-1)d^2$ ($15 \cdot 256^2 \approx 9.8 \times 10^5$) *whether or not* residual connections are present—which is why Res+None still drifts. For *nonlinear* residual blocks, by contrast, Marcotte et al. [13] prove that no conservation law depends only on the two weight matrices adjacent to a skip connection (their Theorem 4.7), consistent with the absence of a relative (block-to-block) continuous symmetry once a nonlinearity is present. Res+None's weight

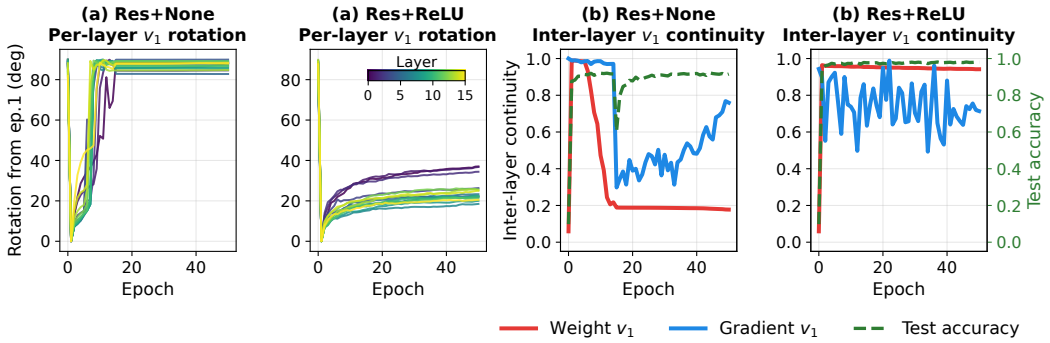


Figure 2: **Rotation drift and continuity collapse without activation** (small initialization σ=0.0001, MNIST, 50 epochs, single seed; Section 4.3). **(a)** Rotation of each layer's weight $\mathbf{v}_1$ from its epoch-1 direction (color = layer). **(b)** Inter-layer weight and gradient $\mathbf{v}_1$ continuity, with test accuracy on the right axis.

continuity, accordingly, forms transiently then collapses (Figure 2, quantified below). Any nonlinear residual branch breaks this remaining symmetry: the identity path cannot absorb $(\mathbf{M}_l\mathbf{M}_{l-1}^{-1}-\mathbf{I})\mathbf{h}$, so only a global $\mathbf{M}$ survives (absorbed by embedding and head), and each layer's weight is constrained to a shared coordinate frame (formalized below). We distinguish this direct continuity ($\mathbf{v}_l \leftrightarrow \mathbf{v}_{l+1}$) from the signal-flow alignment ($|\mathbf{v}_{l+1}^\top \mathbf{u}_l| \to 1$) of Ji and Telgarsky [8], which can dissociate.

Frame pinning, formally. Write the linear-activation chain as $f(\mathbf{x}) = \mathbf{H}\mathbf{T}_L \cdots \mathbf{T}_1\mathbf{E}\mathbf{x}$ (embedding $\mathbf{E}$, head $\mathbf{H}$). Differentiating the conjugation $\mathbf{T}_l \mapsto \mathbf{M}_l\mathbf{T}_l\mathbf{M}_{l-1}^{-1}$ at $\mathbf{M} = \mathbf{I}$ gives the function-preserving directions $\delta\mathbf{W}_l = \mathbf{A}_l\mathbf{T}_l - \mathbf{T}_l\mathbf{A}_{l-1}$ ($\mathbf{A}_l$ arbitrary, $\mathbf{A}_0 = \mathbf{A}_L = \mathbf{0}$). The loss is *exactly* constant along them, so they carry no gradient component and meet no restoring curvature: any update component along them accumulates unopposed. Adam's coordinate-wise preconditioning supplies such components although every gradient is orthogonal to them (38–86% of the update norm; Appendix I), and their relative-rotation part is what erases $\mathbf{v}_1$ alignment. With $\mathbf{h} \mapsto \mathbf{h}+\sigma(\mathbf{W}_l\mathbf{h})$ the substitution survives only if $\sigma(\mathbf{M}\mathbf{z}) = \tilde{\mathbf{M}}\sigma(\mathbf{z})$ on the activation's support, i.e. for $(\mathbf{M}, \tilde{\mathbf{M}})$ in σ's intertwiner set: for element-wise ReLU-family σ, permutations and positive rescalings [12]. Because the residual identity path forces the *same* rescaling at every interface, the continuous part shrinks to a single global diagonal (dimension d, absorbed by $\mathbf{E}$ and $\mathbf{H}$), and for non-homogeneous σ (GELU, SiLU, Tanh) it vanishes. The directions that rotate adjacent layers' frames relative to one another thus acquire nonzero loss curvature; this is pinning. Rotating a hidden frame by $\mathbf{R} = \exp(\theta\mathbf{\Omega})$ ($\mathbf{\Omega}$ skew-symmetric, $\|\mathbf{\Omega}\|_2 = 1$) costs, to second order at stationarity, $(\theta^2/2)\kappa$, with the curvature κ growing with σ's nonlinearity on the actual pre-activation distribution and, for element-wise σ, with its deviation from equivariance (eq-err). The stationary alignment balances off-gradient accumulation along the former orbit against this curvature, predicting the increasing eq-err $\to$ continuity relation of Table 1; a linear activation, whatever symmetry it breaks, reopens the flat manifold. Injected rotations erase continuity at no cost in the linear network but break Res+ReLU (Table 15).

Two-phase dynamics: formation versus retention. This drift dissociates two dynamical phases, formation and retention: Res+ReLU supports both, Res+None only the first. The small-initialization run (Section 4.1) makes the two phases observable in isolation. **Phase 1 (formation)**: at epoch 1, weight $\mathbf{v}_1$ continuity reaches 0.99 (Figure 2b, Res+None panel)—gradient-to-weight transfer *does* occur without activation, driven entirely by residual-mediated gradient coherence. **Phase 2 (retention vs. drift)**: by epoch 15, Res+None's weight continuity has collapsed to $\sim$0.19 as each layer's rotation frame drifts independently along its $SO(d)$ orbit: Res+None layers have rotated $\sim$83–90° from their epoch-1 directions, mutually misaligned; Res+ReLU layers only $\sim$15–31°, coherently, keeping $\sim$0.95 continuity (Figure 2). Crucially, **weight continuity collapses first** (epochs 5–13) and **gradient continuity follows** (abruptly, at epoch 15)—suggesting a cascade in which weight drift induces hidden-state divergence, which in turn destroys gradient coherence. The collapse coincides with a transient accuracy dip (epoch 15: 0.92 $\to$ 0.61) that recovers to 0.92 without restoring continuity (Figure 2b, dashed green): the network finds a functional but structurally different solution. Thus, in this setting, residual connections suffice for formation while retention additionally

requires a nonlinearity. The drift is causal, not merely concurrent: projecting the flat-tangent component out of every Res+None update holds $\mathbf{v}_1$ at 0.83–0.93 through the window in which the baseline collapses to 0.16 (a separate implementation of the Figure 2 setting; slow decay thereafter) and keeps accuracy at 0.85–0.90 without the transient crash; the collapse resumes once projection is released, and removing only 70% of the component merely delays it, so cumulative motion along the orbit, not a threshold event, drives the transition (Appendix I; relating this to feature-learning and edge-of-stability transitions is future work). The collapse is characterized here in the small-initialization linear setting, where $\Delta\mathbf{W}$ dominates $\mathbf{W}^{(0)}$ from the first epoch.

Nonlinearity alone is not enough. The above argument predicts that the *degree* to which an element-wise nonlinearity breaks rotational equivariance should control how strongly the frame is pinned. We make that degree measurable with an equivariance error, evaluated on the trained network's own pre-activation distribution, $\text{eq-err}(\sigma) = \mathbb{E}_{\mathbf{z},\mathbf{R}\sim\text{Haar}(O(d))}\|\sigma(\mathbf{Rz})-\mathbf{R}\sigma(\mathbf{z})\|^2/\mathbb{E}\|\sigma(\mathbf{z})\|^2$ (6 rotations, layer-averaged). Across the configurations of Table 12, eq-err tracks weight continuity closely (Spearman $+0.89$ over the 12 nonlinear residual configurations without normalization; Figure 13). Notably, the LeakyReLU family has the *same* intertwiner group for every $0 < \alpha < 1$ [12], yet continuity varies from 0.19 to 0.91 across it: the group classification is too coarse, and the continuous equivariance error is the operative variable. Accuracy is not the confounder: matched-accuracy pairs differ 3.5–5.7$\times$ in $\mathbf{v}_1$ (Table 14).

Two controls separate the two ingredients. **Nonlinearity without $SO(d)$ symmetry breaking**: a rotation-equivariant activation acts only on the magnitude, $\sigma(\mathbf{x}) = \hat{\mathbf{x}}\, h(\|\mathbf{x}\|)$. The radial control of an earlier version of this work, $\mathbf{x}\tanh(\|\mathbf{x}\|)$, is numerically the identity at the scale the network operates at (Table 12), so it does not test a nonlinear equivariant activation. We therefore use $\sigma(\mathbf{x}) = \hat{\mathbf{x}}\|\mathbf{x}\|^\alpha$, exactly $SO(d)$-equivariant and nonlinear for every $\alpha \neq 1$ ($\alpha = 1$ is the identity); at $\alpha = 0$ (Res+EqNL, nonlinearity 0.33 as defined in Table 12) weight $\mathbf{v}_1$ continuity reaches 0.435—above the linear network's 0.257, but less than half of Res+ReLU's 0.959—and in a separate sweep of α from 1 to 0 no 3-seed mean exceeds 0.45. $SO(d)$ **symmetry breaking without nonlinearity**: Res+Diag (fixed element-wise scaling) and Res+Perm (fixed permutation) have eq-err > 0 but stay at or below the Res+None level (0.163, 0.138): the layers still compose linearly, so the conjugation freedom survives whichever symmetry the activation breaks.

At matched nonlinearity (0.30–0.33) the equivariant control reaches $\mathbf{v}_1 = 0.435$ and LeakyReLU(0.1) 0.911, a 2.1$\times$ gap. The statement we therefore adopt is: *in the architectures studied, continuity requires a non-equivariant nonlinearity, and, if element-wise, its equivariance error then sets, continuously, how strongly the frame is pinned*. Neither ingredient suffices alone; only their combination reaches high continuity (Res+ReLU 0.959).

To localize the loss, we track the cumulative raw gradient $\bar{\mathbf{G}}_l = \sum_t \mathbf{G}_l^{(t)}$ and define $\Delta_{\text{GW}} = \text{cont}(\mathbf{v}_1(\bar{\mathbf{G}}_l)) - \text{cont}(\mathbf{v}_1(\mathbf{W}_l))$. For Res+ReLU, $\Delta_{\text{GW}} = 0.003$ (full transfer); for Res+None and Res+Radial it is 0.594 and 0.660—the cumulative gradient is coherent at $\sim$0.85 in both cases, but drift erases it before it reaches the weight. Res+EqNL, by contrast, transfers as fully as Res+Leaky(.1) ($\Delta_{\text{GW}} = 0.045$); its lower continuity (W $\mathbf{v}_1 = 0.435$) arises upstream, in a less coherent cumulative gradient ($\bar{\mathbf{G}}\,\mathbf{v}_1 = 0.480$; Appendix G).

Activation type matters. Different activations reduce the symmetry group to different extents [12], affecting weight continuity (Table 1). GELU, SiLU, and ReLU all reduce symmetry to permutations (and rescaling for ReLU), none of which rotate adjacent layers' frames relative to one another; at large eq-err (193–247) all three reach similar continuity ($\sim$0.96). Tanh (eq-err $= 0.635$ vs. 193 for ReLU) reaches lower $\mathbf{v}_1$ continuity (0.846), losing a little in both the cumulative gradient ($\bar{\mathbf{G}}\,\mathbf{v}_1 = 0.915$ vs. 0.961) and the transfer ($\Delta_{\text{GW}} = 0.069$ vs. 0.003); additionally, Tanh's output saturation suppresses backward gradient flow in the output direction, collapsing $\mathbf{u}_1$ to 0.058 (detailed mechanism in Appendix G). Here our metric and the rotation-invariant s_l dissociate informatively: Tanh keeps *input* frames pinned ($\mathbf{v}_1 = 0.846$) while its output saturation collapses s_l to 0.151 (ReLU family 0.54–0.57), so s_l conflates absent frame pinning with absent output-direction structure, which $\mathbf{v}_1$ and $\mathbf{u}_1$ separate; we report all three. Among the residual configurations without normalization, s_l still separates the ReLU family from the equivariant and linear controls (0.05–0.35), so the association is not an artifact of our metric's frame sensitivity. Without residual connections, however, s_l is high while $\mathbf{v}_1$ is not (NoRes+ReLU: 0.590 vs. 0.360): a plain chain hands each layer's output direction to the next layer's input by construction, which is the handoff s_l measures,

whereas weight continuity requires a frame shared along the residual stream. Linear (None) keeps the full conjugation freedom, yielding only 0.257.

LayerNorm: a different kind of symmetry breaking. LayerNorm also breaks rotational symmetry, but differently from activation (Table 1; all metrics in Appendix G). Activation concentrates continuity in $\mathbf{v}_1$ (Res+ReLU: W $\mathbf{v}_1 = 0.959$, σ^2-WA $= 0.249$), while LayerNorm distributes it across multiple directions (Res+None+LN: W $\mathbf{v}_1 = 0.665$, σ^2-WA $= 0.315$). When combined (Res+ReLU+LN), activation's $\mathbf{v}_1$-concentrated mode dominates and LN's contribution is largely absorbed (σ^2-WA $= 0.248$, essentially unchanged). The critical requirement is a **symmetry-breaking nonlinearity** of either kind; task structure modulates continuity within an architecture but does not substitute for it (Appendix H).

4.4 How Gradient Accumulation Forms Weight Structure

Gradient continuity precedes weight and reflects the long-term gradient average. Gradient $\mathbf{v}_1$ continuity is 0.95 at the first backward pass (before any weight update), while weight $\mathbf{v}_1$ starts at 0.04 and rises to 0.96 during training (Figure 12, Appendix E)—gradient continuity precedes weight continuity, consistent with a causal direction from gradient to weight. The weight is the initialization plus the accumulated optimizer update, $\mathbf{W}_l^{(T)} = \mathbf{W}_l^{(0)} + \Delta\mathbf{W}_l$; for SGD with momentum μ, $\Delta\mathbf{W}_l = -\frac{\eta}{1-\mu}\bar{\mathbf{G}}_l$ up to the last few steps, while Adam preconditions each step. Under SGD+momentum the Table 1 ordering is unchanged and accumulated-update continuity matches cumulative-gradient continuity (Res+ReLU 0.964 for both; Appendix F); under Adam (seed-42 re-run) the accumulated update's continuity matches the weight's for the pinned configurations (Res+ReLU 0.965 vs. 0.958) and is already lost for the drifting ones (Res+None 0.28 vs. ≈ 0.85 for the raw-gradient sum in Table 1): the drift acts inside the accumulation itself.

