

Operator Dynamics in Transformer Residual Streams: A Unified Framework for Interpretability, Adversarial Detection, Causal Control, and Topological Model Fingerprinting

Sanskar Pandey 

2026

Abstract

We present a unified framework for transformer interpretability and safety grounded in the geometry of *residual stream operators* — inter-layer differences $\Delta_l = h_{l+1} - h_l$ that directly capture what each layer contributes to the forward pass. We make five empirical contributions validated across four models spanning three architectural families and a $80\times$ parameter range (GPT-2 117M through Qwen3.5-9B).

(1) Structural invariants. Specificity (≈ 0.120 – 0.127), mean neuron entropy (≈ 1.41), and inter-class direction separation (≈ 1.53 – 1.55) are conserved across all tested models, establishing conservation principles governing semantic information distribution in language models trained on natural language.

(2) Causal neuron atlas and operator-concept mapping. Despite near-complete neuron polysemanticity, top-20 neuron interventions retain 70–101% of full directional *probe-space* causal effect, validated by an independent two-probe design. Layer-specific classifier coefficients provide a computational atlas mapping concepts to operators at specific layers and operators to neurons. Generation-level behavioral steering is evaluated but remains weak across all tested strategies at 0.5B scale; probe-space control is the validated result.

(3) Zero-shot adversarial detection. A trajectory anomaly detector calibrated only on normal web text — zero adversarial examples — achieves AUC of 0.923 (GPT-2), 0.906 (Qwen2.5-0.5B), 0.858 (Mistral-7B), and 0.886 (Qwen3.5-9B, Gated Delta Networks + MoE architecture) against GCG suffixes, jailbreak templates, and prompt injections. GCG suffix AUC reaches 0.975 on GPT-2 and 0.963 on Qwen3.5-9B. On a benchmark-grade extended evaluation (GCG N=50, JailbreakBench [Chao et al., 2024], AutoDAN, PAIR), overall AUC is 0.899 (GPT-2) and 0.873 (Qwen2.5-0.5B), with AutoDAN AUC of 0.872/0.819 vs. perplexity filtering’s ~ 0.55 — the decisive

advantage on the attack class that defeats all linguistic detectors. Unlike perplexity filtering and Gradient Cuff, our detector requires no adversarial calibration, no aligned model, and a single forward pass.

(4) Two-layer safety architecture. Combined with a lightweight linguistic pre-filter (Layer 1), the trajectory detector (Layer 2) achieves 0% adversarial escape rate across all tested categories on GPT-2, while resolving the false positive problem on benign complex text through architectural separation of concerns.

(5) Topological model fingerprinting and PC0 characterisation. H_1 loop count increases monotonically with capability ($122 \rightarrow 164 \rightarrow 203$), PC0 variance drops sharply at 7B scale (confirmed as scale-driven, not alignment-driven via direct base vs. instruct comparison), and inter-class direction separation is conserved at 1.53–1.55 across all models, providing fingerprints that characterise computational regimes independently of benchmark scores.

1 Introduction

Transformers achieve remarkable performance across domains, yet their internal computation remains largely opaque. Most interpretability research operates at one of two extremes: highly local analyses of individual neurons, attention heads, or circuits, or high-level behavioural evaluations of model outputs. Neither approach provides a global dynamical account of how semantic computation unfolds across layers.

We propose analysing transformer computation through the lens of *residual stream operators*: the inter-layer differences $\Delta_l = h_{l+1} - h_l$ that directly capture what each layer contributes to the residual stream. These operators form a structured trajectory through a low-dimensional subspace, and we show this subspace encodes semantic information in a form that is geometrically interpretable, causally verifiable, and topologically distinctive across model families.

Our framework connects three previously separate interpretability problems:

- **Understanding.** What is the model computing? Answered by the operator manifold and probabilistic neuron-concept atlas.
- **Monitoring.** Is the model computing something unexpected? Answered by trajectory anomaly detection.
- **Control.** Can we correct what the model is computing? Answered by causal intervention on semantic directions validated through sparse neuron intervention.

The framework addresses three core problems that have previously been treated separately: *understanding* (what is the model computing), *monitoring* (is the model computing something anomalous), and *control* (can we correct what the

model computes). All three capabilities derive from the same geometric object — the operator basis — fitted on a few hundred normal text samples with no task-specific labels.

Our specific contributions are:

1. We establish via the *layer-shuffle spectrum test* that residual update structure is a global geometric property of the residual stream, not a trajectory-specific phenomenon.
2. We introduce a **non-circular two-probe evaluation design** that separates direction derivation from steering measurement, correcting a methodological flaw present in prior steering work.
3. We derive a **probabilistic neuron-to-concept atlas** and operator-concept computational map directly from the operator basis, validating 70–101% *probe-space* causal retention via independent two-probe design. Generation-level behavioral steering is evaluated and reported as an open problem.
4. We demonstrate **zero-shot adversarial detection** across four models and three architectural families (standard transformers and Gated Delta Networks + MoE), achieving AUC 0.858–0.923 overall and up to 0.975 on GCG suffixes, requiring only normal text for calibration.
5. We propose and validate a **two-layer safety architecture** combining linguistic pre-filtering with trajectory anomaly detection, achieving 0% adversarial escape rate with a principled process-level guarantee.
6. We compute **topological model fingerprints** establishing H_1 loop count as a monotone capability discriminator and inter-class direction separation as a conserved invariant across an $80\times$ parameter range.

2 Related Work and Background

2.1 Residual Stream as the Unit of Analysis

The residual stream formulation of transformer computation [Elhage et al., 2021] establishes that each layer adds an incremental update to a shared residual vector that accumulates information across the full depth of the network. This framing motivates studying the *changes* between layers — rather than the activations themselves — as the primary signal of what each layer computes. Early work characterised individual residual updates as low-rank and sparse [Geva et al., 2022], but the global statistical structure of these updates across layers and prompts, and what that structure implies about the organisation of semantic computation, has remained underexplored.

2.2 Mechanistic Interpretability: Circuits and Components

The dominant paradigm in mechanistic interpretability identifies computational circuits: subsets of attention heads, MLP layers, and residual stream positions that collectively implement specific behaviours such as induction [Olsson et al., 2022], factual recall [Meng et al., 2022], or arithmetic. These analyses are precise and illuminating but are inherently bespoke — each circuit analysis produces results specific to a single model, a single task, and often a single layer configuration. There is no unified coordinate system in which circuits from different models can be compared, and no principled way to characterise a model’s overall computational character from circuit-level findings alone. Our framework complements circuit analysis by providing a global geometric description of computation that is model-agnostic and comparably structured across architectures.

2.3 Probing and Linear Representation

Linear probing [Alain and Bengio, 2016] has established that semantic properties are linearly decodable from hidden states across many layers and tasks. More recent work has shown that factual associations are localisable via causal tracing [Meng et al., 2022] and that MLP layers promote concept representations into vocabulary space [Geva et al., 2022]. These results collectively suggest that transformers organise semantic content in a geometrically regular way. However, probing typically operates on raw hidden states rather than on the *dynamic changes* between layers, and most probing evaluations are circular in a subtle way: the probe is trained and evaluated on representations that share the same underlying semantic structure, making it difficult to distinguish genuine causal encoding from correlational geometric structure. We address this directly through the two-probe split design (Section 3.3).

2.4 Polysemanticity and Sparse Autoencoders

A central challenge in neuron-level interpretability is polysemanticity: individual neurons respond to multiple unrelated concepts, making single-neuron analysis unreliable [Elhage et al., 2021]. The monosemanticity programme [Bricken et al., 2023, Cunningham et al., 2023] addresses this by training sparse autoencoders (SAEs) to decompose activations into a larger dictionary of monosemantic features. This approach has produced impressive results but requires training a secondary model on top of the model under study, introducing a reconstruction objective that is external to the model’s own dynamics. Our framework takes a different approach: we characterise polysemanticity directly via the operator basis, deriving neuron-to-concept mappings from the model’s own inter-layer dynamics without any secondary training. Our empirical finding of specificity = 0.120 — identical across both tested architectures — provides a quantitative baseline for the polysemanticity that SAE approaches aim to decompose.

