Latent-Aligned Manifolds in Language Models: Geometry, Recovery, and Prompt Translations

Anonymous ACL submission

Abstract

Understanding how large language models maintain and manipulate internal task state remains a central challenge in mechanistic interpretability. Sequential tasks are known to induce low-dimensional structure in activation space, yet how task-defined state, representation geometry, and prompt modulation interact remains poorly understood. We introduce a formal framework that links task-induced latent states to latent-aligned manifolds in the residual stream, in which representations concentrate near low-dimensional trajectories reflecting latent state progression. Building on this perspective, we develop *Auto-Latent*, an unsupervised method that recovers ordered latent-aligned structure directly from activations without access to task rules or handcrafted annotations. Across controlled state-tracking tasks, system-level prompts primarily act as approximately translational offsets on task manifolds, preserving geometric structure while shifting their embedding in representation space. These translation-dominated responses and latent-aligned geometries persist under unsupervised recovery, indicating that prompt modulation and internal task state are governed by intrinsic geometric properties of model computation rather than task-specific annotation artifacts.

1 Introduction

As large language models (LLMs) achieve increasingly sophisticated capabilities, understanding their internal mechanisms has become essential for safety, alignment, and scientific insight. *Mechanistic interpretability* aims to reverse-engineer how models implement computation, moving beyond behavioral evaluation to understand the internal representations and transformations that support observed capabilities.

A broad range of mechanistic interpretability approaches have been proposed to probe internal computation. *Circuit analysis* decomposes models into sparse subgraphs of attention heads and neurons that implement specific behaviors (Olah et al., 2017; Ameisen et al., 2025; Kamath et al., 2025). *Sparse feature extraction* seeks monosemantic units via dictionary learning and autoencoders (Bricken et al., 2023; Marks et al., 2024). *Linear probes* test whether task-relevant information is linearly accessible at intermediate layers (Belrose et al., 2023; Ravfogel et al., 2022), while *activation editing* localizes and modifies specific knowledge representations (Meng et al., 2022; Turner et al., 2024). Work on *superposition* explains how models compress many features into limited dimensions (Elhage et al., 2022).

Additionally, *manifold-based interpretability* characterizes the global geometric structure of representations. For tasks that maintain implicit internal state across token sequences, such states organize representations into low-dimensional subspaces (Li et al., 2023; Hattan et al., 2024; Friedman et al., 2024) and evolve along structured geometric trajectories (Modell et al., 2025). Figure 1 summarizes the commonly adopted analysis pipeline, progressing from token sequences to residual activations and their induced manifold structure.

However, existing work lacks a systematic formalization of how such states are geometrically represented and how their evolution is encoded in manifold structure.

Inspired by observations of structured manifolds in sequential tasks (Gurnee et al., 2025), we address this by introducing a formal abstraction that bridges implicit internal state and manifold geometry, then developing *Auto-Latent* for unsupervised recovery. Leveraging this formalization, we investigate how system-level prompts affect manifold geometry. Figure 2 illustrates this approach: tasks with explicit state definitions enable supervised analysis, Auto-Latent recovers compa-

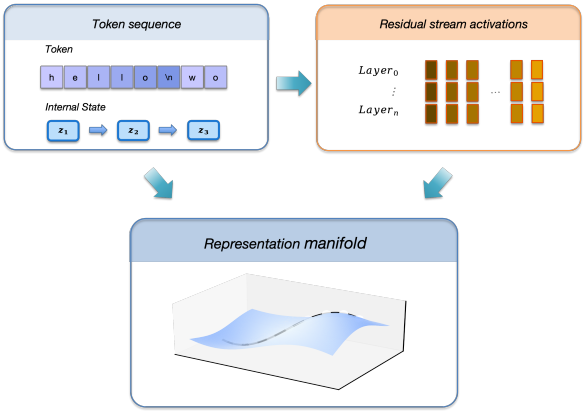

Figure 1: The manifold-based interpretability paradigm: token sequences induce distributions in residual stream activations that concentrate near low-dimensional manifolds.

rable structure unsupervised, and both reveal how prompts induce translational transformations.

Contributions.