From gradient coherence to weight coherence. Combining Conditions 1 and 2 yields the full mechanism. Residual connections ensure $\mathbf{G}_l^{(t)} \approx \mathbf{G}_{l+1}^{(t)}$ (Section 4.2), so cumulative gradients are also cross-layer coherent (≥ 0.73 except Res+EqNL and Res+Abs; Table 12). Once $\|\Delta\mathbf{W}_l\|$ dominates $\|\mathbf{W}_l^{(0)}\|$, weight $\mathbf{v}_1$ converges to that of $\Delta\mathbf{W}_l$, inheriting the gradients' cross-layer coherence; a nonlinearity then stabilizes this transfer by pinning each layer to a shared coordinate frame (Section 4.3). We verify the $\|\Delta\mathbf{W}\|/\|\mathbf{W}^{(0)}\|$ prediction with Res+ReLU trained via Adam $\beta_2{=}0, \varepsilon{=}1$ (Appendix F), which produces small updates: at default Kaiming initialization, $\beta_2{=}0$ yields near-zero continuity ($\mathbf{v}_1 = 0.047$); progressively shrinking the initialization to $\sigma{=}0.0001$ recovers $\mathbf{v}_1 = 0.958$, matching Adam default (Table 8). At $\sigma{=}0.0001$ continuity is high in *all* singular directions, because $\Delta\mathbf{W}$ dominates $\mathbf{W}^{(0)}$ in every direction, while accuracy drops to 0.885: continuity reflects the *direction* of weight structure, performance also its *magnitude* (Table 8).

5 Transformer Experiments

We extend our analysis to transformers to test whether the toy MLP findings generalize and to uncover transformer-specific phenomena. We train a small Llama-style transformer ($\sim$34M parameters, 8 layers) on WikiText-103 with four activation ablations and two failure-mode controls (setup in Appendix A.1).

5.1 Projection-Specific Continuity in Transformers

Our most striking transformer finding is that **continuity is projection-specific**: each projection develops continuity in the space facing the residual stream, with its strength determined by a nearby nonlinearity (Figure 3). The Llama-style MLP computes:

$$\text{MLP}(\mathbf{x}) = \mathbf{W}_{\text{down}}\left(\text{SiLU}(\mathbf{W}_{\text{gate}}\mathbf{x}) \odot \mathbf{W}_{\text{up}}\mathbf{x}\right) \tag{4}$$

The activation is applied *only* to Gate's output. Attention computes Q, K, V linearly, with softmax providing nonlinearity in softmax$(\mathbf{Q}\mathbf{K}^\top/\sqrt{d})$.

Ablating MLP activation drops Gate continuity. Table 2 isolates the MLP-side effect: removing the MLP activation drops Gate $\mathbf{v}_1$ from 0.662 to 0.366 while only modestly affecting perplexity

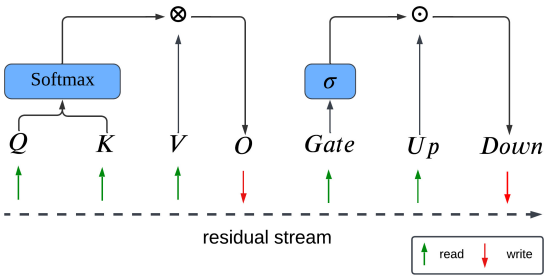


Figure 3: **Residual stream read/write structure of a transformer block.** Each projection either reads from the residual stream (green up arrows: Q, K, V, Gate, Up) or writes to it (red down arrows: O, Down). Nonlinearities (softmax, σ) are shown as blue boxes. Section 5.1 uses this structure to predict each projection's continuity space.

Table 2: Transformer continuity results. *Top*: small Llama-style transformer ablations (10K steps, init $\mathcal{N}(0, 0.02^2)$). *Middle*: pretrained Llama-3.1-8B (mean over the 31 adjacent-layer pairs). *Bottom*: removing residual or norm causes training failure.

Config	PPL	Q $\mathbf{v}_1$	K $\mathbf{v}_1$	Gate $\mathbf{v}_1$	Up $\mathbf{v}_1$	O $\mathbf{u}_1$	Down $\mathbf{u}_1$	V $\mathbf{v}_1$
Res+SiLU+Norm	41.9	0.405	0.627	**0.662**	0.225	0.129	0.554	0.209
Res+ReLU+Norm	42.8	0.398	0.671	**0.820**	0.308	0.339	0.532	0.188
Res+GELU+Norm	41.9	0.370	0.609	**0.804**	0.350	0.147	0.507	0.115
Res+None+Norm	43.2	0.492	0.668	0.366	0.359	0.221	0.517	0.237
Llama-3.1-8B	—	**0.82**	**0.77**	**0.80**	0.61	**0.89**	**0.70**	0.13
NoRes+SiLU+Norm	1700	0.036	0.062	0.045	0.036	0.066	0.052	0.054
NoNorm (any)	NaN	—	—	—	—	—	—	—

($41.9 \to 43.2$, $\sim 3\%$ increase). The attention-side projections (Q, K, V) are architecturally independent of the MLP activation, so their continuity changes far less; isolating the softmax mechanism for Q and K would require architectural modification and is left to future work.

Perplexity stays near baseline because the bilinear Gate$\odot$Up multiplication and RMSNorm remain active; removing residual or RMSNorm (Table 2, bottom) causes training failure, underscoring that activation is dispensable for function while residual and norm are not.

A nonlinearity on V creates V continuity. V, the one projection without a nonlinearity on its own coordinates, admits a causal test of the post-nonlinearity account (Section 5.2): a SiLU on its output (3-seed re-run of the Res+SiLU+Norm setting of Table 2) creates V continuity ($\mathbf{u}_1$ $0.04 \to 0.47$, $\mathbf{v}_1$ $0.20 \to 0.33$) at no perplexity cost beyond seed variation ($42.25 \pm 0.22 \to 42.36 \pm 0.37$).

V and O both show modest continuity in this small-transformer regime, but O $\mathbf{u}_1$ sharpens substantially in pretrained Llama-3.1-8B ($\mu = 0.89$; Section 5.2)—suggesting that output-space continuity for O consolidates with scale and longer training, while V's low continuity persists across scales.

5.2 Projection-Specific Pattern in Pretrained LLMs

The full projection-specific pattern emerges clearly in pretrained Llama-3.1-8B [5] (32 layers, d=4096; Table 2, Appendix A): Q, K, and Gate show high and Up intermediate $\mathbf{v}_1$ mean adjacent-layer cosine similarity ($\mu = 0.82, 0.77, 0.80, 0.61$) while V shows the lowest ($\mu = 0.13$); O and Down show $\mathbf{u}_1$ continuity ($\mu = 0.89, 0.70$). At small scale, several projections (notably O and Up) are still consolidating—only Gate, K, and Down exceed 0.5 (Table 2)—while V remains low at both scales. CNN ablation, vision-model, and per-head analyses are in Appendices C and D.

These patterns are consistent with our mechanism. Q and K develop input-space ($\mathbf{v}_1$) continuity because softmax acts on their output in QK-space. Gate develops input-space continuity from the MLP activation. Up shows intermediate input-space continuity because the activated Gate masks its

gradient via element-wise multiplication: Up's gradient is suppressed where $\sigma(g) \approx 0$, while Gate's gradient is active whenever $\sigma'(g)$ is nonzero. This asymmetric masking is a plausible contributor, though the precise mechanism warrants dedicated analysis. O and Down develop output-space ($\mathbf{u}_1$) continuity, which we attribute to the post-nonlinearity signals they receive. V shows neither, as no nonlinearity acts on its own coordinates; adding one creates V continuity (Section 5.1).

Four of five additional models spanning 1.5B–70B parameters show the same pattern; in Gemma-3-12B, O and Down show little output-space structure (Appendix B).

6 Discussion

Projection-specific nonlinearity determines projection-specific structure. The transformer experiments reveal a precise correspondence: each projection's continuity appears in the space—input ($\mathbf{v}_1$) or output ($\mathbf{u}_1$)—that faces the residual stream, and is stabilized by the nonlinearity acting on the projection (softmax for Q/K, the MLP activation for Gate, gated multiplication for Up, post-nonlinearity signals for O and Down); V, which lacks any direct nonlinearity, shows only low continuity unless one is added (Section 5.1). This provides a mechanistic explanation for the space-specific continuity we observe in pretrained LLMs (Appendix A).

Practical implications. The projection-specific pattern identifies where each matrix family's cross-layer shared structure lives—the input space for Q, K, Gate, and Up, the output space for O and Down, and, for V, only the composite $\mathbf{W}_O\mathbf{W}_V$ (Appendix A)—and hence the coordinate system that layer pruning [3, 4] and cross-layer parameter sharing [2] would have to respect. Whether exploiting it yields practical gains is an open empirical question (a first weight-sharing probe is reported in Appendix G). The V+SiLU intervention (Section 5.1) shows that the geometry can also be controlled at the architecture level, creating V continuity at no perplexity cost.

Limitations. Our mechanistic claim for projection-specific continuity rests on two direct interventions (removing the Gate activation; adding a SiLU to V); the Q/K claims are indirect since softmax cannot be ablated without changing the attention mechanism. The drift, rotation and projection runs (Figure 2, Appendix I) are single-seed, as are the small (34M) transformer's ablations (Table 2); the same patterns appear in pretrained Llama-3.1-8B and in most models up to 70B (Appendices A–B), but we have not verified the training dynamics (rotation drift) at those scales, and transformer seed variance is characterized only for the V+SiLU intervention. The absolute scale of the continuity metrics is configuration dependent, so we compare them only between conditions (Appendix G).

7 Conclusion

We investigated the origin of geometric continuity in weight matrices through controlled experiments on toy MLPs, with validation in a small trained transformer and pretrained LLMs. Two necessary conditions emerge: residual connections provide cross-layer gradient coherence, and symmetry-breaking nonlinearities constrain all layers to a shared coordinate frame, preventing rotation drift that would otherwise destabilize weight structure. In a linear network, continuity does not stably develop; even when it forms transiently under small initialization, it collapses as layers' rotation frames drift apart. In the architectures studied, continuity requires a non-equivariant nonlinearity—a genuinely nonlinear yet $SO(d)$-equivariant control reaches less than half of ReLU's continuity—and, if element-wise, its equivariance error then sets, continuously, how strongly the frame is pinned. In transformers, the same mechanism is consistent with the projection-specific pattern of Section 5: continuity in the space that faces the residual stream, and low continuity for V.

Future work. Our framework opens three directions. (1) **Designing continuity profiles**: partial symmetry breakers (activations preserving specific rotational subgroups) could engineer targeted continuity patterns, concentrating structure in $\mathbf{v}_1$ or distributing it across directions on demand. (2) **Drift as a forgetting indicator**: rotation drift during fine-tuning may diagnose loss of pretrained structure and signal catastrophic forgetting. (3) **Orbit-aware model merging**: projecting fine-tuned checkpoints that occupy different points of a shared $SO(d)$ orbit onto a common representative could reduce interference in merging and souping.

Acknowledgments and Disclosure of Funding

Funding. This work was conducted at Hyntel. GPU resources were provided by Hyntel Inc. and Hyperintelligence Inc. (a company sharing its chief executive with Hyntel); no author received other third-party funding or support (personnel, data collection or processing, or equipment) in the 36 months before submission. **Competing interests.** K. Jeong and H. Suh are employed by Hyntel; W.-G. Paeng carried out this work at Hyntel and is now affiliated with Neurometry and Hyntel. The authors declare no competing financial or non-financial interests related to the submitted work.

References

[1] Zeping Min and Xinshang Wang. DOCS: Quantifying weight similarity for deeper insights into large language models. In *International Conference on Learning Representations*, 2025.

[2] Jingcun Wang, Yu-Guang Chen, Ing-Chao Lin, Bing Li, and Grace Li Zhang. Basis sharing: Cross-layer parameter sharing for large language model compression. In *International Conference on Learning Representations*, 2025.

[3] Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations*, 2025.

[4] Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

[5] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[6] Jianing Li and Vardan Papyan. Residual alignment: Uncovering the mechanisms of residual networks. In *Advances in Neural Information Processing Systems*, 2023.

[7] Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Nikolai Gerasimenko, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. Your transformer is secretly linear. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

[8] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019.

[9] Pierre Marion, Yu-Han Wu, Michael E Sander, and Gérard Biau. Implicit regularization of deep residual networks towards neural ODEs. In *International Conference on Learning Representations*, 2024.

[10] Daniel Beaglehole, Ioannis Mitliagkas, and Atish Agarwala. Feature learning as alignment: a structural property of gradient descent in non-linear neural networks. *Transactions on Machine Learning Research*, 2024.

[11] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014.

[12] Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. On the symmetries of deep learning models and their internal representations. In *Advances in Neural Information Processing Systems*, 2022.

[13] Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyré. Transformative or conservative? Conservation laws for ResNets and Transformers. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

[16] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

[17] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[18] Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[19] LG AI Research. EXAONE 4.0: Unified large language models integrating non-reasoning and reasoning modes. *arXiv preprint arXiv:2507.11407*, 2025.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[22] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[23] Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

[24] DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[25] Team OLMo. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

[26] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[27] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. SmolLM – blazingly fast and remarkably powerful. Hugging Face blog, `https://huggingface.co/blog/smollm`, 2024.

[28] Raymond Li, Loubna Ben Allal, Yangtian Zi, et al. StarCoder: may the source be with you! *Transactions on Machine Learning Research*, 2023.

[29] Loubna Ben Allal, Raymond Li, Denis Kocetkov, et al. SantaCoder: don't reach for the stars! *arXiv preprint arXiv:2301.03988*, 2023.

[30] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, et al. The Falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[31] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[32] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[33] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[35] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference (EUSIPCO)*, 2007.

A Geometric Continuity in Pretrained LLMs

A.1 Experimental Setup

Model and Weight Matrices. We analyze **Llama-3.1-8B** [5], a 32-layer transformer with hidden dimension $d = 4096$ and intermediate dimension $d_{\text{ff}} = 14336$. We extract seven weight matrices per layer:

- Attention: $\mathbf{W}_Q \in \mathbb{R}^{4096\times4096}$, $\mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{1024\times4096}$ (GQA with 8 KV heads), $\mathbf{W}_O \in \mathbb{R}^{4096\times4096}$
- MLP: $\mathbf{W}_{\text{up}}, \mathbf{W}_{\text{gate}} \in \mathbb{R}^{14336\times4096}$, $\mathbf{W}_{\text{down}} \in \mathbb{R}^{4096\times14336}$

Transformer training (Section 5). We train a small Llama-style transformer (D=256, 8 layers, 4 heads, SiLU + RMSNorm, ∼34M parameters) on WikiText-103 [15] using the GPT-2 tokenizer. All linear and embedding layers are initialized with $\mathcal{N}(0, 0.02^2)$, matching standard LLM practice. Blocks are pre-norm, with a gated MLP of width 1024 and no biases. Training uses Adam (β_1=0.9, β_2=0.999, constant learning rate 6×10^{-4}, gradient-norm clipping at 1) for 10,000 steps with batch size 128, sequence length 128, seed 42; perplexity is measured on the WikiText-103 validation split. We run 4 activation ablations and 2 failure-mode controls (Table 2); the no-normalization control comes from an earlier, 6,000-step version of this setup, in which both no-normalization runs (SiLU and linear activation) diverged.