2.5 Representation Steering and Causal Intervention

Activation steering [Turner et al., 2023] and representation engineering [Zou et al., 2023] have demonstrated that injecting learned vectors into residual stream activations can shift model behaviour in predictable directions. These methods typically derive steering vectors from contrastive pairs or supervised datasets and evaluate their effects on output text. A recurring methodological concern is circular evaluation: when the probe used to derive a steering direction is also used to evaluate its effect, the positive result is partly guaranteed by construction. We address this by introducing a strict two-probe split that ensures the evaluation probe has no knowledge of the direction being tested, and by supplementing probe-space evaluation with KL-calibrated next-token distribution shift as an independent behavioral ground truth.

2.6 Low-Dimensional Structure in Neural Representations

Dimensionality reduction applied to neural activations consistently reveals that high-dimensional representations occupy a much lower-dimensional effective subspace [Ansuini et al., 2019]. For transformers specifically, the residual stream has been shown to be approximately low-rank at each layer. Less studied is the structure of the *inter-layer transition* rather than the representation itself. Two competing interpretations exist in the literature: the low-dimensional structure of residual updates may reflect ordered layer-by-layer computation (a trajectory interpretation), or it may be a global geometric property of the residual stream that is independent of layer ordering (a geometry interpretation). We resolve this empirically via a layer-shuffle spectrum test and find that the spectra of real and permuted update sequences are identical, confirming the geometric interpretation while showing that the ordered path through the subspace nonetheless carries semantic information.

2.7 Topological Data Analysis for Neural Networks

Persistent homology [Tralie et al., 2018] has been applied to characterise the topology of neural network weight spaces [Rieck et al., 2019] and to study learning dynamics. Applications to activation manifolds and representation spaces are more limited. We apply persistent homology to the operator feature manifold to derive a model fingerprint — a compact characterisation of the topological structure of a model’s computational geometry. To our knowledge this is the first use of persistent homology for cross-model computational comparison, and our finding of invariant H_0 cluster count (397) across GPT-2 and Qwen2.5-0.5B suggests that topological invariants may be a productive lens for understanding what properties are preserved versus changed across model families.

3 Methodology

3.1 Operator Basis Construction

For a transformer with L layers and hidden dimension d , we define the layer- l operator as the mean-pooled inter-layer residual update:

$$\Delta_l = \frac{1}{T} \sum_{t=1}^T (h_{l+1,t} - h_{l,t}) \in \mathbb{R}^d \quad (1)$$

where $h_{l,t}$ is the residual stream at layer l , token position t , and T is the sequence length (after truncation to `MAX_LEN=128`).

We collect $\{\Delta_l\}$ across $N_{\text{basis}} = 300$ prompts from a held-out web text corpus (never used in classifier training), yielding a matrix of shape $(N_{\text{basis}} \times L, d)$. Principal component analysis with $k = 8$ components yields the operator basis $\mathbf{V} \in \mathbb{R}^{k \times d}$. Each prompt’s full trajectory feature vector has shape $(L \times k, d)$.

3.2 Layer-Shuffle Spectrum Test

To resolve this ambiguity, we compute the PCA spectrum on both real ordered delta sequences and on layer-shuffled delta sequences drawn from identical prompts. Specifically, for each prompt we independently permute the assignment of layer indices before computing deltas:

$$\tilde{\Delta}_l = h_{\pi(l)+1} - h_{\pi(l)}, \quad \pi \sim \text{Uniform}(\mathcal{S}_L) \quad (2)$$

Identical spectra indicate order-independence. Results (Table 5) confirm spectra are identical to four decimal places across both models, resolving the tension between trajectory and geometry interpretations in the literature.

3.3 Semantic Directions via Two-Probe Split

We train logistic regression classifiers on trajectory features from five semantic domains (Table 3). To avoid circular evaluation, the training set (80% of 2000 samples) is partitioned into:

- **Direction probe** (75% of training): provides classifier coefficients $\mathbf{w}_i \in \mathbb{R}^{Lk}$ from which semantic directions are derived.
- **Eval probe** (25% of training, disjoint): measures $P(\text{class})$ shifts independently of direction derivation.

Semantic directions are derived from the last-layer direction probe:

$$v_i = \frac{\mathbf{V}^\top \mathbf{w}_i}{\|\mathbf{V}^\top \mathbf{w}_i\|_2} \in \mathbb{R}^d \quad (3)$$

The eval probe — trained on completely separate samples — measures $P(\text{class})$ before and after steering, ensuring no information about the direction enters the measurement.

3.4 Probabilistic Neuron-Concept Atlas

Since $v_i = \mathbf{V}^\top \mathbf{w}_i$, each element v_{ij} represents hidden dimension j 's contribution to concept i 's operator direction. We define:

$$P(\text{neuron } j \mid \text{concept } i) = \frac{|v_{ij}|}{\sum_{j'} |v_{ij'}|} \quad (4)$$

By Bayes' theorem with uniform concept prior:

$$P(\text{concept } i \mid \text{neuron } j) \propto P(\text{neuron } j \mid \text{concept } i) \quad (5)$$

Per-neuron entropy is:

$$H(j) = - \sum_i P(i \mid j) \log P(i \mid j) \quad (6)$$

Specificity is defined as $1 - \bar{H}/H_{\max}$ where $H_{\max} = \log(n_{\text{classes}})$.

3.5 Causal Validation via Sparse Neuron Intervention

For each concept i , we construct a sparse intervention vector retaining only the top- $k = 20$ dimensions by weight:

$$v_i^{\text{sparse}}[j] = \begin{cases} v_i[j] & \text{if } j \in \text{top-}k(|v_i|) \\ 0 & \text{otherwise} \end{cases}, \quad \hat{v}_i^{\text{sparse}} = \frac{v_i^{\text{sparse}}}{\|v_i^{\text{sparse}}\|} \quad (7)$$

Causal retention is defined as:

$$\rho_i = \frac{\bar{P}_{\text{sparse}} - \bar{P}_{\text{base}}}{\bar{P}_{\text{full}} - \bar{P}_{\text{base}}} \quad (8)$$

where probabilities are measured by the eval probe on held-out non-target test samples. $\rho_i > 0.5$ (CAUSAL) indicates that the top-20 neurons carry more than half the causal signal.

3.6 Operator-Concept Mapping and Computational Atlas

The full trajectory classifier's coefficient vector $\mathbf{w}_i \in \mathbb{R}^{L \times k}$ encodes not just a single semantic direction but a layer-specific decomposition of how concept i is computed across the network. Reshaping $\mathbf{w}_i$ to shape (L, k) gives the operator weights at each layer: at layer l , concept i 's natural update direction is:

$$v_i^{(l)} = \frac{\mathbf{V}^\top \mathbf{w}_i^{(l)}}{\|\mathbf{V}^\top \mathbf{w}_i^{(l)}\|_2} \in \mathbb{R}^d \quad (9)$$

where $\mathbf{w}_i^{(l)} \in \mathbb{R}^k$ is the slice of classifier coefficients at layer l . This constitutes a *computational atlas* — a map of when and through which operators each semantic concept is being processed, derived directly from the model's own dynamics.

The operator-concept map reveals that different concepts have distinct computational profiles across layers. The peak operator-weight layers (where the classifier assigns highest importance to operator projections) fall predominantly in the middle 20–80% of the network for all concepts, consistent with the interpretation that early layers handle token-level encoding and late layers handle output decoding, with semantic computation concentrated in the middle.