Sign Alignment. Singular vectors are only defined up to sign: if $\mathbf{W}\mathbf{v} = \sigma\mathbf{u}$, then $\mathbf{W}(-\mathbf{v}) = -\sigma\mathbf{u}$, so both $(\mathbf{u}, \mathbf{v})$ and $(-\mathbf{u}, -\mathbf{v})$ are valid singular vector pairs. Since SVD algorithms return signs arbitrarily, we apply **sequential sign alignment** to ensure consistent orientation across layers. For each layer $l > 0$, we flip the sign if the dot product with the previous layer is negative:

$$\mathbf{v}_1^{(l)} \leftarrow \text{sign}\left(\langle \mathbf{v}_1^{(l)}, \mathbf{v}_1^{(l-1)} \rangle\right) \cdot \mathbf{v}_1^{(l)} \tag{5}$$

This alignment is essential for visualizing smooth trajectories in PCA plots, as arbitrary sign flips would create artificial discontinuities in the visualization. (The scalar continuity metric defined in Section 3 is sign-invariant by construction, via the absolute value, so it does not rely on this alignment.)

A.2 Geometric Continuity

PCA Visualization. To analyze the geometric structure of principal vectors across layers, we perform PCA on the collection $\{\mathbf{v}_1^{(l)}\}_{l=0}^{31}$. Note that this constitutes a *second* SVD: the first extracts principal directions from each weight matrix, while the second (PCA) analyzes how these directions are distributed across layers. Specifically, we: (1) L2-normalize each vector, (2) center by subtracting the mean, and (3) apply SVD to obtain principal components. The explained variance ratio $\text{EVR}_i = \sigma_i^2 / \sum_j \sigma_j^2$ quantifies what fraction of the *cross-layer variance* is captured by each PC. As a baseline, 32 random unit vectors in $\mathbb{R}^{4096}$ yield expected 3-PC EVR of $\sim 10\%$. Thus, observed EVR $> 50\%$ in 3 PCs indicates that the layer-wise vectors lie on a low-dimensional manifold—far from random.

The 3D PCA visualizations (Figures 4 and 5) reveal space-specific continuity. Different projections exhibit geometric continuity in different spaces: **right singular vectors** ($\mathbf{v}_1$) for Q, K, Gate, Up—these projections show continuity in input space, reflecting how they "read" from the residual stream; **left singular vectors** ($\mathbf{u}_1$) for O, Down—these projections show continuity in output space, reflecting how they "write" to downstream computations. The value projection shows low continuity in both spaces, motivating composite $\mathbf{W}_O\mathbf{W}_V$ analysis.

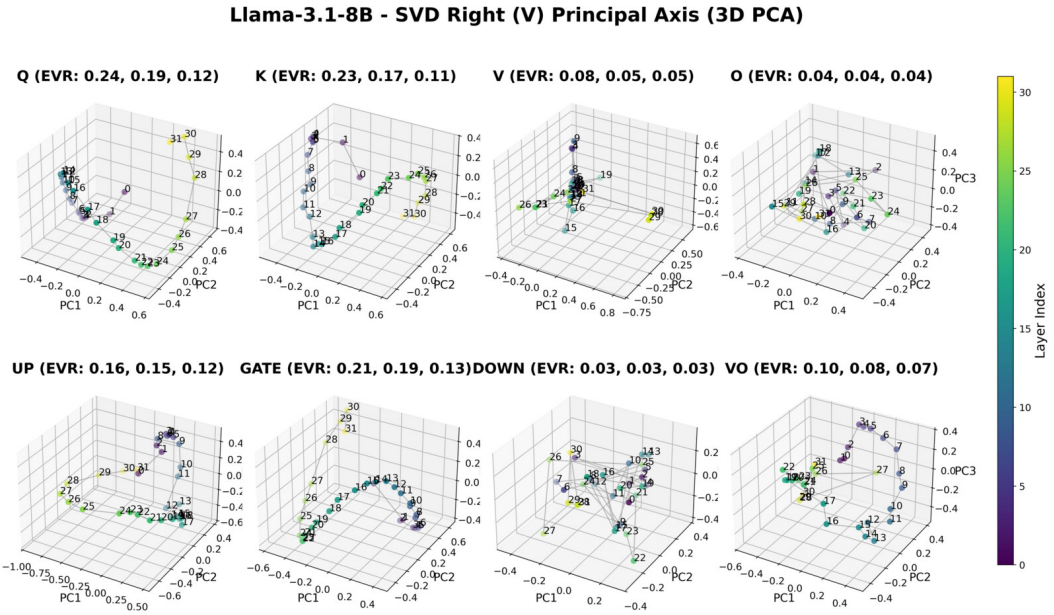


Figure 4: 3D PCA of principal **right singular vectors** ($\mathbf{v}_1$) across 32 layers. Colors indicate layer index (dark purple to yellow). Q, K, Up, Gate show smooth trajectories; V, O, Down are scattered; OV composite ($\mathbf{W}_O\mathbf{W}_V$) shows moderate structure.

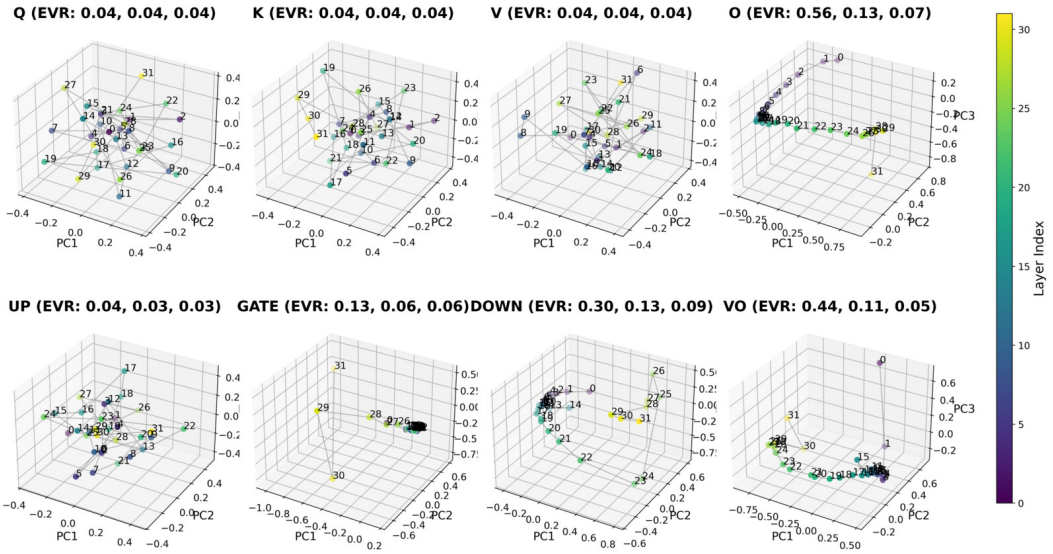


Figure 5: 3D PCA of principal **left singular vectors** ($\mathbf{u}_1$) across 32 layers. Colors indicate layer index (dark purple to yellow). O, Down show smooth trajectories; Q, K, V, Up are scattered—opposite to Figure 4—and Gate keeps moderate structure; OV composite ($\mathbf{W}_O\mathbf{W}_V$, labeled VO) shows high continuity similar to O.

Right Singular Vectors (Figure 4). Q, K, Up, and Gate form smooth curved trajectories with high explained variance (Q: 55%, K: 51%, Up: 42%, Gate: 53% in first 3 PCs), confirming consistent input-space geometry across layers. V shows weak structure (19% EVR), notably lower than Q/K/Up/Gate but higher than O/Down. O and Down show scattered, unstructured distribu-

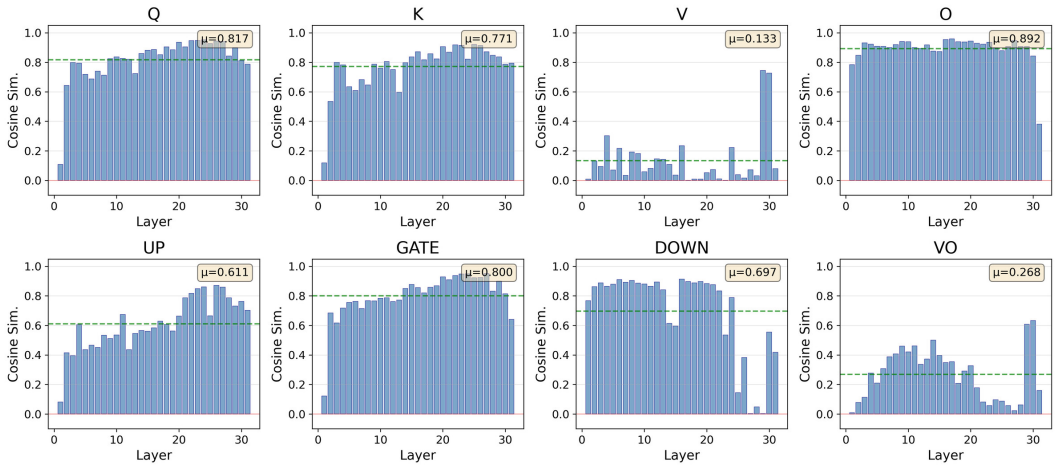


Figure 6: Layer-wise continuity (cosine similarity) between adjacent principal singular vectors. Each projection uses its high-continuity space: right vectors ($\mathbf{v}_1$) for Q, K, Up, Gate; left vectors ($\mathbf{u}_1$) for O, Down; V and VO use right vectors (VO is more continuous in its left vectors, $\mu = 0.80$; Figure 5). High-continuity projections maintain $\mu > 0.75$, while V ($\mu = 0.13$) shows only weakly aligned adjacent vectors.

tions (EVR $\approx 11\%$), indicating layer-specific principal input directions. The OV composite shows moderate structure (24% EVR), indicating partial recovery from V's lack of coherence.

Left Singular Vectors (Figure 5). O exhibits a remarkably dominant first PC (EVR $= 0.56$) with total 76% in 3 PCs, forming a nearly linear progression from layer 0 to 31. Down shows clear trajectory structure (52% EVR) in contrast to its scattered right-vector pattern. Q, K, V, and Up are now scattered (EVR $\approx$ 10–12%), opposite to their right-vector behavior; Gate keeps moderate output-space continuity (adjacent-layer $\mu = 0.57$). The OV composite exhibits high continuity (61% EVR) with a smooth trajectory similar to O, demonstrating geometric recovery through composition.

V Projection. The value projection $\mathbf{W}_V$ shows only low continuity in its input space ($\mu = 0.13$, against a random level of 0.0125) and none in its output space ($\mu = 0.02$, the random level), consistent with the main text finding that V lacks direct nonlinearity (Section 5.1).

Layer-wise Continuity. The cosine similarity analysis (Figure 6) reveals three groups: **high-continuity projections** (O: $\mu = 0.89$, Q: 0.82, Gate: 0.80, K: 0.77), **intermediate** (Down: 0.70, Up: 0.61), and **low-continuity** (V: 0.13). Continuity is measured as defined in Section 3.

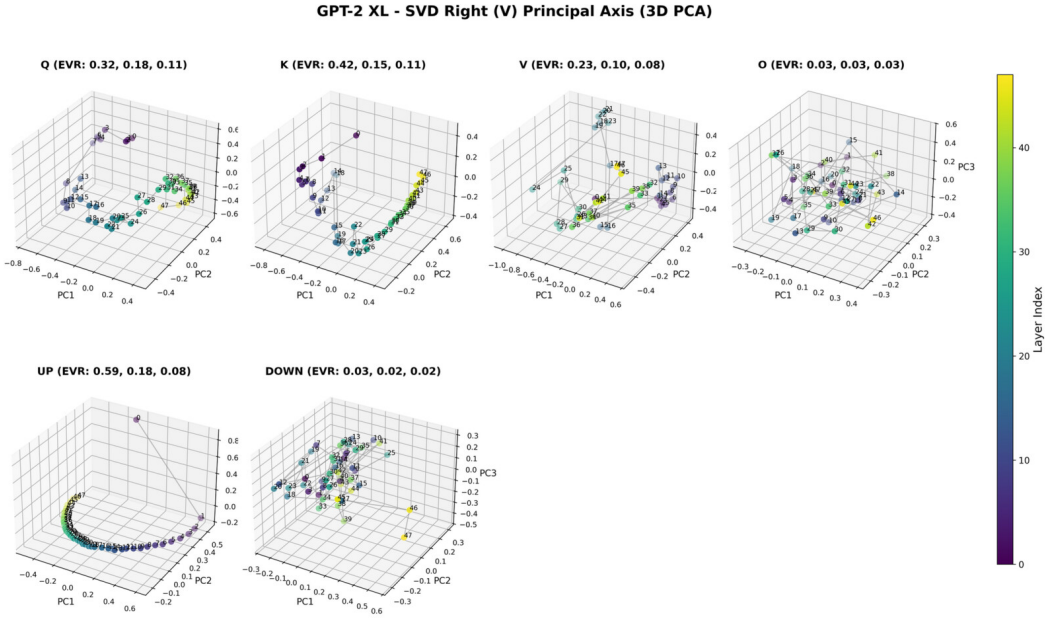
Space-Specific Continuity. The central observation is **space-specific continuity**: projections show high continuity in one space (input or output) but not the other, with Gate, which is also moderately continuous in its output space ($\mu = 0.57$), the one partial exception. Q, K, Gate, Up show input-space ($\mathbf{v}_1$) continuity; O, Down show output-space ($\mathbf{u}_1$) continuity; V shows neither. In the main text (Section 5.1), we trace this pattern to the nonlinearity acting on each projection: softmax for Q/K, SiLU for Gate, and no nonlinearity for V.

B Additional Transformer Models

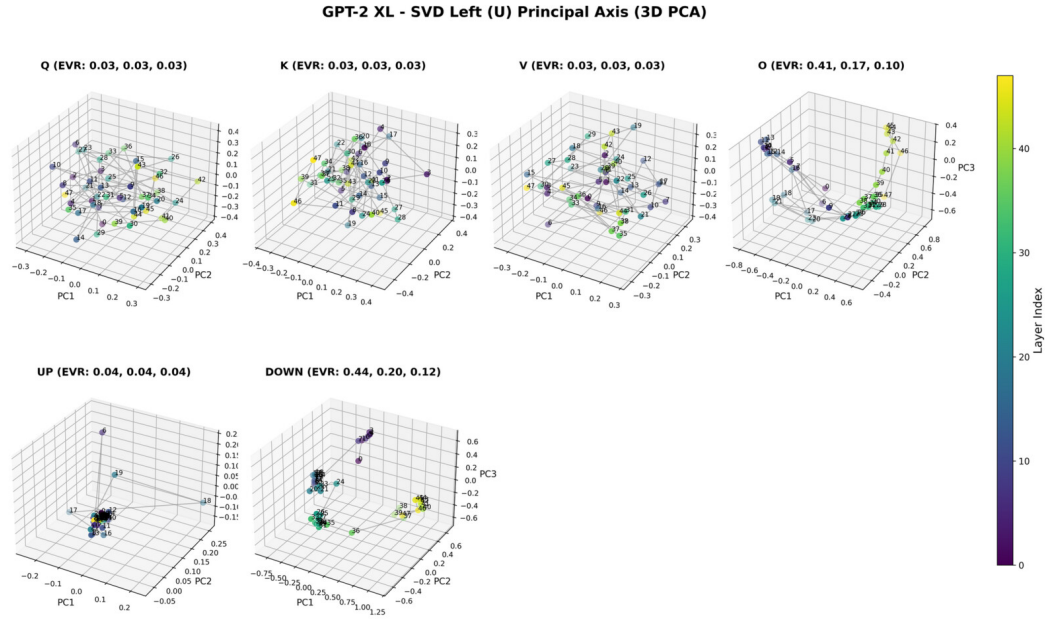
We replicate our analysis on five additional models spanning 1.5B–70B parameters. Four exhibit the same space-specific continuity pattern; Gemma-3-12B is the exception on the output side (below).