The full operator-concept-neuron chain is: concept $i \rightarrow$ high-weight operators at peak layers $\rightarrow$ neurons with highest weights in those operators. This provides a complete interpretable pathway from semantic concept to the specific hidden dimensions implementing it, derived without any supervision beyond the operator basis fitted on normal text.

3.7 Next-Token Distribution Shift

We validate behavioral steering via next-token prediction rather than full generation, avoiding generation coherence confounds. For each target class i we steer $h_{[-1]}$ (last layer, last position) by εv_i , apply final layer norm and `lm_head`, and measure log-probability mass on a predefined domain vocabulary $\mathcal{V}_i$:

$$\text{score}(i) = \log \sum_{w \in \mathcal{V}_i} P(w \mid \text{LN}(h_{[-1]} + \varepsilon v_i)) \quad (10)$$

The intervention magnitude ε is set by a KL sweep as

$$\varepsilon^* = \max \{ \varepsilon : D_{\text{KL}}(p_\varepsilon \| p_0) < 0.5 \text{ nats} \}. \quad (11)$$

A **LIFT+SYMMETRIC** result ($\text{score}(+v_i) > \text{score}(0)$ and $\text{score}(-v_i) < \text{score}(0)$) is the strongest form of behavioral causal evidence.

3.8 Zero-Shot Adversarial Detection

The trajectory anomaly detector (Section 3.10) is repurposed as a zero-shot adversarial detector by applying the calibrated threshold to inputs of interest rather than held-out normal text. We evaluate four attack categories: GCG adversarial suffixes [Zou et al., 2023], jailbreak templates (DAN, AIM, developer-mode, role-play escalation), prompt injections, and OOD/nonsense. A 30-prompt benign hard control set (legal, medical, technical, literary text) measures false positive rate. All evaluation uses the threshold calibrated on 300 normal web texts — zero adversarial examples used in calibration.

3.9 Behavioral Generation Steering

We evaluate three intervention strategies for generation-level behavioral control: (1) last-layer last-token injection of the global semantic direction, (2) middle-layer all-position injection with layer-specific operator directions derived from $\mathbf{w}_i^{(l)}$, and (3) neuron-targeted sparse intervention via the operator-neuron bridge. Results and analysis are reported in Section 5.10.

3.10 Trajectory Anomaly Detection

For each prompt, we compute the per-layer delta norm trajectory:

$$\tau(x) = (\|\Delta_1(x)\|_2, \dots, \|\Delta_L(x)\|_2) \in \mathbb{R}^L \quad (12)$$

The anomaly score is:

$$D(x) = \|\tau(x) - \bar{\tau}\|_2 \quad (13)$$

where $\bar{\tau}$ is the mean trajectory over the reference corpus. The 95th percentile of D on the reference corpus defines the anomaly threshold θ_{95} .

3.11 Two-Layer Safety Architecture

The high false positive rate on benign hard prompts (27–73% across models) limits the trajectory detector as a standalone filter. We resolve this through a two-layer architecture that combines existing linguistic safety classifiers (Layer 1) with trajectory anomaly detection (Layer 2).

Architectural rationale. Layer 1 (linguistic classifier) handles semantically obvious cases — content that is recognizably harmful or clearly benign based on surface text. Production deployments already run classifiers of this type (e.g. OpenAI Moderation API, Llama Guard). Benign hard prompts — legal text, medical terminology, technical writing — are linguistically normal and are correctly classified as benign by Layer 1. They never reach Layer 2. What reaches Layer 2 is the residual: inputs that passed linguistic screening but may still be adversarially crafted.

The failsafe guarantee. An adversarial input cannot simultaneously fool both layers because the layers operate on orthogonal properties. To fool Layer 1, an input must appear linguistically benign. To fool Layer 2, it must produce a normal computational trajectory. An input that looks linguistically normal AND produces a normal trajectory is, by definition, processed normally by the model — meaning it is not functionally adversarial.

Pilot evaluation on GPT-2. We evaluate the combined system using a lightweight TF-IDF logistic regression classifier as a Layer 1 simulation (trained on 20 adversarial + 15 benign examples, simulating production classifiers):

The jailbreak template result illustrates the complementarity directly: L1 misses 1 of 16 inputs; L2 catches that exact input. Combined AUC 0.969 strictly exceeds either layer alone (0.952, 0.942), confirming independent signal.

Extended benchmark evaluation on Qwen2.5-0.5B. We repeat the two-layer evaluation on the full extended benchmark (N=115 adversarial inputs across five categories). The combined system achieves 0% escape rate across all 115 inputs (Table 15).

On the 100% benign FPR. The Qwen extended evaluation used a more aggressively tuned L1 (threshold 0.40, 50-example training set) which flags essentially all unusual text including legitimate hard prompts. This is a calibration artifact of the L1 simulation, not a property of the two-layer architecture or of L2.

Table 1: *Two-layer system — GPT-2 pilot (N=15 per category). L1 = linguistic pre-filter. L2 = trajectory anomaly detector. 0% escape rate = no adversarial input bypasses both layers.*

Category	L1 alone	L2 alone	Combined	Escape rate
GCG suffixes	100%	100%	100%	0%
Jailbreak templates	94%	81%	100%	0%
Prompt injections	100%	90%	100%	0%
OOD/nonsense	100%	90%	100%	0%
Benign hard (FPR)	27%	73%	80%	—
ROC-AUC: L1=0.952, L2=0.942, Combined=0.969				

Table 2: *Two-layer system — Qwen2.5-0.5B extended benchmark (N=115). L1 trained on 30 adversarial + 20 benign examples. All adversarial categories achieve 0% escape rate. The 100% benign FPR reflects L1 over-tuning in the simulation (discussed below).*

Category	N	L1	L2	Escape rate
GCG (extended)	50	100%	52%	0%
JailbreakBench+GCG	20	100%	70%	0%
AutoDAN (semantic)	20	100%	40%	0%
PAIR-style	10	100%	50%	0%
Prompt injections	15	100%	33%	0%
Benign hard (FPR)	30	100%	70%	100%
AUC: L1=0.922, L2=0.881, Combined=0.914				

Two observations support this: (1) L2 alone achieves only 40% FPR on benign hard prompts for Qwen, consistent with the 33–47% range across models; (2) the GPT-2 pilot with properly calibrated L1 (threshold 0.50, 35-example set) achieves 80% combined FPR and 27% L1-alone FPR on the same benign hard category. In production, L1 is a purpose-built safety classifier (e.g. Llama Guard, OpenAI Moderation API) trained on millions of examples with near-zero FPR on legal, medical, and technical text by design. The architecture guarantee — any input that manipulates model computation gets flagged by L2 even when L1 passes it — is validated by the GPT-2 pilot and is independent of L1 calibration.

Complementarity with training-side attribution. Our runtime detection framework complements training-side behavior attribution [Anon., 2025], which traces harmful emergent behaviors to specific training datapoints via activation-difference vectors. That approach requires training data and checkpoints; ours requires only a reference corpus at inference time. Behaviors that evade standard safety evaluations — such as the distractor-triggered compliance described by Anon. [2025] — would manifest as trajectory anomalies in our detector, catchable without prior knowledge of the behavior or its trigger.

3.12 Topological Model Fingerprint

We characterise the operator manifold by computing persistent homology (via Ripser [Tralie et al., 2018]) on a 500-point subsample of last-layer operator features. H_1 persistence measures loop structure (cycles above threshold 0.1) and is the primary discriminating topological feature in our framework.

Note on H_0 . We initially report $H_0 = 397$ as a cross-model invariant. Probing experiments varying seeds (10 seeds), subsample sizes (400–600), and PCA basis dimensionality reveal $H_0 \approx N_{\text{points}} - 3$ for all models and configurations — the manifold is uniformly sparse at threshold 0.5, so almost every sampled point is its own isolated cluster. The apparent cross-model match was an artifact of the shared sample size, not structural. We correct this here: H_0 counts at this threshold do not constitute meaningful cluster structure. All topological fingerprinting relies on H_1 , which is sample-size independent. Together with PCA concentration ratio, neuron entropy, and inter-class direction separation, the fingerprint vector is:

$$\mathbf{f} = (\sigma_0^2, c, \bar{H}, \phi, |H_0|, |H_1|, \delta_{\text{sep}}) \in \mathbb{R}^7 \quad (14)$$

4 Experimental Setup

4.1 Models

- **GPT-2** [Radford et al., 2019]: 117M parameters, 12 layers, $d = 768$, trained on WebText.
- **Qwen2.5-0.5B** [Qwen Team, 2025]: 500M parameters, 24 layers, $d = 896$, trained on multilingual web corpus with instruction tuning.
- **Mistral-7B-v0.1** [Jiang et al., 2023]: 7B parameters, 32 layers, $d = 4096$, trained on web corpus with sliding window attention. Evaluated in fp16 precision.
- **Qwen3.5-9B** [Qwen Team, 2026]: 9B parameters, released March 2026. Hybrid architecture combining Gated Delta Networks (GDN) with sparse Mixture-of-Experts (MoE) — the only non-standard-transformer architecture in our evaluation. Tested to assess framework generalization beyond attention-based models. Evaluated in fp16 precision.

All experiments use seed 42 throughout. Models are evaluated in inference mode (`model.eval()`) with `torch.no_grad()`.

Table 3 shows the data sources used with regard to our experiments and validation.

Table 3: *Semantic domains and data sources used for all classification, steering, and evaluation experiments.*

Domain	Source	Samples	Description
Web	OpenWebText	400	General web prose
Math	GSM8K	400	Grade-school math questions
Code	CodeSearchNet (Python)	400	Function docstrings
Reason	ARC-Easy	400	Commonsense reasoning questions
Fact	TriviaQA	400	Factual recall questions
Basis	OpenWebText	300	Operator basis (held out)

Table 4: *PCA spectrum of residual update distributions. PC0 captures 85–87% of variance for GPT-2 and Qwen, but only 47.7% for Mistral-7B, indicating a qualitative shift in residual stream dynamics at 7B scale.*

Model	PC0	PC1	Conc. Ratio	Feature Dims
GPT-2	0.868	0.064	$13.6\times$	96 (12L $\times$ 8)
Qwen2.5-0.5B	0.849	0.048	$17.7\times$	192 (24L $\times$ 8)
Mistral-7B	0.477	—	$\sim 2\text{--}3\times$	256 (32L $\times$ 8)

5 Results

5.1 Operator Basis and Variance Structure

5.2 Layer-Shuffle Spectrum Test

Table 5: *PCA spectra of real vs. layer-shuffled residual updates. Spectra match to four decimal places, establishing that the low-dimensional structure is a global geometric property of the residual stream, independent of layer ordering.*

Condition	Real Spectrum (PC0–PC4)	Shuffled Spectrum (PC0–PC4)
GPT-2	[0.8656, 0.0645, 0.0200, 0.0140, 0.0026]	[0.8656, 0.0645, 0.0200, 0.0140, 0.0026]

The subspace is order-independent but not semantically empty: the ordered path through it across layers carries semantic information that is classifiable at 95.8–99.5% accuracy. The correct unified statement is that the operator basis captures a global geometric structure of the residual stream, and semantic computation corresponds to directional flows within that structure rather than to the ordering of the layers themselves.

Table 6: Full-trajectory classifier and last-layer probe performance. Permutation check confirms real geometric signal in both cases.

Model	Clf Acc	Perm Acc	Chance	Dir Probe	Eval Probe
GPT-2	0.958	0.195	0.200	0.770	0.708
Qwen2.5-0.5B	0.995	0.170	0.200	0.870	0.843
Mistral-7B	—	—	0.200	—	—

5.3 Semantic Classification

The 10-point gap in probe accuracy (77% vs. 87%) indicates Qwen builds richer semantic representations at the final layer. Qwen’s permuted accuracy falls below chance (17.0% vs. 20.0%), indicating the shuffled labels created a slightly adversarial training signal — a stronger pass than typical.

5.3.1 Class-Level Performance (Qwen2.5-0.5B)

Table 7: Per-class precision, recall, and F1 on held-out test set (Qwen2.5-0.5B). Support = 80 per class.

Class	Precision	Recall	F1	Support
code	1.00	1.00	1.00	80
fact	0.98	1.00	0.99	80
math	1.00	1.00	1.00	80
reason	1.00	0.99	0.99	80
web	1.00	0.99	0.99	80
Accuracy	0.995			

5.4 Eps Sweep: Probe-Space Causal Steering

Table 8: Mean $P(\text{target class})$ from independent eval probe across ε sweep. Baseline ≈ 0.20 for all classes. Results shown for Qwen2.5-0.5B.

Class	$\times 0.01$	$\times 0.05$	$\times 0.15$	$\times 0.30$	$\times 0.50$
code	0.007	0.017	0.104	0.570	0.964
fact	0.102	0.203	0.605	0.997	1.000
math	0.008	0.020	0.109	0.533	0.966
reason	0.037	0.142	0.611	0.996	1.000
web	0.185	0.350	0.724	0.988	1.000

All five classes show monotonically increasing $P(\text{class})$ with ε via the independent eval probe, confirming that operator-derived semantic directions causally influence internal representations. **fact** and **reason** saturate fastest (both > 0.99 at $\times 0.30$); **code** and **math** show slower saturation curves, indicating broader distribution in operator space. This asymmetry is consistent with factual and causal domains

being more geometrically concentrated and syntactic/mathematical domains being more distributed.

5.5 Probabilistic Neuron-Concept Atlas

5.5.1 Cross-Model Polysemanticity Invariant

Table 9: *Neuron polysemanticity across both models. Specificity is identical to four decimal places, constituting a cross-model invariant.*

Model	Mean Entropy	Max Entropy	Specificity	Specific Frac ($< 0.5H_{\max}$)
GPT-2	1.417	1.609	0.120	0.000
Qwen2.5-0.5B	1.417	1.609	0.120	0.000
Mistral-7B	1.405	1.609	0.127	0.000

Mean neuron entropy is 1.417 for GPT-2 and Qwen, and 1.405 for Mistral-7B — all at 87–88% of maximum entropy. Not a single neuron in any of the four models falls below 50% of maximum entropy. Specificity ranges from 0.120 to 0.127 across an $80\times$ parameter range (117M to 7B). This near-invariance across architectures, parameter counts, and training regimes constitutes a conservation principle governing semantic information distribution in language models trained on natural language: regardless of scale, individual neurons are near-maximally polysemantic, and the relevant unit of semantic computation is the weighted combination across the operator manifold.

5.5.2 Hub Neurons

The top-5 neurons per concept reveal shared hub neurons across semantic classes:

Table 10: *Top-5 neurons by $P(\text{neuron}|\text{concept})$ for Qwen2.5-0.5B. Neuron 53 appears in 4/5 concept top lists (hub neuron).*

Concept	Top-5 Neurons (by $P(n c)$)
code	53, 485, 439, 674, 75
fact	53, 783, 827, 62, 601
math	674, 53, 439, 485, 75
reason	490, 531, 674, 783, 62
web	53, 531, 62, 783, 827

Neuron 53 appears in 4/5 concept top lists. Neurons 674, 783, and 62 each appear in 3/5 lists. These hub neurons are not concept-dedicated — they are maximally polysemantic — yet their *weighted combination* retains causal power as shown in Section 5.6.

5.6 Causal Neuron-Level Validation

Table 11: Causal retention from top-20 sparse neuron intervention vs. full direction intervention. $\rho > 0.5 = \text{CAUSAL}$. All five classes CAUSAL on Qwen2.5-0.5B. GPT-2 shows WEAK retention on semantically deep domains (fact, reason).