GPT-2 XL (1.5B). OpenAI [16], GELU activation, no gating (Figure 7).

Qwen3-8B (8B). Alibaba [17], SwiGLU, grouped-query attention (Figure 8).

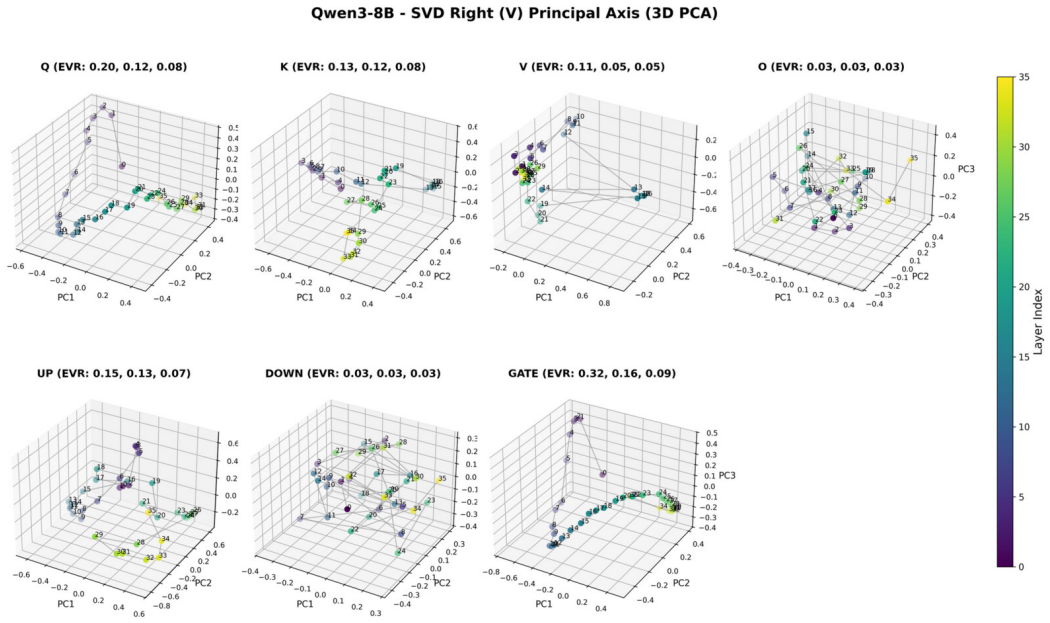


(a) Right singular vectors ($\mathbf{v}_1$)

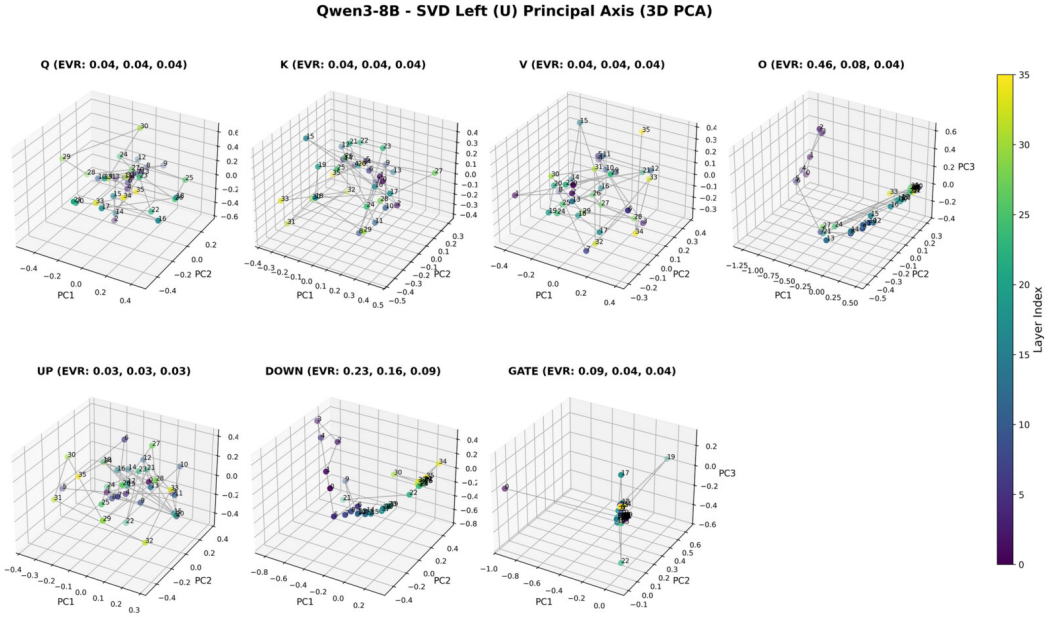


(b) Left singular vectors ($\mathbf{u}_1$)

Figure 7: **GPT-2 XL** geometric continuity.



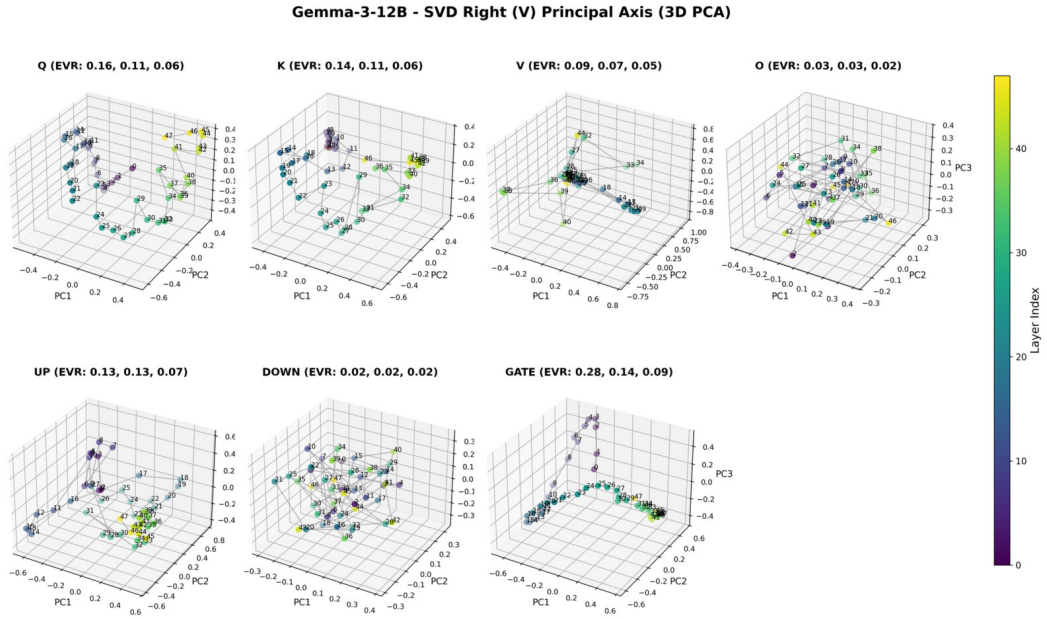
(a) Right singular vectors ($\mathbf{v}_1$)



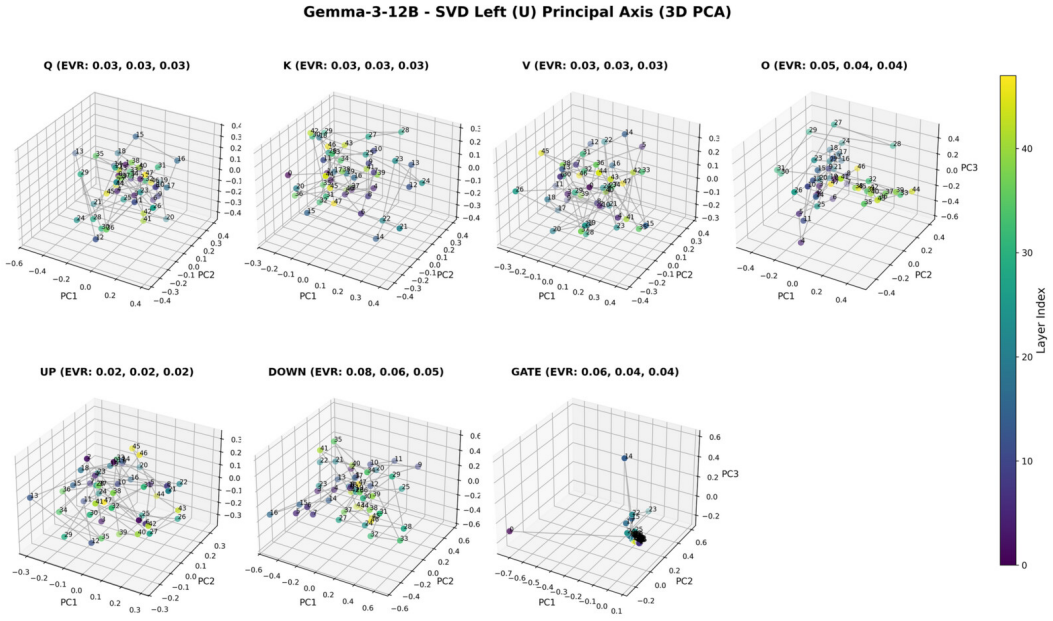
(b) Left singular vectors ($\mathbf{u}_1$)

Figure 8: **Qwen3-8B** geometric continuity.

Gemma-3-12B (12B). Google [18], 48 layers, alternating local/global attention (Figure 9).



(a) Right singular vectors ($\mathbf{v}_1$)



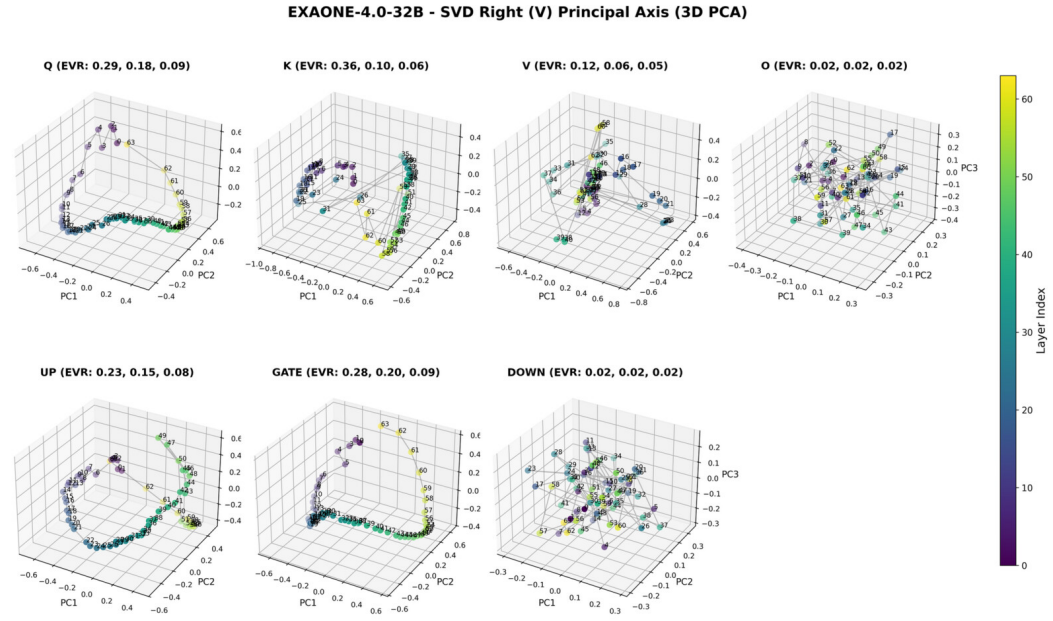
(b) Left singular vectors ($\mathbf{u}_1$)

Figure 9: **Gemma-3-12B** geometric continuity.

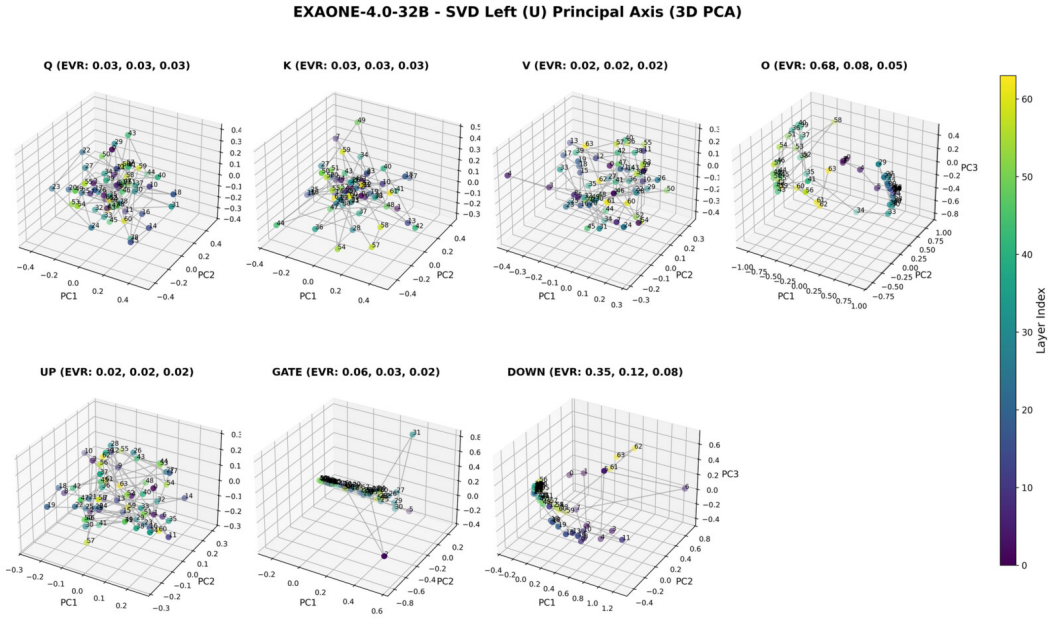
EXAONE-4.0-32B (32B). LG AI Research [19], trilingual (Korean, English, Spanish; Figure 10).

Llama-3.1-70B (70B). Meta [5], same architecture as 8B at larger scale (Figure 11).

Output-space continuity of O and Down holds in four of the five models: in Gemma-3-12B their left singular vectors are scattered (first three PCs explain 13% and 19%, Figure 9b, versus 31–81% in the other four models).