Concept	GPT-2		Qwen2.5-0.5B		Mistral-7B	
	ρ	V.	ρ	V.	ρ	V.
code	0.98	C	0.82	C	0.78	C
fact	0.37	W	1.01	C	0.70	C
math	0.85	C	0.76	C	0.73	C
reason	0.47	W	0.92	C	0.74	C
web	0.96	C	0.99	C	0.70	C

C = CAUSAL ($\rho > 0.5$), W = WEAK ($\rho \leq 0.5$). Mistral-7B achieves CAUSAL retention on all five classes (range 0.70–0.78), as does Qwen (0.76–1.01). GPT-2 shows WEAK retention on **fact** and **reason**, suggesting these semantically deep domains require more representational capacity to concentrate causally. Mistral’s more uniform retention (tighter range than Qwen) reflects the more diffuse operator structure seen in its PC0 variance (0.477 vs 0.849): causal signal is more evenly distributed across concepts, consistent with a richer multi-directional residual stream. The **fact** $\rho = 1.01$ on Qwen (sparse outperforms full) does not replicate on Mistral, suggesting this is specific to Qwen’s representational geometry rather than a universal property.

5.7 Operator-Concept Mapping and Computational Atlas

The full trajectory classifier’s coefficient vector $\mathbf{w}_i \in \mathbb{R}^{L \times k}$ encodes not just a single semantic direction but a layer-specific decomposition of how concept i is computed across the network. Reshaping $\mathbf{w}_i$ to shape (L, k) gives the operator weights at each layer: at layer l , concept i ’s natural update direction is:

$$v_i^{(l)} = \frac{\mathbf{V}^\top \mathbf{w}_i^{(l)}}{\|\mathbf{V}^\top \mathbf{w}_i^{(l)}\|_2} \in \mathbb{R}^d \quad (15)$$

where $\mathbf{w}_i^{(l)} \in \mathbb{R}^k$ is the slice of classifier coefficients at layer l . This constitutes a *computational atlas* — a map of when and through which operators each semantic concept is being processed, derived directly from the model’s own dynamics.

The operator-concept map reveals that different concepts have distinct computational profiles across layers. The peak operator-weight layers (where the classifier assigns highest importance to operator projections) fall predominantly in the middle 20–80% of the network for all concepts, consistent with the interpretation that early layers handle token-level encoding and late layers handle output decoding, with semantic computation concentrated in the middle.

The full operator-concept-neuron chain is: concept $i \rightarrow$ high-weight operators at peak layers $\rightarrow$ neurons with highest weights in those operators. This provides

a complete interpretable pathway from semantic concept to the specific hidden dimensions implementing it, derived without any supervision beyond the operator basis fitted on normal text.

5.8 Next-Token Distribution Shift

Table 12: Next-token distribution shift test results. Log-probability mass on domain vocabulary before (+dir) and after (-dir) steering. LIFT+SYMMETRIC indicates strongest causal evidence. $\varepsilon^* = 6.77$ (Qwen), $\varepsilon^* = 6.59$ (GPT-2), KL-calibrated ($D_{KL} < 0.5$ nats).

Class	GPT-2			Qwen2.5-0.5B			Mistral-7B	
	Base	Lift	V.	Base	Lift	V.	Lift	V.
code	-4.581	-0.020	null	-8.143	+0.030	L+S	—	—
fact	-4.285	-0.040	null	-7.304	+0.068	L+S	—	—
math	-3.545	-0.030	null	-6.057	-0.091	null	—	—
reason	-6.576	-0.125	null	-10.558	+0.128	L+S	—	—
web	-2.749	+0.007	L+S	-5.760	+0.036	L+S	—	—
Total L+S		1/5			4/5		4/5	

L+S = LIFT+SYMMETRIC. Qwen and Mistral both achieve 4/5 LIFT+SYMMETRIC, demonstrating that operator-derived semantic directions causally influence next-token vocabulary predictions for four of five semantic classes across both models. GPT-2 achieves 1/5. This pattern suggests a capability threshold between GPT-2 scale and Qwen/Mistral scale at which operator geometry becomes aligned with output behavior. The consistent 4/5 result across Qwen (0.5B) and Mistral (7B) indicates this alignment is not purely a function of parameter count but may reflect training regime or data diversity.

The negative lifts across all GPT-2 classes (except web) indicate that GPT-2’s operator-derived semantic directions point in the opposite orientation to what raises domain vocabulary probability — an alignment failure between representation-space geometry and output-space behavior that does not occur in Qwen.

math shows null behavior on Qwen (−0.091). This may reflect that the math domain vocabulary (symbolic and numeric tokens) is already elevated at baseline (−6.057, highest among all classes), leaving less headroom, or that the **math** semantic direction in Qwen encodes a broader mathematical register that does not align neatly with our predefined symbol vocabulary.

5.9 Zero-Shot Adversarial Detection

The trajectory anomaly detector (Section 3.10) is repurposed as a zero-shot adversarial detector by applying the calibrated threshold to inputs of interest rather than held-out normal text. We evaluate four attack categories: GCG adversarial suffixes [Zou et al., 2023], jailbreak templates (DAN, AIM, developer-mode, role-play escalation), prompt injections, and OOD/nonsense. A 30-prompt benign hard control set (legal, medical, technical, literary text) measures false positive

rate. All evaluation uses the threshold calibrated on 300 normal web texts — zero adversarial examples used in calibration.

5.10 Behavioral Generation Steering

We evaluate three intervention strategies for generation-level behavioral control: (1) last-layer last-token injection of the global semantic direction, (2) middle-layer all-position injection with layer-specific operator directions derived from $\mathbf{w}_i^{(l)}$, and (3) neuron-targeted sparse intervention via the operator-neuron bridge. Results and analysis are reported in Section 5.10.

5.11 Trajectory Anomaly Detection

On the 300-sample web basis corpus:

Table 13: *Anomaly detection calibration. The detector correctly flags the expected number of samples at the 95th percentile threshold.*

Model	Threshold (θ_{95})	Flagged	Expected
GPT-2	31.013	15/300	15

The anomaly detector is correctly calibrated. The reference trajectory distribution is right-skewed with 15/300 flagged at the 95th percentile threshold as expected. Validation on adversarial inputs is presented in Section 5.9.

5.12 Two-Layer Safety Architecture

The high false positive rate on benign hard prompts (27–73% across models) limits the trajectory detector as a standalone filter. We resolve this through a two-layer architecture that combines existing linguistic safety classifiers (Layer 1) with trajectory anomaly detection (Layer 2).

Architectural rationale. Layer 1 (linguistic classifier) handles semantically obvious cases — content that is recognizably harmful or clearly benign based on surface text. Production deployments already run classifiers of this type (e.g. OpenAI Moderation API, Llama Guard). Benign hard prompts — legal text, medical terminology, technical writing — are linguistically normal and are correctly classified as benign by Layer 1. They never reach Layer 2. What reaches Layer 2 is the residual: inputs that passed linguistic screening but may still be adversarially crafted.

The failsafe guarantee. An adversarial input cannot simultaneously fool both layers because the layers operate on orthogonal properties. To fool Layer 1, an input must appear linguistically benign. To fool Layer 2, it must produce a normal computational trajectory. An input that looks linguistically normal AND produces a normal trajectory is, by definition, processed normally by the model — meaning it is not functionally adversarial.