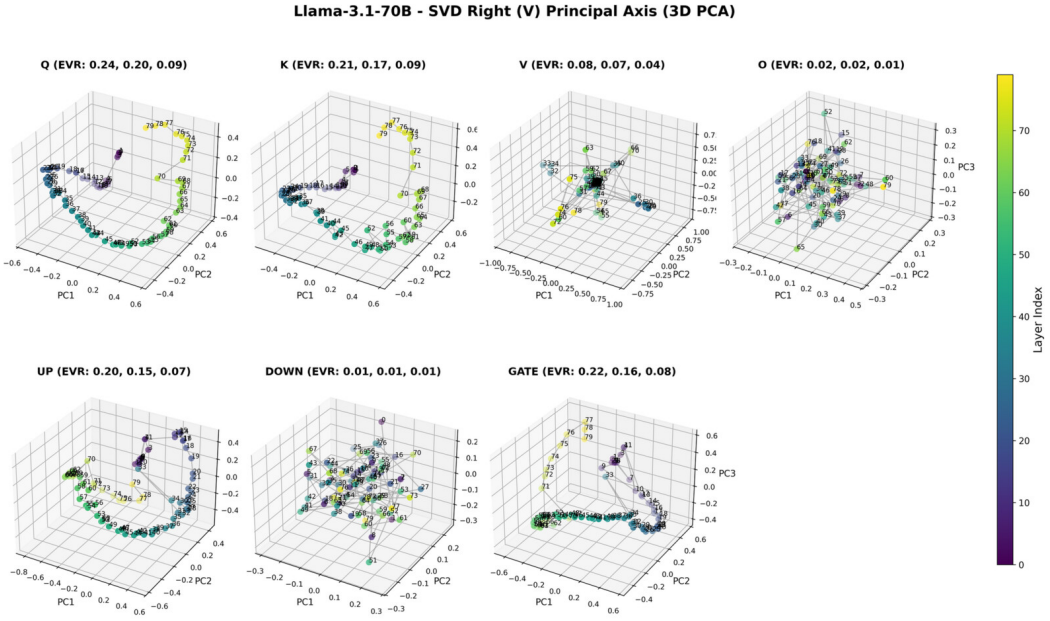


(a) Right singular vectors ($\mathbf{v}_1$)

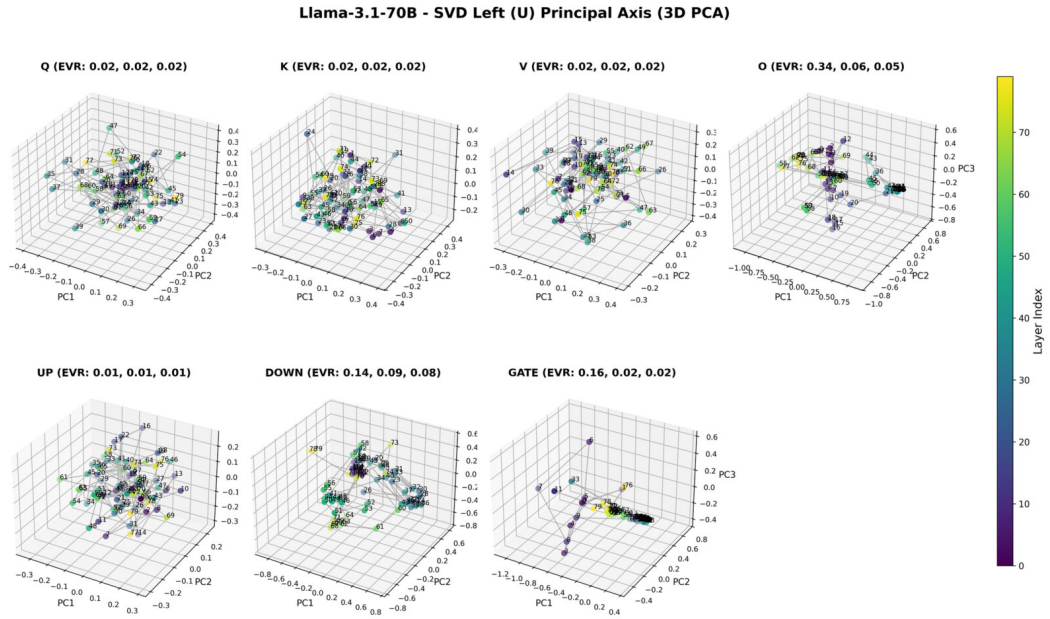


(b) Left singular vectors ($\mathbf{u}_1$)

Figure 10: **EXAONE-4.0-32B** geometric continuity.



(a) Right singular vectors ($\mathbf{v}_1$)



(b) Left singular vectors ($\mathbf{u}_1$)

Figure 11: **Llama-3.1-70B** geometric continuity.

C Other Architectures: Convolutional and Vision Models

We extend the two lines of evidence—the controlled ablation of Section 4 and the projection-specific pattern of Section 5—beyond MLPs and language models.

Trained CNN ablation. We repeat the core ablation on CIFAR-10 with a 3-stage residual CNN (widths $96/192/384$, four 3×3 blocks per stage, no normalization, as in the MLP; Adam 10^{-3}, batch 256, bf16, random-crop and horizontal-flip augmentation, 15 epochs, 2 seeds). Continuity is measured on the block convolutions reshaped to $(C, 9C)$, as the $\mathbf{v}_1$ cosine between adjacent same-stage blocks (9 pairs). The Table 1 ordering replicates on convolutions (Table 3). Accuracy co-varies with continuity here because Res+None CNNs barely train; the accuracy-matched dissociation is established in the MLP experiments (Section 4.3).

Table 3: CIFAR-10 residual CNN ablation (2 seeds, mean). Block-level $\mathbf{v}_1$ continuity within stages.

Configuration	Acc	W $\mathbf{v}_1$
Res+ReLU	.877	**.671**
NoRes+ReLU	.746	.383
Res+None	.358	.177

Pretrained ViT-Base/16. The projection-specific pattern replicates in a vision transformer [20] (Hugging Face `google/vit-base-patch16-224`; ImageNet-21k pretraining, ImageNet-1k fine-tuning; 12 layers; Table 4). Projections that read from the residual stream (Q, K, MLP-in) show input-space ($\mathbf{v}_1$) continuity; those that write to it (O, MLP-out) show output-space ($\mathbf{u}_1$) continuity; and V is the lowest of the stream-reading projections in its input-side metric (0.29 vs. 0.59–0.71 for Q/K), as in the language models of Section 5.

Table 4: ViT-Base/16, adjacent-layer continuity over 12 layers. **Bold** marks each projection's stream-facing space.

	Q	K	V	O	MLP-in	MLP-out
$\mathbf{v}_1$	**.59**	**.71**	**.29**	.03	**.87**	.13
$\mathbf{u}_1$	.03	.03	.03	**.59**	.86	**.49**

Pretrained ResNet-50. In a bottleneck ResNet [21] (Hugging Face `microsoft/resnet-50`, ImageNet-1k), the stream-reading 1×1 convolutions show $\mathbf{v}_1 = 0.59$ between adjacent same-stage blocks—7–47$\times$ the random-alignment level of each stage (0.050–0.018 for 256–2048 input channels)—increasing with stage depth ($0.35 \to 0.45 \to 0.73 \to 0.83$ over the four stages), while the internal 3×3 convolutions, which (unlike the 1×1 convolutions) have no adjacent contact with the residual stream, show 0.09 (Table 5). The stream-writing 1×1 convolutions, however, deviate from the LayerNorm-transformer pattern: their output-space continuity is at the baseline level ($\mathbf{u}_1 = 0.03$). Whether this reflects BatchNorm absorbing output-side structure—consistent with Section 4.3's finding that normalization redistributes continuity across directions—is an open question that we do not resolve here.

Table 5: ResNet-50, adjacent same-stage block continuity (mean over stages; per-stage $\mathbf{v}_1$ in brackets).

Convolution	Role	$\mathbf{v}_1$	$\mathbf{u}_1$
1×1 in	reads stream	**.59** [.35, .45, .73, .83]	.56
3×3 mid	internal	.09 [.07, .01, .07, .23]	.23
1×1 out	writes stream	.18 [.03, .13, .22, .35]	.03

D Head-Level Continuity

Section 5 measures each projection as a whole matrix. Because grouped-query and multi-query attention could hide head-level structure, we repeat the analysis per head. For a projection with H heads of width h_d, head h of layer l is the row block $\mathbf{W}_{l,h} \in \mathbb{R}^{h_d \times d}$ (Q, K, V; input-space $\mathbf{v}_1$) or the column block (O; output-space $\mathbf{u}_1$), and we compare adjacent layers. Head indices carry no canonical correspondence across layers (permuting the heads of a layer, with the matching permutation of O's columns, leaves the network unchanged), so index-matched comparison is only a lower bound; we therefore report the *best match* (Table 6): for each head of layer l, the maximum $|\cos|$ over the heads of layer $l{+}1$. Selecting a maximum inflates the chance level, so every value is accompanied by its chance level, simulated with random unit vectors under the same procedure at the same (H, d) (e.g. 0.071 for $H{=}12$, $d{=}768$). Models span the four head structures in use: multi-head (MHA), grouped-query (GQA) [22], multi-query (MQA) [23], and multi-head latent attention (MLA) [24]. The checkpoints (Hugging Face, used without training, one per model) are GPT-2 and GPT-2 XL [16], OLMo-2-0425-1B [25], Llama-3.1-8B [5], Qwen2.5-1.5B-Instruct [26], Qwen3-0.6B and Qwen3-4B [17], SmolLM-135M [27], tiny_starcoder_py [28], SantaCoder [29], Falcon-7B [30], and DeepSeek-V2-Lite [24].

Table 6: Head-level adjacent-layer continuity (best match; chance multiple in parentheses). H/G = query heads / key-value heads; Q and O are compared over the H query heads, K and V over the G key-value heads (MLA: see ^{a,b}). MQA has a single K and V head, so best match reduces to the index match. ^{a}MLA reads the stream through a single shared projection (`kv_a`); ^{b}per-head K/V live in the 512-dimensional latent space, where chance is 0.09.

Model	Type	H/G	Q	K	V	O
GPT-2	MHA	12/12	0.44 (6×)	0.42 (6×)	0.17 (2×)	0.67 (9×)
GPT-2 XL	MHA	25/25	0.48 (8×)	0.57 (10×)	0.22 (4×)	0.41 (7×)
OLMo-2 1B	MHA	16/16	0.38 (8×)	0.38 (8×)	0.30 (7×)	0.07 (1.5×)
Llama-3.1-8B	GQA	32/8	0.37 (10×)	0.51 (18×)	0.09 (3×)	0.21 (6×)
Qwen2.5-1.5B	GQA	12/2	0.22 (4×)	0.18 (6×)	0.07 (2×)	0.23 (5×)
Qwen3 0.6B	GQA	16/8	0.26 (4×)	0.26 (5×)	0.16 (3×)	0.20 (3×)
Qwen3 4B	GQA	32/8	0.29 (6×)	0.26 (7×)	0.11 (3×)	0.15 (3×)
SmolLM 135M	GQA	9/3	0.35 (5×)	0.43 (8×)	0.09 (2×)	0.22 (3×)
tiny_starcoder	MQA	12/1	0.45 (6×)	0.56 (20×)	0.09 (3×)	0.24 (3×)
SantaCoder 1.1B	MQA	16/1	0.34 (7×)	0.21 (12×)	0.09 (5×)	0.22 (5×)
Falcon 7B	MQA	71/1	0.38 (10×)	0.32 (26×)	0.03 (2×)	0.11 (3×)
DeepSeek-V2-Lite	MLA	16/shared	0.34 (7×)	0.44 (25×)^{a}	0.09 (1×)^{b}	0.33 (7×)

Findings. (i) Q heads are continuous at head level in every architecture (4–$10\times$ chance), as are K heads in MHA, GQA, and MQA (5–$26\times$; for MLA see (iv)); the continuity of Section 5 is not an artifact of whole-matrix aggregation. (ii) V is the least continuous stream-reading projection at head level in all 12 models; at whole-matrix level this holds in all but OLMo-2 (V 0.53, Q 0.42). (iii) The ordering among Q, K, and O is architecture-dependent: GPT-2's O-dominated ordering ($\mathrm{O} > \mathrm{Q} \approx \mathrm{K} \gg \mathrm{V}$) does not generalize, and O ranges from $9\times$ chance (GPT-2) to $1.5\times$ (OLMo-2, whose attention output passes through a normalization layer before the residual addition—consistent with the normalization effect of Section 4.3). (iv) In MLA the shared stream-reading projection is strongly continuous ($25\times$ chance) while the per-head latent K/V are at chance: the latent space has no shared frame across layers. (v) Best-match and whole-matrix values have different chance levels and are not directly comparable; where the whole matrix is strongly continuous the head-level value is lower (Llama-3.1-8B: Q $0.82 \rightarrow 0.37$, O $0.89 \rightarrow 0.21$), but it is higher for Q and K in Qwen2.5-1.5B and for O in OLMo-2 and tiny_starcoder_py. (vi) Qwen2.5-1.5B has the lowest Q and K continuity of the 12 models, already at whole-matrix level (Q 0.20, K 0.14, versus 0.42–0.87 and 0.21–0.89 in the other models); the Qwen3 models do not share this, and we have not identified the cause.

E Gradient Accumulation Mechanism

Figure 12 tracks the temporal evolution underlying the claims in Section 4.4. Gradient $\mathbf{v}_1$ continuity is already high at the first backward pass ($\sim$0.95, before any weight update; 0.81–0.99 across residual configurations and seeds, versus 0.36–0.37 for NoRes+ReLU), while weight $\mathbf{v}_1$ continuity rises from 0.04 to 0.96 over training, and the final weight $\mathbf{v}_1$ aligns most strongly with the long-term gradient EMA (about 0.5 for the raw gradient, rising to $\sim$0.85 at β=0.999; Figure 12b).

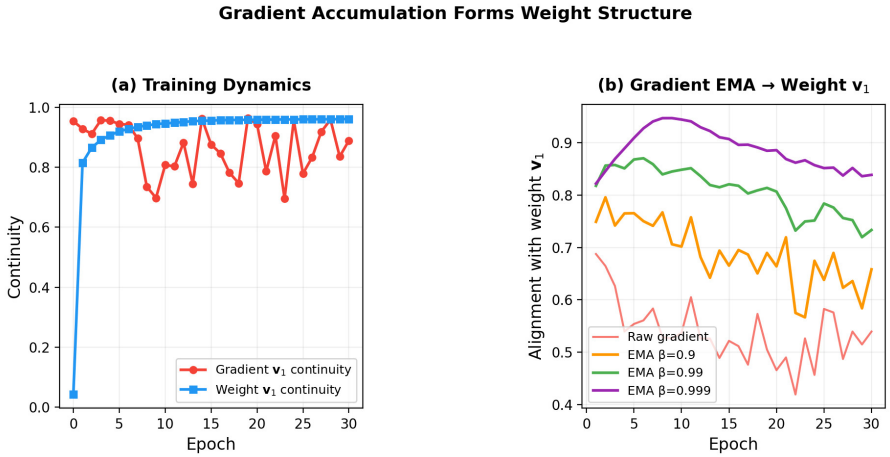


Figure 12: **Gradient accumulation forms weight structure.** (a) Weight $\mathbf{v}_1$ continuity starts at 0.04 and rises to 0.96 during training, while gradient $\mathbf{v}_1$ continuity is already high at the first backward pass, consistent with a causal direction from gradient to weight (seed 42, 30 epochs). (b) Weight $\mathbf{v}_1$ aligns most strongly with the long-term gradient average (EMA β=0.999: alignment $\sim$0.85), confirming that weight structure reflects cumulative gradient history.