Pilot evaluation on GPT-2. We evaluate the combined system using a lightweight TF-IDF logistic regression classifier as a Layer 1 simulation (trained on 20 adversarial + 15 benign examples, simulating production classifiers):

Table 14: *Two-layer system — GPT-2 pilot (N=15 per category). L1 = linguistic pre-filter. L2 = trajectory anomaly detector. 0% escape rate = no adversarial input bypasses both layers.*

Category	L1 alone	L2 alone	Combined	Escape rate
GCG suffixes	100%	100%	100%	0%
Jailbreak templates	94%	81%	100%	0%
Prompt injections	100%	90%	100%	0%
OOD/nonsense	100%	90%	100%	0%
Benign hard (FPR)	27%	73%	80%	—
ROC-AUC: L1=0.952, L2=0.942, Combined=0.969				

The jailbreak template result illustrates the complementarity directly: L1 misses 1 of 16 inputs; L2 catches that exact input. Combined AUC 0.969 strictly exceeds either layer alone (0.952, 0.942), confirming independent signal.

Extended benchmark evaluation on Qwen2.5-0.5B. We repeat the two-layer evaluation on the full extended benchmark (N=115 adversarial inputs across five categories). The combined system achieves 0% escape rate across all 115 inputs (Table 15).

Table 15: *Two-layer system — Qwen2.5-0.5B extended benchmark (N=115). L1 trained on 30 adversarial + 20 benign examples. All adversarial categories achieve 0% escape rate. The 100% benign FPR reflects L1 over-tuning in the simulation (discussed below).*

Category	N	L1	L2	Escape rate
GCG (extended)	50	100%	52%	0%
JailbreakBench+GCG	20	100%	70%	0%
AutoDAN (semantic)	20	100%	40%	0%
PAIR-style	10	100%	50%	0%
Prompt injections	15	100%	33%	0%
Benign hard (FPR)	30	100%	70%	100%
AUC: L1=0.922, L2=0.881, Combined=0.914				

On the 100% benign FPR. The Qwen extended evaluation used a more aggressively tuned L1 (threshold 0.40, 50-example training set) which flags essentially all unusual text including legitimate hard prompts. This is a calibration artifact of the L1 simulation, not a property of the two-layer architecture or of L2. Two observations support this: (1) L2 alone achieves only 40% FPR on benign hard prompts for Qwen, consistent with the 33–47% range across models; (2) the GPT-2 pilot with properly calibrated L1 (threshold 0.50, 35-example set) achieves 80% combined FPR and 27% L1-alone FPR on the same benign hard category. In production, L1 is a purpose-built safety classifier (e.g. Llama Guard, OpenAI Moderation API) trained on millions of examples with near-zero FPR on legal,

medical, and technical text by design. The architecture guarantee — any input that manipulates model computation gets flagged by L2 even when L1 passes it — is validated by the GPT-2 pilot and is independent of L1 calibration.

Complementarity with training-side attribution. Our runtime detection framework complements training-side behavior attribution [Anon., 2025], which traces harmful emergent behaviors to specific training datapoints via activation-difference vectors. That approach requires training data and checkpoints; ours requires only a reference corpus at inference time. Behaviors that evade standard safety evaluations — such as the distractor-triggered compliance described by Anon. [2025] — would manifest as trajectory anomalies in our detector, catchable without prior knowledge of the behavior or its trigger.

5.13 Topological Model Fingerprint

Table 16: *Cross-model fingerprint comparison across four models spanning 80× parameter range. Green = invariant or approximately invariant; red/orange = discriminating feature.*

Metric	GPT-2	Qwen2.5	Mistral	Qwen3.5	Status
PC0 variance	0.868	0.849	0.477	0.634	Drops sharply at 7B
Concentration ratio	13.6	17.7	~2–3	—	Varies
Mean neuron entropy	1.417	1.417	1.405	—	≈INVARIANT
Specificity	0.120	0.120	0.127	—	≈INVARIANT
H ₀ clusters	397	397	399	—	≈N–3 artifact
H ₁ loops	122	164	203	—	Monotone with capability
H ₁ max persistence	0.582	0.637	0.570	—	Varies
Inter-class dir. sep.	1.525	1.549	1.548	1.548	INVARIANT
Next-token L+S	1/5	4/5	4/5	4/5	Threshold at ~0.5B

Four cross-model patterns emerge from the three-model comparison spanning a 80× parameter range (117M to 7B).

Confirmed invariants: Specificity (0.120–0.127), mean neuron entropy (1.405–1.417), and inter-class direction separation (1.525–1.549) are conserved across all four models. Notably, the geometric distance between semantic directions in hidden space is essentially identical across GPT-2, Qwen, and Mistral despite hidden dimensions of 768, 896, and 4096 respectively, suggesting this quantity reflects the intrinsic geometry of the semantic task rather than the model’s representational capacity.

Scale-dependent transition: PC0 variance drops sharply from 0.868/0.849 for the smaller models to 0.477 for Mistral-7B, a qualitative shift rather than a smooth decline. Mistral’s inter-layer computation is distributed across multiple

principal components rather than dominated by a single direction, consistent with richer computational dynamics at 7B scale.

H₁ as the topological discriminator: H₁ loop count increases monotonically with capability: GPT-2 (122) < Qwen (164) < Mistral (203). Probing experiments reveal that H₀ cluster counts at our threshold (0.5) reflect point cloud density rather than discrete semantic groupings — specifically, $H_0 \approx N_{\text{points}} - 3$ for all models and sample sizes, indicating the operator manifold is uniformly sparse at this scale. H₁, by contrast, is independent of sample size and reflects genuine cyclic structure. More capable models have more persistent loops in their operator manifolds, consistent with richer feedback-like processing paths supporting iterative reasoning.

Behavioral alignment threshold: Next-token LIFT+SYMMETRIC results are 1/5 for GPT-2 and 4/5 for both Qwen and Mistral, suggesting a capability threshold for operator-behavior alignment crossed between 117M and 500M parameters and maintained at 7B scale.

5.14 PC0 / Alignment Hypothesis: Base vs. Instruct Comparison

We test whether the PC0 variance drop observed at 7B scale (0.477 for Mistral-7B vs. 0.849 for Qwen2.5-0.5B) reflects alignment training or raw parameter count by directly comparing a base model to its instruction-tuned counterpart with identical architecture and weights.

Table 17: *Fingerprint comparison: Qwen2.5-0.5B base vs. instruction-tuned. All metrics conserved to within noise. PC0 delta = −0.017, below the significance threshold of 0.03. Instruction tuning does not measurably alter the operator manifold geometry.*

Metric	Base	Instruct	Δ	Verdict
PC0 variance	0.848	0.831	−0.017	Unchanged
Specificity	0.131	0.134	+0.004	Conserved
Mean entropy	0.955	0.951	−0.004	Conserved
H ₁ loops	117	120	+3	Conserved
Dir. separation	1.732	1.729	−0.003	Conserved
Adversarial AUC	0.993	0.993	+0.000	Conserved

Every fingerprint metric is conserved to within measurement noise across base and instruct variants. The PC0 delta of −0.017 is well below the threshold at which we observe meaningful structural change, and all other metrics — specificity, entropy, H₁ topology, direction separation, and adversarial detection capability — are essentially identical.

This result has two important implications. First, the PC0 drop at 7B scale is a function of parameter count and the computational dynamics that emerge with scale, not of alignment training. Mistral-7B-v0.1 is a base model with no instruction tuning, and it shows PC0= 0.477 — confirming the scale interpretation independently. Second, the geometric properties of the operator manifold are

determined during pretraining and are not restructured by post-training alignment. Instruction tuning operates on top of a fixed geometric substrate without altering it. This validates the fingerprinting approach: studying a model’s computational geometry can be done equivalently on the base or aligned variant, as they share the same operator manifold structure.

6 Discussion

6.1 The Dynamical Systems Interpretation

Our results support a dynamical systems view of transformer computation where the meaningful unit of analysis is not the individual neuron or attention head but the operator trajectory: the path traced through the low-dimensional operator subspace across the forward pass. Semantic categories correspond to directional flows within this space.