F Optimizer and Depth Ablations

SGD replication. Section 4.4 relates weights to the accumulated update $\Delta\mathbf{W}_l$; under Adam this is a preconditioned quantity, so we repeat the key ablations with plain SGD+momentum (lr 0.05, momentum $\mu = 0.9$, gradient-norm clipping at 1, 2 seeds), where $\Delta\mathbf{W}_l = -\frac{\eta}{1-\mu}\bar{\mathbf{G}}_l$ up to the last $\sim 1/(1-\mu)$ steps, with $\bar{\mathbf{G}}_l$ the sum of the clipped minibatch gradients (Table 7). The Table 1 ordering is unchanged, and the accumulated-update continuity matches the cumulative-gradient continuity to within 0.03. For Res+None the SGD cumulative gradient is itself less coherent (0.468; 0.851 under Adam, though SGD sums clipped gradients along a different trajectory), and since $\mathbf{W}_T = \mathbf{W}_0 + \Delta\mathbf{W}$ exactly, the low weight continuity (0.075 vs. 0.440 for $\Delta\mathbf{W}$) shows that $\Delta\mathbf{W}$ does not come to dominate $\mathbf{W}_0$ in the leading direction; these runs do not test whether SGD drifts along the flat orbit. NoRes+ReLU at depth 16 fails to train under SGD at this learning rate (a known plain-network trainability issue; we did not tune it further), so that configuration is Adam-only.

Table 7: SGD+momentum replication (MNIST, 16-layer MLP, lr 0.05, momentum 0.9, gradient-norm clipping at 1, 2 seeds, mean). $(\mathbf{W}_T - \mathbf{W}_0)\,\mathbf{v}_1$ = continuity of the accumulated update, which under SGD with momentum μ is $-\frac{\eta}{1-\mu}\bar{\mathbf{G}}$ up to the last few steps. †numerically linear at the operating scale (Section 4.3).

Configuration	Acc	$\bar{\mathbf{G}}\,\mathbf{v}_1$	$(\mathbf{W}_T - \mathbf{W}_0)\,\mathbf{v}_1$	$\mathrm{W}\,\mathbf{v}_1$	Δ_{GW}	$\mathrm{W}\,\mathbf{u}_1$
Res+ReLU	.977	.964	.964	**.933**	+.031	**.968**
Res+None	.870	.468	.440	.075	+.393	.072
Res+Radial†	.879	.395	.400	.070	+.325	.068
NoRes+ReLU	.113	*fails to train under SGD (Adam-only in Table 1)*				

Adam β_2=0. Adam β_2 controls the second-moment estimate that provides per-parameter adaptive learning rates. Setting β_2=0 with ε=1 disables the adaptive scaling that normally amplifies updates: the update becomes $\text{lr} \cdot m_t/(|g_t| + 1) \approx \text{lr} \cdot m_t$, effectively SGD with momentum (β_1=0.9). This produces small weight updates relative to initialization ($\|\mathbf{W}_{\text{final}}\| \approx \|\mathbf{W}_{\text{init}}\|$), which we exploit in Section 4.4 to verify the $\|\Delta\mathbf{W}\|/\|\mathbf{W}^{(0)}\|$ ratio prediction: reducing initialization scale with β_2=0 progressively recovers continuity (Table 8).

Table 8: Optimizer and initialization-scale ablations (Res+ReLU, MNIST, 16-layer MLP, 3 seeds, mean). σ^2-WA = σ^2-weighted alignment. **Bold** = notably high; underline = notably low.

Configuration	Acc	G $\mathbf{v}_1$	G σ^2-WA	W $\mathbf{v}_1$	W $\mathbf{v}_2$	W σ^2-WA	W $\mathbf{u}_1$
Optimizer ablation:							
Adam β_1=0, β_2=.999	.971	.868	.832	.920	.421	.251	.857
Adam β_1=.9, β_2=0	.968	**.945**	**.931**	.047	.042	.050	.052
Init scale ablation (Adam β_2=0):							
init std = 0.005	.923	.993	.981	.529	.330	.084	.337
init std = 0.001	.907	.997	.996	**.929**	**.804**	**.520**	.853
init std = 0.0001	.885	.999	.998	**.958**	**.921**	**.949**	**.948**

Depth sweep. To check that depth 16 is not special, we repeat Res+ReLU and Res+None at $L = 4, 8, 32$ (2 seeds; Table 9). ReLU continuity rises with depth and saturates ($0.82 \to 0.93 \to 0.96 \to 0.95$) while accuracy is flat, and the linear control stays low at every depth. L=16 is where the phenomenon is clearest, not its cause: it is already present at L=4.

Table 9: Depth sweep (MNIST, D=256, Adam, 20 epochs; 2 seeds, mean; the L=16 row is Table 1, 3 seeds).

	Res+ReLU				Res+None			
Depth L	Acc	W $\mathbf{v}_1$	W $\mathbf{u}_1$	W σ^2-WA	Acc	W $\mathbf{v}_1$	W $\mathbf{u}_1$	W σ^2-WA
4	.978	.822	.947	.235	.921	.183	.159	.059
8	.975	.929	.951	.252	.921	.145	.167	.059
16	.976	**.959**	.937	.249	.909	.257	.260	.061
32	.976	.952	.838	.203	.878	.119	.128	.051

G Full Continuity Metrics

Table 10 and Table 11 report the complete set of continuity metrics for the main configurations of an earlier run (MNIST, 16-layer MLP, 3 seeds, mean $\pm$ std). Where LayerNorm is used it is applied to the stream after the residual addition, $\mathbf{h}_{l+1} = \text{LN}(\mathbf{h}_l + \text{act}(\mathbf{W}_l\mathbf{h}_l))$; the eq-err columns of Tables 1 and 12 and the *nonlin* column of Table 12 are those of the activation alone. For ReLU the definition is exact in closed form, $\text{eq-err} = \mathbb{E}\|\mathbf{z}\|^2/(2\,\mathbb{E}\|\sigma(\mathbf{z})\|^2)$ (the rotation-averaged numerator is exactly $\|\mathbf{z}\|^2/2$), so values $\gg 1$ (ReLU, GELU, SiLU) mean that little of the pre-activation energy lies where σ is active; sign-symmetric pre-activations would give 1.

Three regimes of Δ_{GW}. The cumulative-gradient analysis (Table 1, main text) shows three principal regimes: (i) symmetry-breaking activations (ReLU, GELU, SiLU) achieve $\Delta_{\text{GW}} \leq 0.012$ (near-full transfer); (ii) linear configurations—Res+None, Res+Radial, Res+Diag and Res+Perm, whether or not they break symmetry—show $\Delta_{\text{GW}} = 0.59$–$0.70$ despite having coherent cumulative gradients (0.84–0.86), demonstrating drift-driven loss; (iii) NoRes configurations have $\Delta_{\text{GW}} \leq 0.01$, or negative where the weight is more continuous than its incoherent cumulative gradient (NoRes+ReLU -0.255), because the cumulative gradient itself never becomes coherent ($\bar{\mathbf{G}}\mathbf{v}_1 \leq 0.16$). Res+None and Res+Radial agree because the latter is numerically linear at the operating scale (Section 4.3); the genuinely nonlinear equivariant control Res+EqNL transfers as fully as Res+Leaky(.1) ($\Delta_{\text{GW}} = 0.045$ for both) but reaches only $\mathbf{v}_1 = 0.435$, because its cumulative gradient is itself less coherent ($\bar{\mathbf{G}}\mathbf{v}_1 = 0.480$, versus 0.956): for this strong nonlinearity the loss arises not in the transfer but upstream, in the gradient.

Table 10: Full gradient continuity metrics. Residual connections ensure high gradient coherence across all right singular vector directions (σ^2-WA($\mathbf{v}$) > 0.75). Without residual, gradient coherence vanishes. Notably, among the architecture ablations Res+None has the highest gradient σ^2-WA in both $\mathbf{v}$- and $\mathbf{u}$-space (0.93, 0.94), yet transfers almost nothing to weights (Table 11).

Config	Acc	G $\mathbf{v}_1$	G $\mathbf{v}_2$	G σ^2-WA($\mathbf{v}$)	G $\mathbf{u}_1$	G $\mathbf{u}_2$	G σ^2-WA($\mathbf{u}$)
Res+GELU	.978	.901±.059	.808±.096	.887±.066	.041±.030	.010±.004	.039±.027
Res+SiLU	.977	.831±.024	.684±.086	.809±.030	.051±.024	.035±.013	.051±.022
Res+ReLU	.977	.792±.040	.687±.027	.782±.043	.035±.008	.037±.010	.035±.007
Res+Tanh	.971	.846±.080	.678±.180	.756±.134	.068±.007	.072±.016	.066±.007
Res+None	.912	.927±.021	.932±.011	**.928**±.018	.941±.020	.943±.007	**.941**±.016
Res+None+LN	.918	.885±.013	.857±.032	.872±.015	.883±.010	.867±.027	.875±.010
Res+ReLU+LN	.980	.876±.008	.734±.118	.863±.016	.204±.035	.203±.005	.202±.035
NoRes+GELU	.924	.046±.010	.052±.015	.047±.011	.041±.008	.048±.013	.043±.007
NoRes+ReLU	.848	.067±.012	.059±.011	.068±.013	.034±.006	.045±.010	.037±.003
NoRes+None	.873	.057±.005	.045±.011	.055±.004	.037±.003	.044±.008	.039±.002
Adam β_1=0	.971	.868±.088	.697±.153	.832±.106	.036±.016	.049±.015	.035±.012
Adam β_2=0	.968	.945±.013	.910±.026	.931±.022	.444±.068	.495±.032	.455±.044
β_2=0 init .005	.923	.993±.003	.992±.002	.981±.005	.708±.049	.766±.010	.704±.033
β_2=0 init .001	.907	.997±.001	.997±.001	.996±.002	.916±.014	.923±.011	.910±.011
β_2=0 init .0001	.885	.999±.000	.998±.001	.998±.001	.986±.002	.985±.003	.984±.003

Table 11: Full weight continuity metrics. Activation (ReLU/GELU/SiLU) concentrates structure in $\mathbf{v}_1$ (> 0.95) with much lower $\mathbf{v}_2$ ($\sim$0.2). LayerNorm alone (Res+None+LN) distributes structure most widely, yielding the highest weight σ^2-WA (0.303) among the architecture ablations. Tanh shows high $\mathbf{v}_2$ (0.42) but kills $\mathbf{u}_1$ (0.06). Reducing initialization scale with β_2=0 recovers continuity in *all* directions, consistent with the $\|\Delta\mathbf{W}\|/\|\mathbf{W}_{\text{init}}\|$ ratio controlling continuity.

Config	Acc	W $\mathbf{v}_1$	W $\mathbf{v}_2$	W σ^2-WA($\mathbf{v}$)	W $\mathbf{u}_1$	W $\mathbf{u}_2$	W σ^2-WA($\mathbf{u}$)
Res+GELU	.978	**.964**±.001	.219±.011	.249±.001	**.955**±.001	.056±.012	.202±.002
Res+SiLU	.977	**.964**±.001	.177±.034	.242±.003	**.962**±.001	.064±.012	.200±.001
Res+ReLU	.977	**.959**±.001	.198±.027	.249±.002	**.940**±.001	.050±.006	.206±.001
Res+Tanh	.971	.842±.017	**.418**±.042	**.285**±.010	.061±.011	.043±.006	.051±.001
Res+None	.912	.217±.042	.155±.028	.059±.003	.218±.039	.147±.035	.058±.002
Res+None+LN	.918	.631±.021	**.506**±.032	**.303**±.008	.622±.032	**.499**±.031	**.298**±.005
Res+ReLU+LN	.980	.889±.008	**.321**±.051	.252±.006	**.914**±.002	**.122**±.028	.201±.003
NoRes+GELU	.924	.175±.016	.262±.021	.093±.005	.549±.054	.109±.038	.148±.008
NoRes+ReLU	.848	.378±.014	.059±.006	.089±.003	**.701**±.030	.041±.006	.131±.003
NoRes+None	.873	.048±.011	.052±.012	.050±.002	.059±.012	.046±.002	.051±.001
Adam β_1=0	.971	**.920**±.009	**.421**±.028	.251±.002	**.857**±.011	**.148**±.011	.191±.003
Adam β_2=0	.968	.047±.004	.042±.003	.050±.001	.052±.009	.044±.001	.050±.000
β_2=0 init .005	.923	.529±.030	.330±.088	.084±.006	.337±.022	.213±.054	.069±.003
β_2=0 init .001	.907	**.929**±.033	**.804**±.025	**.520**±.012	**.853**±.037	**.716**±.036	**.469**±.010
β_2=0 init .0001	.885	**.958**±.021	**.921**±.026	**.949**±.012	**.948**±.021	**.914**±.025	**.938**±.013

Tanh: a tentative account of why $\mathbf{v}_1$ succeeds but $\mathbf{u}_1$ fails. Tanh achieves high $\mathbf{v}_1$ continuity (0.846) but near-random $\mathbf{u}_1$ continuity (0.058; Table 1). We offer the following as a *plausible* mechanism rather than a verified cause. Activation affects the forward pass as follows: $\mathbf{W}_l$ maps its dominant input direction $\mathbf{v}_1$ to the output direction $\sigma_1 \cdot \mathbf{u}_1$. With ReLU, this output is partially preserved (only negatives are zeroed), so the $\mathbf{u}_1$ signal propagates to the next layer's input, creating $\mathbf{u}_1$ coherence across layers (W $\mathbf{u}_1 = 0.937$). With Tanh, 92–94% output saturation ($|\tanh(\cdot)| > 0.95$) causes the derivative $\tanh'(x) = 1 - \tanh^2(x) \approx 0$, which would suppress gradient information in the output direction during backpropagation. Consistent with this account, we observe that ReLU's cumulative gradient develops $\mathbf{u}_1$ coherence over training ($0 \to 0.71$) while Tanh's fails to develop ($0.3 \to 0.14$). The input direction $\mathbf{v}_1$ is determined *before* activation is applied and is therefore unaffected by saturation; its moderately lower continuity (0.846 vs. 0.959 for ReLU) splits, by the definition of Δ_{GW}, into a slightly less coherent cumulative gradient (0.915 vs. 0.961) and a slightly less complete transfer ($\Delta_{\text{GW}} = 0.069$ vs. 0.003; Table 12), in line with its far smaller equivariance error (0.635 vs. 193). Varying the saturation regime supports the $\mathbf{u}_1$ account: with input gain $g < 1$ ($\tanh(gx)$; Table 13), $\mathbf{u}_1$ stays near random down to g=0.25 (0.04–0.06), begins to recover at g=0.1 (0.23), and reaches ReLU's level at g=0.02 (0.93 vs. 0.94).

Table 12: All 21 configurations of the symmetry-class ablation (MNIST, 16-layer residual MLP, 3 seeds, mean; 20 epochs, Adam 10^{-3}). Columns as in Table 1, plus *nonlin* = relative residual of the activation's best linear fit on the trained network's own pre-activations (0 = linear). Table 1 and this table are an independent re-run of the configurations in Tables 10–11 with the same implementation; differences (e.g. Res+None+LN 0.631 vs. 0.665) are mostly within seed variance; the largest relative to seed spread, NoRes+ReLU $\mathbf{u}_1$ (0.701 vs. 0.647), exceeds it. $^{\dagger}\mathbf{x}\tanh(\|\mathbf{x}\|)$, i.e. $h(r) = r\tanh(r) \approx r$ at the observed $\|\mathbf{x}\| \approx 5$–41: numerically the identity (relative deviation from its best linear fit 4×10^{-5}; ReLU 0.50); $^{\ddagger}\sigma(\mathbf{x}) = \hat{\mathbf{x}}\|\mathbf{x}\|^{\alpha}$, exactly $SO(d)$-equivariant; the separate sweep of $\alpha \in \{1, .9, .75, .5, .25, .1, 0\}$ (Section 4.3; 3 seeds) used an earlier implementation with a bias in the embedding layer, so its values (W $\mathbf{v}_1$ 0.14–0.45; 0.449 at α=0) differ slightly from this table's. Leaky(a): LeakyReLU with negative slope a; Abs: $\sigma(\mathbf{x}) = |\mathbf{x}|$ element-wise.

Configuration	eq-err	nonlin	Acc	$\bar{\mathbf{G}}\,\mathbf{v}_1$	W $\mathbf{v}_1$	W σ^2-WA	Δ_{GW}	s_l	W $\mathbf{u}_1$
*Equivariant (*eq-err ≈ 0*)*									
Res+None	0	.000	.909	.851	.257	.061	$+.594$	.261	.260
Res+Radial†	0	.000	.913	.851	.192	.059	$+.660$	.180	.191
Res+EqNL(α=.5)‡	0	.120	.970	.484	.312	.220	$+.172$	.127	.192
Res+EqNL(α=0)‡	0	.332	.957	.480	.435	.311	$+.045$	.354	.548
Non-equivariant but linear									
Res+Diag	.188	.000	.913	.857	.163	.057	$+.694$	.164	.179
Res+Perm	1.99	.000	.914	.840	.138	.057	$+.702$	.049	.048
Non-equivariant nonlinear									
Res+Leaky(.9)	.006	.010	.932	.845	.265	.069	$+.580$	.075	.272
Res+Leaky(.7)	.066	.039	.965	.879	.194	.067	$+.686$	.123	.193
Res+Leaky(.5)	.271	.077	.971	.914	.532	.089	$+.381$	.350	.549
Res+Tanh	.635	.459	.970	.915	.846	.283	$+.069$	.151	.058
Res+Leaky(.3)	1.00	.146	.973	.937	.810	.107	$+.126$	.474	.850
Res+Abs	2.02	.199	.883	.658	.498	.134	$+.160$	.058	.059
Res+Leaky(.1)	8.76	.296	.970	.956	.911	.163	$+.045$	.389	.929
Res+ReLU	193	.497	.976	.961	.959	.249	$+.003$	.564	.937
Res+GELU	206	.515	.975	.977	.964	.249	$+.012$	.535	.955
Res+SiLU	247	.523	.976	.977	.965	.243	$+.012$	.570	.963
Normalization									
Res+None+LN	0	.000	.924	.733	.665	.315	$+.068$	.741	.650
Res+ReLU+LN	15	.178	.979	.899	.892	.248	$+.007$	.301	.912
No residual									
NoRes+None	0	.000	.706	.050	.041	.049	$+.009$	.312	.055
NoRes+ReLU	4.36	.030	.727	.106	.360	.088	$-.255$	.590	.647
NoRes+GELU	11	.044	.939	.162	.225	.083	$-.063$	.312	.637

Two factors: equivariance error and update dominance. Among the non-equivariant activations, Res+Tanh with input gain 0.02 is the one configuration whose continuity exceeds what its equivariance error alone would predict (eq-err $= 0.12$, $\mathbf{v}_1 = 0.95$; Table 13). It is consistent with the second factor of Section 4.4, the dominance of the accumulated update over the initialization: its update is no larger overall (Frobenius ratio 2.1, versus 2.3–2.6 for the other gains) but is concentrated in its leading direction (σ_1^2 share 0.23 versus 0.09–0.11, spectral ratio 8.4 versus 5.7–6.9), as in ReLU (0.24). Update dominance amplifies the retention that the equivariance error provides but cannot substitute for it: with a linear activation, the small-initialization regime forms continuity and then loses it (Figure 2).

Accuracy is not the confounder. Because symmetry-breaking activations also train better, continuity could in principle be a by-product of accuracy. Table 14 rules this out in both directions, on MNIST and on a harder task (CIFAR-10 [31] fed to the same 16-layer residual MLP as flattened 3072-dimensional inputs, $\sim$50% accuracy regime). At matched accuracy, Res+Tanh and Res+Leaky(.7) differ $3.5\times$ (MNIST) and $5.7\times$ (CIFAR-10) in weight $\mathbf{v}_1$; conversely, Res+Tanh with input gain 0.02 is among the least accurate configurations yet among the most continuous, and NoRes+ReLU trains poorly yet keeps output-side ($\mathbf{u}_1$) continuity. The 3-seed runs of Table 12 show the same matched pair (accuracy .970 vs. .965; $\mathbf{v}_1$ .846 vs. .194).

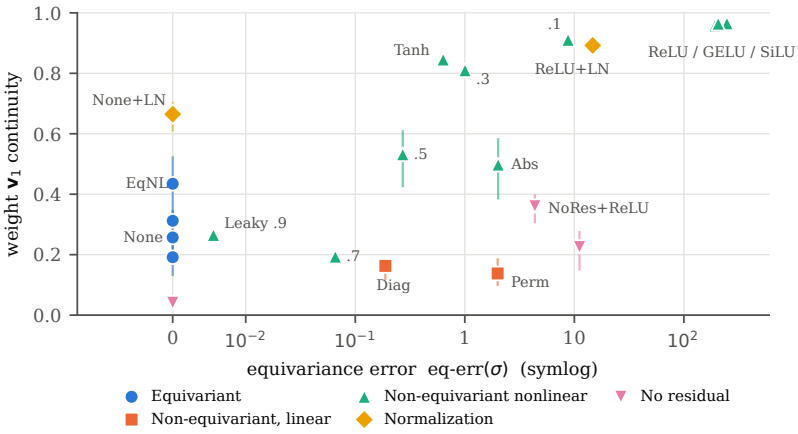


Figure 13: **Equivariance error versus weight $\mathbf{v}_1$ continuity** for the 21 configurations of Table 12 (point = 3-seed mean, bar = seed range; symlog axis so that eq-err $= 0$ is shown). Spearman $+0.89$ over the 12 nonlinear residual configurations without normalization (EqNL, LeakyReLU, Tanh, Abs, ReLU, SiLU, GELU), $+0.80$ with the four single-seed Tanh input-gain variants of Table 13 added to these 12, and $+0.62$ over all 21, which also include the four linear configurations (Res+None, Res+Radial, Res+Diag, Res+Perm), the three without the residual stream (NoRes), and the two with LayerNorm (Res+None+LN, Res+ReLU+LN), whose eq-err does not count the normalization. Within a single intertwiner group, the LeakyReLU slopes span $\mathbf{v}_1 = 0.19$–0.91, rising with eq-err above eq-err ≈ 0.1; the linear symmetry breakers (Diag, Perm) stay at or below the linear Res+None level despite eq-err > 0.

Table 13: Tanh input gain, $\tanh(gx)$, with both factors tabulated (MNIST, 16-layer residual MLP, seed 42, as in Table 14). Update ratios are layer means of $\|\Delta\mathbf{W}_l\|/\|\mathbf{W}_l^{(0)}\|$ in the spectral (2) and Frobenius (F) norms; σ_1^2 share = fraction of $\|\Delta\mathbf{W}_l\|_F^2$ in its leading singular direction. The large eq-err of ReLU-family activations varies strongly between runs, even at a fixed seed (Res+ReLU: 264 here, 214 for seed 42 in Table 12, 131–232 over its three seeds), while its continuity does not (W $\mathbf{v}_1$ .958 in both seed-42 runs).

Configuration	Acc	eq-err	ratio$_2$	ratio$_F$	σ_1^2 share	W $\mathbf{v}_1$	W $\mathbf{u}_1$
Res+Tanh (g=1)	.963	.635	5.98	2.26	.107	.862	.053
Res+Tanh (g=.5)	.970	.595	5.68	2.32	.091	.651	.061
Res+Tanh (g=.25)	.974	.523	6.00	2.40	.096	.604	.038
Res+Tanh (g=.1)	.971	.373	6.95	2.55	.114	.642	.228
Res+Tanh (g=.02)	.930	.121	**8.38**	2.12	**.231**	**.952**	**.929**
Res+ReLU	.976	264	6.43	1.65	.240	.958	.938
Res+Leaky(.7)	.964	.068	1.58	1.02	.038	.249	.295

Adjacent-layer weight sharing. As a first probe of whether continuity matters for cross-layer sharing, we take the MNIST models of Table 14, replace each disjoint adjacent pair of block weights, $(\mathbf{W}_0, \mathbf{W}_1), (\mathbf{W}_2, \mathbf{W}_3), \ldots, (\mathbf{W}_{14}, \mathbf{W}_{15})$, by its mean in both slots, and evaluate without retraining. Accuracy drops by 0.19 for Res+Tanh (.963 $\rightarrow$ .771), 0.30 for Res+Leaky(.7) (.964 $\rightarrow$ .660), and 0.06 for Res+ReLU (.976 $\rightarrow$ .919).

The MNIST bench is an instrument, not a tuned classifier. The 16-layer bench deliberately omits learning-rate decay, augmentation and biases, since each could interact with the interventions under study, so its accuracy (97.6% in Table 1; $97.55\% \pm 0.05$ in the 3-seed re-run used here) is below what MNIST permits. Cosine learning-rate decay raises Res+ReLU to $98.68\% \pm 0.10$ (3 seeds), and adding augmentation (± 2 px shifts, $\pm 10^\circ$ rotations, 40 epochs) to $99.13\% \pm 0.01$ (2 seeds). The conclusions are unchanged there: weight $\mathbf{v}_1$ continuity is 0.950 ± 0.000 for Res+ReLU, 0.190 ± 0.002 for Res+None and 0.194 ± 0.010 for Res+Radial†. Depth does not buy accuracy on this task: a plain 2-layer MLP (width 1024, ± 3 px, $\pm 12^\circ$, 60 epochs, cosine decay) reaches $99.20\% \pm$

Table 14: Accuracy–continuity double dissociation (16-layer residual MLP, Adam 10^{-3}, 20 epochs, seed 42; an independent re-run of the Table 12 configurations, whose 3-seed means differ from it, e.g. Res+Leaky(.7) $\mathbf{v}_1$ 0.194). Matched-accuracy pairs in **bold**.

	MNIST			CIFAR-10 (flattened)		
Configuration	Acc	W $\mathbf{v}_1$	W $\mathbf{u}_1$	Acc	W $\mathbf{v}_1$	W $\mathbf{u}_1$
Res+ReLU	.976	.958	.938	.510	.918	.951
Res+Tanh	**.963**	**.862**	.053	**.483**	**.803**	.048
Res+Leaky(.7)	**.964**	**.249**	.295	**.480**	**.142**	.174
Res+Tanh, gain 0.02	.930	.952	.929	.432	.756	.743
Res+None	.915	.167	.171	.393	.057	.065
NoRes+ReLU	.818	.397	.703	–	–	–

0.03, and with its activation removed the 16-layer net collapses to a single linear map that scores 91.44%—below multinomial logistic regression on raw pixels (92.55%)—and errs on the same test points (error-set Jaccard 0.687, versus 0.041 expected for independent errors at these accuracies). Depth in this bench is a means of observing cross-layer geometry, not a route to accuracy. Across these recipes the *ordering* of configurations, and the agreement of Res+Radial with Res+None, is stable, but the absolute scale of continuity is not (Res+ReLU 0.93–0.97; Res+None from 0.04 in a seed-42 run with cosine decay alone to 0.64 for a wider net, width 512, trained 40 epochs with decay). We therefore use the continuity metrics as comparative quantities between conditions rather than as constants of the phenomenon.

H Gradient Rank and Task Structure

To test whether gradient rank (rather than the architectural conditions of Section 4) drives weight continuity, we measure weight $\mathbf{v}_1$ continuity and gradient effective rank across six classification datasets (MNIST [14], EMNIST-Letters [32], Fashion-MNIST [33], SVHN [34], CIFAR-10 and CIFAR-100 [31]; loaded through torchvision) and three architectures (Res+ReLU, Res+None, NoRes+ReLU) with 3 seeds each (Figure 14). The effective rank [35] of a matrix with singular values $\sigma_1, \ldots, \sigma_n$ is

$$\text{erank}(\mathbf{W}) = \exp\left(-\sum_i p_i \log p_i\right), \quad p_i = \frac{\sigma_i^2}{\sum_j \sigma_j^2}, \tag{6}$$

the effective dimensionality of the matrix: a rank-1 matrix has erank $= 1$, and a matrix with all equal singular values has erank $= n$.

Two patterns emerge: **(i)** within Res+ReLU, continuity trends weakly downward with gradient rank (Figure 14a, erank 1.6–15.4), consistent with more concentrated gradients producing stronger continuity; **(ii)** across architectures at matched gradient ranks, architecture dominates—Res+ReLU achieves $\mathbf{v}_1 > 0.92$ *regardless* of gradient rank, while Res+None remains below 0.27 and NoRes+ReLU in the 0.38–0.48 band (Figure 14b). Gradient rank therefore modulates continuity within an architecture but does not substitute for the architectural conditions.