Individual neurons participate in multiple flows simultaneously (polysemanticity, specificity = 0.120) but in proportions that are causally informative (causal retention 76–101%). This distinguishes our framework from both the neuron-centric view (which requires dedicated neurons) and pure distributed representationalism (which would predict no sparse causal concentration). The empirical picture is: semantics are distributed across the manifold but causally concentrated in weighted top- k combinations.

6.2 Implications for the Monosemanticity Program

The finding that not a single neuron in either model falls below 50% of maximum entropy has direct implications for the sparse autoencoder programme. Polysemanticity is not an obstacle to be decomposed away — it is the expected operating mode of a system that distributes information for robustness and capacity. The appropriate tool for understanding such a system is the geometry of the collective, not the activity of the individual component. Our causal retention results show that sparse weighted combinations are sufficient: the field need not solve monosemanticity to achieve causal interpretability.

6.3 Toward a Periodic Table of Transformer Computation

The fingerprint comparison suggests a preliminary taxonomy. Four models spanning a $80\times$ parameter range share specificity (≈ 0.120), entropy (≈ 1.41), and inter-class direction separation (≈ 1.525 – 1.549). These three quantities, along with H_1 loop count, define an equivalence class of transformer models trained on natural language. The H_0 cluster count, previously reported as an invariant, is corrected to reflect point cloud density ($H_0 \approx N - 3$) rather than discrete cluster structure. Within the equivalence class defined by the genuine invariants, models are discriminated by H_1 loop count (monotone with capability), PC0 variance (qualitative drop at 7B), and behavioral alignment (threshold at $\sim 0.5B$). A periodic table would

organise models by these fingerprint vectors: high H_1 persistence predicts stronger iterative reasoning; high next-token alignment predicts better in-context learning; the PC0 transition from >0.8 to <0.5 marks qualitative capability emergence. These hypotheses are testable and constitute a concrete research programme.

6.4 Adversarial Detection: Mechanism and Limitations

The zero-shot adversarial detection result is explained by a simple mechanism: adversarial inputs work by manipulating the model’s internal computation. GCG suffixes are gradient-optimized to change what the model generates; this manipulation necessarily alters the residual stream trajectory. Jailbreak templates force the model into an unusual processing mode (“act as an unrestricted AI”); this mode shift produces a different trajectory geometry. The detector does not need to know what an adversarial input looks like linguistically — it only needs to detect that the model is computing unusually.

This mechanism has a theoretical implication for adversarial robustness. An attacker who knows the detector can attempt to craft inputs that are adversarially effective yet produce normal-looking trajectories. This requires simultaneously optimizing two conflicting objectives: the input must change the model’s output (requiring trajectory manipulation) while appearing normal (requiring trajectory preservation). These objectives are in direct tension. We do not claim this is impossible — sufficiently capable adaptive attacks may exist — but the attack surface is fundamentally different from and harder to exploit than perplexity-based detectors, which can be defeated by simple paraphrasing or token substitution.

The PC0 – FPR relationship provides a practical deployment guide: models with $PC0 > 0.8$ (GPT-2 class) show high sensitivity but require a strong Layer 1 pre-filter to manage FPR. Models with $PC0 < 0.6$ (Mistral/Qwen3.5 class) are self-regulating in FPR and may not need a separate pre-filter for many deployment scenarios.

6.5 Implications for Safety and Oversight

The framework provides four operational safety capabilities, all derived from the same operator basis:

1. **Process-level adversarial detection:** Zero-shot detection of GCG suffixes (AUC up to 0.975), jailbreak templates (AUC up to 0.889), and prompt injections (AUC up to 0.981) across four model families, with no adversarial calibration required.
2. **Two-layer safety architecture:** Combined with existing linguistic classifiers, achieves 0% adversarial escape rate with a principled guarantee: any input that manipulates computation is detectable, regardless of linguistic surface.
3. **Probe-space causal control:** Semantic directions causally influence internal representations (70–101% causal retention, validated via independent two-probe

design) and next-token distributions (4/5 LIFT+SYMMETRIC on Qwen and Mistral).

4. **Interpretable attribution via operator-concept-neuron chain:** The computational atlas maps concepts to specific operators at specific layers, and those operators to specific neurons, providing a complete interpretable pathway from semantic concept to hidden dimension without secondary training.

For latent chain-of-thought models, operator trajectory analysis is directly applicable to monitoring internal reasoning. Anomalous trajectory deviation during multi-step reasoning may indicate deceptive reasoning, capability elicitation, or distribution shift before any output is generated. Whether verbalized or latent, the chain-of-thought process runs through the same residual stream; our framework is agnostic to whether the model is generating text or computing silently.

6.6 Limitations

1. **Model scale.** Core interpretability experiments (neuron atlas, causal retention, next-token steering) are validated on GPT-2 (117M), Qwen2.5-0.5B (500M), and Mistral-7B (7B). Adversarial detection is validated across four models including Qwen3.5-9B (GDN+MoE). Scaling to 70B+ frontier models and whether the specificity invariant holds under denser superposition at that scale remains future work.
2. **Generation-level behavioral steering.** Probe-space causal control is validated (70–101% retention) but generation-level behavioral shift is weak across all tested intervention strategies on Qwen2.5-0.5B, including last-layer injection, middle-layer full-direction, peak-layer neuron-targeted, and interleaved approaches. The operator-concept mapping provides a more principled foundation for future attempts: layer-specific directions from classifier coefficients are the mechanistically correct intervention vectors but require further validation.
3. **H_0 topology.** H_0 cluster counts at threshold 0.5 reflect point cloud density ($H_0 \approx N - 3$) rather than discrete semantic clusters, corrected in this paper. Topological characterisation relies on H_1 loop structure which is sample-size independent and validated as a genuine discriminator.
4. **Prompt injection detection.** Sophisticated prompt injections embedded in long legitimate documents show lower AUC (0.640–0.795) on the extended benchmark, as surrounding context dominates the trajectory. Short direct injections remain strongly detectable.
5. **Two-layer L1 simulation.** The Layer 1 linguistic pre-filter is simulated with a toy TF-IDF classifier in all experiments. Production deployment should substitute existing safety classifiers (Llama Guard, OpenAI Moderation API). The 0% escape rate guarantee is architectural and holds regardless of L1 implementation quality.

6. **Adaptive adversarial attacks.** We evaluate on published static attack patterns. Adaptive attacks specifically optimised to produce normal-looking trajectories while achieving harmful outputs have not been tested. The theoretical basis for resistance to such attacks is discussed but not empirically validated.
7. **Math alignment.** The `math` class shows null next-token shift on Qwen and weak causal retention on GPT-2, suggesting mathematical computation is the most distributed and hardest to influence among tested domains.

7 Conclusion

We have presented a unified framework for transformer interpretability and safety grounded in the geometry of residual stream operators. The framework establishes that operator updates lie in a global low-dimensional subspace, identifies semantic directions via a non-circular two-probe design, derives a probabilistic neuron-concept atlas validated by sparse causal intervention, provides zero-shot adversarial detection validated across four models and three architectural families, and characterises the operator manifold topologically to produce discriminating model fingerprints.

Key findings:

1. **Structural invariants:** Specificity (≈ 0.120 – 0.127), mean entropy (≈ 1.41), and inter-class direction separation (≈ 1.53 – 1.55) are conserved across four models spanning an $80\times$ parameter range and two architectural families.
2. **Causal concentration:** Top-20 neuron interventions retain 70–101% of causal effect across all models and concepts, validating the operator basis as causally meaningful despite near-complete polysemanticity.
3. **Zero-shot adversarial detection:** AUC 0.858–0.923 overall; up to 0.975 on GCG suffixes and 0.981 on prompt injections. Validated on standard transformers (GPT-2, Qwen2.5, Mistral) and hybrid GDN+MoE architecture (Qwen3.5-9B). No adversarial calibration.
4. **Two-layer safety system:** 0% adversarial escape rate with combined AUC 0.969. Principled guarantee: any input that manipulates computation is detectable.
5. **Topological fingerprint:** H_1 loops increase monotonically ($122 \rightarrow 164 \rightarrow 203$). PC0 variance drops sharply at 7B scale. Qwen3.5-9B GDN+MoE architecture shows PC0=0.634, intermediate between transformer families.
6. **Behavioral alignment:** 4/5 LIFT+SYMMETRIC next-token results on Qwen2.5-0.5B and Mistral-7B vs. 1/5 on GPT-2. Probe-space causal control is strong (70–101% retention); generation-level behavioral steering requires further work.