I Intervention Experiments

Removing continuity: rotation injection. Is continuity functionally inert—a by-product that could be erased without consequence? We remove it directly from trained networks (seed-42 reruns of Table 1) by applying the conjugation of Section 4.3, $\mathbf{T}_l \mapsto \mathbf{R}_l \mathbf{T}_l \mathbf{R}_{l-1}^\top$ with $\mathbf{T}_l = \mathbf{I} + \mathbf{W}_l$, to every interior interface, using independent partial rotations $\mathbf{R}_l = \exp(\theta \mathbf{S}_l)$ ($\mathbf{S}_l$ random skew-symmetric, $\|\mathbf{S}_l\|_2 = 1$, so $\mathbf{R}_l$ is orthogonal with maximal rotation angle θ; it preserves the norm of $\mathbf{T}_l$, but in Res+ReLU, where no compensating reparameterization exists, it also adds $\mathbf{R}_l \mathbf{R}_{l-1}^\top - \mathbf{I}$ to $\mathbf{W}_l$, so the spectrum-preserving rotations of $\mathbf{W}_l$ itself below are the cleaner test). In the linear network (Res+None) we additionally absorb the boundary rotations into the embedding and head, which makes the substitution exactly function-preserving; no such compensation exists once a non-equivariant nonlinearity sits between the factors (Res+ReLU). Table 15 shows the two cases separating. In the linear network continuity is erased ($\mathbf{v}_1$ 0.17 $\to$ 0.05) while accuracy, loss and gradient

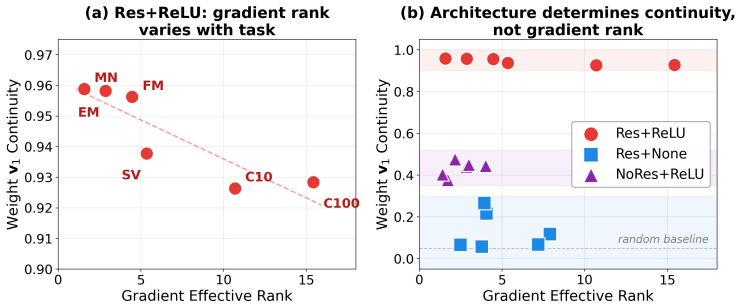


Figure 14: **Gradient rank and weight continuity across datasets and architectures.** (a) Within Res+ReLU, weight $\mathbf{v}_1$ continuity trends downward with gradient effective rank across six classification datasets (erank 1.6–15.4). (b) Across three architectures ($\times$ six datasets $=$ 18 points), continuity is determined by *architecture* (color bands), not gradient rank (x-axis): Res+ReLU achieves $\mathbf{v}_1 > 0.92$ regardless of gradient rank, while Res+None remains below 0.27. Architecture—not gradient rank—is the primary factor controlling weight continuity.

norm are unchanged: there, continuity is indeed a gauge choice of a function-preserving reparameterization, and only the gradients' frames rotate with the weights (gradient $\mathbf{v}_1$ $0.94 \to 0.31$). In Res+ReLU, continuity and function degrade together, and the rotation triggers a large *coherent* corrective signal: the gradient norm grows by nearly four orders of magnitude while cross-layer gradient coherence stays at $\geq$0.82, i.e. the residual pathway that formed the alignment immediately works to restore it. Among the directions we tested we found no norm-preserving rotation that removes continuity while sparing the function, as the flat-manifold analysis predicts: under a non-equivariant nonlinearity every rotational direction carries loss curvature.

Table 15: Rotation injection into trained networks (MNIST, 16-layer residual MLP, seed 42). G $\mathbf{v}_1$ = cross-layer continuity of the gradient; $\|\nabla\|$ = gradient norm. Res+None is compensated at the boundaries (function exactly preserved); Res+ReLU cannot be.

	Res+ReLU				Res+None (compensated)			
θ	W $\mathbf{v}_1$	Acc	$\|\nabla\|$	G $\mathbf{v}_1$	W $\mathbf{v}_1$	Acc	$\|\nabla\|$	G $\mathbf{v}_1$
0	.958	.976	0.42	.84	.167	.915	1.53	.94
0.25	.946	.973	0.62	.89	.146	.915	1.53	.93
0.5	.910	.963	1.06	.94	.098	.915	1.53	.88
1.0	.776	.502	32.4	.92	.053	.915	1.53	.73
2.0	.387	.103	3,153	.82	.049	.915	1.53	.31

Full random rotations: what an invariant summary cannot see. Extending the θ-sweep to independent Haar-random rotations applied to the weights themselves, $\mathbf{W}_l \mapsto \mathbf{R}_{l+1}\mathbf{W}_l\mathbf{R}_l^\top$, leaves every $\mathbf{W}_l$'s singular-value spectrum exactly unchanged (relative deviation below 8×10^{-5}, float32) yet destroys Res+ReLU: $\mathbf{v}_1$ continuity falls from 0.958 to 0.05 and accuracy from 0.976 to 0.10, chance level (mean of 5 draws; the same with identity or random boundary rotations). A summary that is invariant to these rotations, such as the spectra, cannot distinguish a trained network from a functionally destroyed one. Some frame sensitivity is therefore necessary for the questions studied here, which is why the frame-sensitive $\mathbf{v}_1$/$\mathbf{u}_1$ continuities are our primary metrics and the rotation-invariant s_l is reported alongside them (Table 1).

Preventing the collapse: flat-direction projection. We test directly whether rotation drift along the flat manifold *causes* the continuity collapse of Section 4.3 by removing that motion from the optimizer. In the small-initialization Res+None setting of Figure 2 (σ=0.0001, D=256, L=16, MNIST, Adam 10^{-3}, 50 epochs, single seed), at every step, before the Adam update $\Delta\mathbf{W}$ is applied, we project it onto the flat tangent space $\{\delta\mathbf{W}_l = \mathbf{A}_l\mathbf{T}_l - \mathbf{T}_l\mathbf{A}_{l-1}\}$ (orthogonal projection in the Frobenius metric, solved by warm-started conjugate gradients) and subtract the projected component. We use two projectors: the *interior* one above ($\mathbf{A}_0 = \mathbf{A}_L = \mathbf{0}$) and a *full* one that also removes the boundary directions $\mathbf{A}_0$ and $\mathbf{A}_L$, compensated in the embedding and head. Figure 15 compares

interior projection at every step, full projection at 70% strength, and full projection released at epoch 25 with the untouched baseline and a Res+ReLU control.

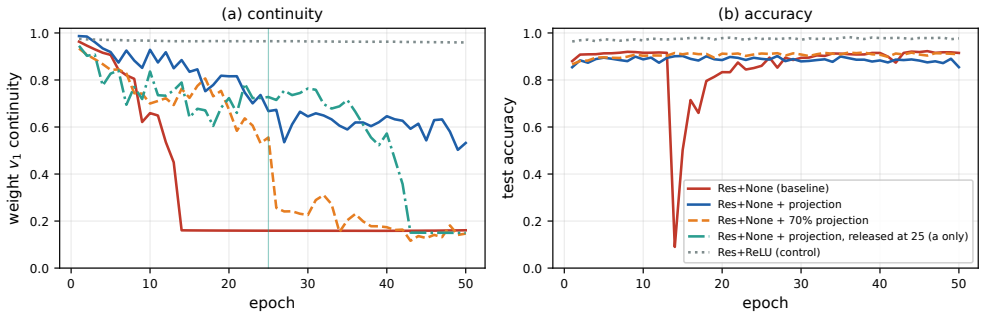


Figure 15: **Flat-direction projection prevents the drift-driven collapse.** Res+None, σ=0.0001, 50 epochs, single seed. **(a)** Weight $\mathbf{v}_1$ continuity. The baseline collapses from 0.96 to 0.16 by epoch 14; with the flat component (38–86% of the update norm) removed at every step by the interior projector, continuity stays at 0.83–0.93 through that window and then decays slowly to 0.53 at epoch 50; with the full projector, removing only 70% of the component delays the collapse to epoch 26, and releasing it at epoch 25 (vertical line) lets continuity decline and collapse by epoch 43. **(b)** Test accuracy. The baseline's transient crash ($0.92 \rightarrow 0.09$ at epoch 14, recovering) disappears under projection (0.85–0.90 throughout).

Findings. (i) The flat component is large—38% of the update norm at epoch 1, peaking at 86% by epoch 21—yet removing it leaves training functional (accuracy 0.85–0.90), consistent with these directions carrying no gradient signal. (ii) Projection holds continuity at 0.83–0.93 through the window in which the baseline collapses to 0.16, and removes the accompanying accuracy crash; the drift is therefore causal for both. (iii) The slow remaining decay after epoch 16 (to 0.53 at epoch 50) is not removed by also projecting out the boundary directions: the full projector at full strength behaves similarly but holds continuity somewhat lower (0.64–0.84 in epochs 5–15, 0.51 at epoch 50; accuracy 0.78–0.89). Its source, possibly the residual of the inexact conjugate-gradient solve, is not identified. (iv) With the full projector, partial projection (70%) merely delays the collapse (to epoch 26 instead of 14), and once the projection is released at epoch 25 the drift resumes and continuity collapses by epoch 43. The transition is thus driven by *cumulative* motion along the orbit, with no single threshold event. The connection to feature-learning and edge-of-stability transitions is left to future work. A single seed is used here. The runs use a separate implementation of the Figure 2 setting with the same seed; its baseline crashes deeper ($0.92 \rightarrow 0.09$ vs. 0.61 in Figure 2) at nearly the same time (epoch 14 vs. 15).

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 abstract and introduction claim (1) two necessary conditions for geometric continuity—residual connections (which create cross-layer gradient coherence) and symmetry-breaking nonlinearities (which prevent rotation drift)—and (2) in transformers, continuity is projection-specific, with Q/K/Gate/Up developing input-space ($\mathbf{v}_1$) continuity, O/Down developing output-space ($\mathbf{u}_1$) continuity, and V alone (lacking an adjacent nonlinearity) developing only low continuity. These claims are directly supported by the controlled ablation experiments in Section 4 (toy MLP) and Section 5 (small transformer + pretrained Llama-3.1-8B validation).

2. **Limitations**

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

Answer: [Yes]

Justification: Section 6 (Discussion) includes a dedicated *Limitations* paragraph covering: (1) the indirect nature of the Q/K mechanism claim (softmax cannot be ablated without changing the attention mechanism) and that V is tested only by adding a nonlinearity (V+SiLU, 3 seeds); (2) small transformer scale (34M parameters); (3) training dynamics (rotation drift) not verified at larger scales; (4) single-seed drift, rotation-injection and projection runs (Figure 2, Appendix I) and transformer ablations (Table 2); and (5) the configuration-dependent absolute scale of the continuity metrics.

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: [N/A]

   Justification: The paper is primarily empirical. The rotational symmetry breaking argument (Section 4.3) builds on classical results for deep linear networks [8, 11] and conservation-law analysis for residual architectures [13]; Section 4.3 gives a flat-manifold argument for frame pinning (an exactly function-preserving reparameterization and a second-order stability argument) without stating formal theorems; a rigorous treatment of the drift dynamics is left to future work. No new theorems or proofs are claimed.

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: Section 4.1 specifies the toy MLP architecture, optimizer, dataset, and training duration. Appendix A.1 specifies the transformer setup (D=256, 8 layers, 4 heads, WikiText-103, 10,000 steps, batch 128). All ablation conditions are fully described. Llama-3.1-8B is publicly available.

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: [Yes]

   Justification: Core code is available at `https://github.com/smallhours19/geometric-continuity` (MIT license): the singular-vector, alignment and equivariance-error metrics, toy-MLP training with the seeds of Table 1, weight analysis of Llama-style pretrained LLMs (Llama-3.1-8B row of Table 2), and trajectory plotting, with a README giving the commands. The remaining experiments (small-transformer training and the appendix studies) follow the setups specified in Section 4.1, Appendix A.1 and the respective appendices. All datasets used (MNIST, EMNIST-Letters, Fashion-MNIST, SVHN, CIFAR-10, CIFAR-100, WikiText-103) are publicly available, and every pretrained checkpoint analyzed (Llama-3.1-8B and the models of Appendices B–D) is a public Hugging Face release.

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: Section 4.1 and Appendix A.1 provide all hyperparameters. Standard train/test splits are used. No architecture-specific tuning was performed; optimizer settings and initialization scales were varied in Appendix F to probe the mechanism.

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: Table 1 (toy MLP) reports means over 3 seeds, with seed ranges of weight $\mathbf{v}_1$ in Figure 13; Tables 10–11 give standard deviations for an earlier 3-seed run of the main configurations. Table 2 (transformer) is single-seed (Appendix A.1); the V+SiLU intervention uses 3 seeds. Figure 2 and the rotation and projection runs of Appendix I are single-seed. $\beta_2{=}0$ experiments (Appendix F) report means over 3 seeds; other appendix experiments state their seed counts. Single-seed transformer is justified by consistency with pretrained Llama-3.1-8B patterns and the large effect sizes observed (e.g., Gate $\mathbf{v}_1$: 0.66 vs. 0.37).

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: Most experiments ran on NVIDIA A100 GPUs; those added in this final version (including the re-run of Table 1) ran on NVIDIA H200 GPUs. Toy MLP experiments run on a single GPU in under 10 minutes each. Each transformer config (10,000 steps) takes $\sim$20 minutes on a single A100. The Llama-3.1-8B analysis reads $\sim$16 GB of bf16 weights one matrix at a time; the released script runs it on a CPU. All analyses of pretrained checkpoints (up to Llama-3.1-70B; Appendices A–D) read the weights only, without forward passes; the head-level analysis of all 12 models (Appendix D) took $\sim$37 minutes on a single H200. The CIFAR-10 CNN (Appendix C) and the flat-direction projection runs (Appendix I; one conjugate-gradient solve per step) train on a single GPU.

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: This is a foundational research paper analyzing weight matrix structure. It does not involve human subjects, private data, or applications with direct societal risks.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Section 6 (Discussion, *Practical implications*) discusses potential applications—the coordinate system that layer pruning and cross-layer parameter sharing would have to respect, and a first weight-sharing probe; whether continuity yields practical gains is stated there as an open question. The paper is foundational research on weight-matrix structure and does not introduce new capabilities or data; accordingly, no direct negative societal risks are identified.

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: The paper does not release any pre-trained models or datasets. Analysis code poses no misuse risk.

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: Datasets (cited where used): MNIST [14] (MIT, per its Hugging Face dataset card), Fashion-MNIST [33] (MIT), SVHN [34] (non-commercial use, as stated by its creators), EMNIST-Letters [32], CIFAR-10 and CIFAR-100 [31] (no license stated by their creators), WikiText-103 [15] (CC BY-SA 3.0). Pretrained checkpoints, each cited where used, with their licenses: Llama-3.1-8B and -70B (Llama 3.1 Community License); GPT-2 and GPT-2 XL (MIT); Qwen2.5-1.5B-Instruct, Qwen3-0.6B/4B/8B, OLMo-2-0425-1B, SmolLM-135M, Falcon-7B, ViT-Base/16 and ResNet-50 (Apache 2.0); Gemma-3-12B (Gemma Terms of Use); EXAONE-4.0-32B (EXAONE AI Model License); tiny_starcoder_py and SantaCoder (BigCode OpenRAIL-M); DeepSeek-V2-Lite (DeepSeek Model License). All are used for weight analysis only and are not redistributed. All prior work is properly cited.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper does not introduce new datasets or pretrained models. It releases core code (metrics, toy-MLP training, pretrained-LLM weight analysis) under the MIT license at `https://github.com/smallhours19/geometric-continuity`, with a README describing dependencies, usage, and the seeds used.

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: No crowdsourcing or human subjects research is involved.

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 involved.

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: [N/A]

    Justification: LLMs (Llama-3.1-8B) are used as an analysis subject, not as a methodological component. The model is analyzed via SVD of its weight matrices, which is a standard mathematical operation.