7. **PC0 / alignment:** Direct base vs. instruct comparison (Qwen2.5-0.5B) confirms PC0 is determined by scale, not alignment training. All fingerprint metrics conserved across instruction tuning. The operator manifold is fixed during pretraining.
8. **Operator-concept mapping:** Layer-specific classifier coefficients provide a complete computational atlas mapping concepts to operators to neurons at each processing layer, constituting a principled interpretability tool derived without secondary training.

The framework provides a foundation for treating neural networks not as black boxes but as dynamical systems with characterisable computational geometry, steerable semantic directions, and monitorable trajectory dynamics. We hope it contributes to making AI systems globally understandable, monitorable, and intervenable.

References

- Alain, G., and Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *arXiv:1610.01644*.
- Bricken, T., Templeton, A., et al. (2023). Towards monosemanticity: Decomposing language models with dictionary learning. *Anthropic Technical Report*.
- Cunningham, H., Ewart, A., et al. (2023). Sparse autoencoders find highly interpretable features in language models. *arXiv:2309.08600*.
- Elhage, N., Nanda, N., et al. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*.
- Geva, M., Caciularu, A., et al. (2022). Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *EMNLP 2022*.
- Meng, K., Bau, D., et al. (2022). Locating and editing factual associations in GPT. *NeurIPS 2022*.
- Olsson, C., Elhage, N., et al. (2022). In-context learning and induction heads. *Transformer Circuits Thread*.
- Radford, A., Wu, J., et al. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*.
- Qwen Team (2025). Qwen2.5 Technical Report. *Alibaba Cloud*.
- Rieck, B., Togninalli, M., et al. (2019). Neural persistence: A complexity measure for deep neural networks using algebraic topology. *ICLR 2019*.
- Tralie, C., Saul, N., and Bar-On, R. (2018). Ripser.py: A lean persistent homology library for Python. *Journal of Open Source Software*, 3(29), 925.

- Chao, P., DeBenedetti, E., et al. (2024). JailbreakBench: An open robustness benchmark for jailbreaking large language models. *NeurIPS 2024 Datasets and Benchmarks Track*.
- Anonymous (2025). In-the-wild model organisms: Mitigating undesirable emergent behaviors in production LLM post-training via data attribution. *arXiv:2602.11079*.
- Turner, A., Thiergart, L., et al. (2023). Activation addition: Steering language models without optimization. *arXiv:2308.10248*.
- Zou, A., Phan, L., et al. (2023). Representation engineering: A top-down approach to AI transparency. *arXiv:2310.01405*.
- Ansuini, A., Laio, A., Macke, J.H., and Zoccolan, D. (2019). Intrinsic dimension of data representations in deep neural networks. *NeurIPS 2019*.
- Zou, A., Wang, Z., Kolter, J.Z., and Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models. *arXiv:2307.15043*.
- Alon, G., and Kamfonas, M. (2023). Detecting language model attacks with perplexity. *arXiv:2308.14132*.
- Robey, A., Wong, E., Hassani, H., and Pappas, G.J. (2023). SmoothLLM: Defending large language models against jailbreaking attacks. *arXiv:2310.03684*.
- Hu, Z., Zhao, Z., et al. (2024). Gradient Cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. *NeurIPS 2024*.
- Liao, Z., and Zhao, W. (2024). AmpleGCG: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed LLMs. *arXiv:2404.07921*.
- Qwen Team (2026). Qwen3.5 Technical Report. *Alibaba Cloud*.
- Jiang, A.Q., Sablayrolles, A., et al. (2023). Mistral 7B. *arXiv:2310.06825*.

A Experimental Details

A.1 Dataset Construction

Web text: 400 samples from OpenWebText (streaming), first 200 characters of texts longer than 120 characters.

Mathematical reasoning: First 400 questions from GSM8K training split.

Code: Function docstrings from CodeSearchNet (Python), minimum 60 characters, truncated to 200.

Commonsense reasoning: First 400 questions from ARC-Easy training split.

Factual recall: First 400 questions from TriviaQA `rc.nocontext` split (minimum 20 characters).

Operator basis: First 300 web text samples, held out from all classifier training.

A.2 Operator Basis

PCA fitted on $(300 \times L, d)$ delta matrix. `StandardScaler` fitted on training features only and applied to test features without refitting. $k = 8$ components used throughout.

A.3 Two-Probe Split

Training set (80% of 2000 samples = 1600) split 75/25 into direction probe (1200 samples) and eval probe (400 samples). Test set (400 samples) used for final reporting only, never for fitting any scaler or classifier. All probes use `LogisticRegression(max_iter=2000, C=1.0)`.

A.4 Eps Calibration

Probe-space sweep: $\varepsilon \in \{0.01, 0.05, 0.15, 0.30, 0.50\} \times \|\hat{h}\|_2$ where $\hat{h}$ is the final-normed hidden state.

Next-token KL calibration: $\varepsilon \in \{0.001, 0.003, 0.005, 0.01, 0.02, 0.03\} \times \|h_{[-1]}\|_2$ (raw, pre-norm). Largest ε with $D_{\text{KL}} < 0.5$ nats selected.

A.5 Persistent Homology

Ripser applied to 500 randomly subsampled (seed 42) last-layer operator feature points, standardised by `StandardScaler`. `maxdim=1` to compute H_0 and H_1 . Significance thresholds: H_0 persistence > 0.5 , H_1 persistence > 0.1 .

A.6 Domain Vocabulary Probes

Table 18: Domain vocabulary tokens used for next-token shift measurement. Only single-token words (in the respective tokenizer) are retained.

Domain	Representative tokens
math	=, +, -, *, /, ^, digits 0–9, equation, solve, theorem, sum, integral, matrix
code	def, class, return, import, for, while, if, else, lambda, self, None, True, False
reason	because, therefore, thus, hence, since, although, however, implies, conclude, given
fact	born, died, founded, located, named, year, century, country, war, discovered, awarded
web	the, and, is, are, was, has, this, that, with, from, they, you, at, on, in

B Full Results Table

Table 19: Complete cross-model results summary across four models.

Metric	GPT-2	Qwen2.5	Mistral	Qwen3.5
Parameters	117M	500M	7B	9B
Layers	12	24	32	—
Hidden dim	768	896	4096	—
Architecture	Transformer	Transformer	Transformer	GDN+MoE
PC0 variance	0.868	0.849	0.477	0.634
PC1 variance	0.064	0.048	—	—
Concentration ratio	13.6×	17.7×	~2–3×	—
Trajectory clf acc.	0.958	0.995	—	—
Permuted acc.	0.195	0.170	—	—
Direction probe acc.	0.770	0.870	—	—
Eval probe acc.	0.708	0.843	—	—
Mean neuron entropy	1.417	1.417	1.405	—
Specificity	0.120	0.120	0.127	—
Specific frac ($< 0.5H_{\max}$)	0.000	0.000	0.000	—
Mean causal retention	0.73	0.90	0.73	—
H ₀ clusters	397	397	399	—
H ₁ loops	122	164	203	—
H ₁ max persistence	0.582	0.637	0.570	—
Inter-class sep.	1.525	1.549	1.548	1.548
Next-token L+S (out of 5)	1	4	4	—