- We formalize task-induced latent states and show how they organize representations into low-dimensional manifolds in the residual stream (Section 3).
- We introduce **Auto-Latent**, an unsupervised method that recovers manifold structure from activations, enabling geometric analysis across diverse tasks (Section 4).
- We discover that system-level prompts induce **approximately translational transformations** of manifolds, revealing geometric separation between prompt modulation and core task state (Section 6.2).

2 Background

Mechanistic interpretability asks how a fixed language model implements structured computation from discrete token sequences. We adopt a representation based perspective in which explanatory structure is encoded in residual stream activations. For a given task distribution, these activations induce a distribution in representation space. Manifold based interpretability hypothesizes that for structured tasks this distribution concentrates near low dimensional embedded manifolds, whose geometric properties encode underlying computational state.

Definition 1 (Token sequence)**.** *Fix a tokenizer that maps an input text to a finite token sequence. Let $\mathcal{X}$ denote the tokenizer vocabulary, viewed as a finite set of token ids. We write an input as a finite sequence $\mathbf{x} = (x_1, x_2, \ldots, x_T) \in \mathcal{X}^*$, where T is the sequence length and $x_t \in \mathcal{X}$.*

Table 1: Notation used throughout.

Symbol	Meaning
x_t	Token at position t
$\mathcal{X}$	Token vocabulary
$h_{\ell,t}$	Residual activation at layer ℓ, position t
H_ℓ	Residual stream dimensionality at layer ℓ
$\mathcal{M}_\ell$	Representation manifold at layer ℓ
ϕ_ℓ	Embedding map for manifold
d_ℓ	Intrinsic dimension of manifold ($d_\ell \ll H_\ell$)
z_t	Task-induced latent state at position t
$\mathcal{Z}$	Latent state space, $\lvert\mathcal{Z}\rvert = K$
F	Recursive transition: $z_{t+1} = F(z_t, x_{t+1})$
z_t^{explicit}	Explicit latent (from task rules)
$z_t^{\text{recovered}}$	Recovered latent (from Auto-Latent)
π	Mapping from explicit to recovered: $\pi : \mathcal{Z} \to \{1, \ldots, K\}$
$\mu_{\ell,k}$	Latent-conditioned mean: $\mathbb{E}[h_{\ell,t} \lvert z_t = k]$

Definition 2 (Residual stream activations)**.** *Fix a transformer model with L layers. Let $E : \mathcal{X} \to \mathbb{R}^{H_0}$ denote the token embedding map and let $p_t \in \mathbb{R}^{H_0}$ denote the positional embedding at position t. Define the initial residual stream vector as $h_{0,t}(\mathbf{x}) = E(x_t) + p_t$. For each layer $\ell \in \{1, \ldots, L\}$ and position t, let $h_{\ell,t}(\mathbf{x}) \in \mathbb{R}^{H_\ell}$ denote the residual stream activation produced by the forward pass of the first ℓ layers on input $\mathbf{x}$. Thus, for fixed model parameters, each $h_{\ell,t}$ is a deterministic function of the token sequence.*

Definition 3 (Representation manifold)**.** *Let $\mathcal{D}$ be a task distribution over valid pairs $(\mathbf{x}, t)$, where $\mathbf{x} \in \mathcal{X}^*$ and t indexes a position in $\mathbf{x}$. We say that computation at layer ℓ is organized along a representation manifold if there exist a smooth manifold $\mathcal{M}_\ell$ of intrinsic dimension d_ℓ with $d_\ell \ll H_\ell$ and a smooth embedding $\phi_\ell : \mathcal{M}_\ell \to \mathbb{R}^{H_\ell}$ such that activations concentrate near the embedded manifold. Concretely, for small $\varepsilon > 0$ and $\delta \in (0, 1)$,*

$$\Pr_{(\mathbf{x},t)\sim\mathcal{D}}\Big[\operatorname{dist}\big(h_{\ell,t}(\mathbf{x}), \phi_\ell(\mathcal{M}_\ell)\big) \le \varepsilon\Big] \ge 1 - \delta.$$

Under this hypothesis, manifold-based interpretability reduces to recovering $\phi_\ell(\mathcal{M}_\ell)$ from samples of $h_{\ell,t}(\mathbf{x})$ and estimating geometric invariants of the underlying structure, such as effective dimension, topology, curvature, and the induced progression of representations across token

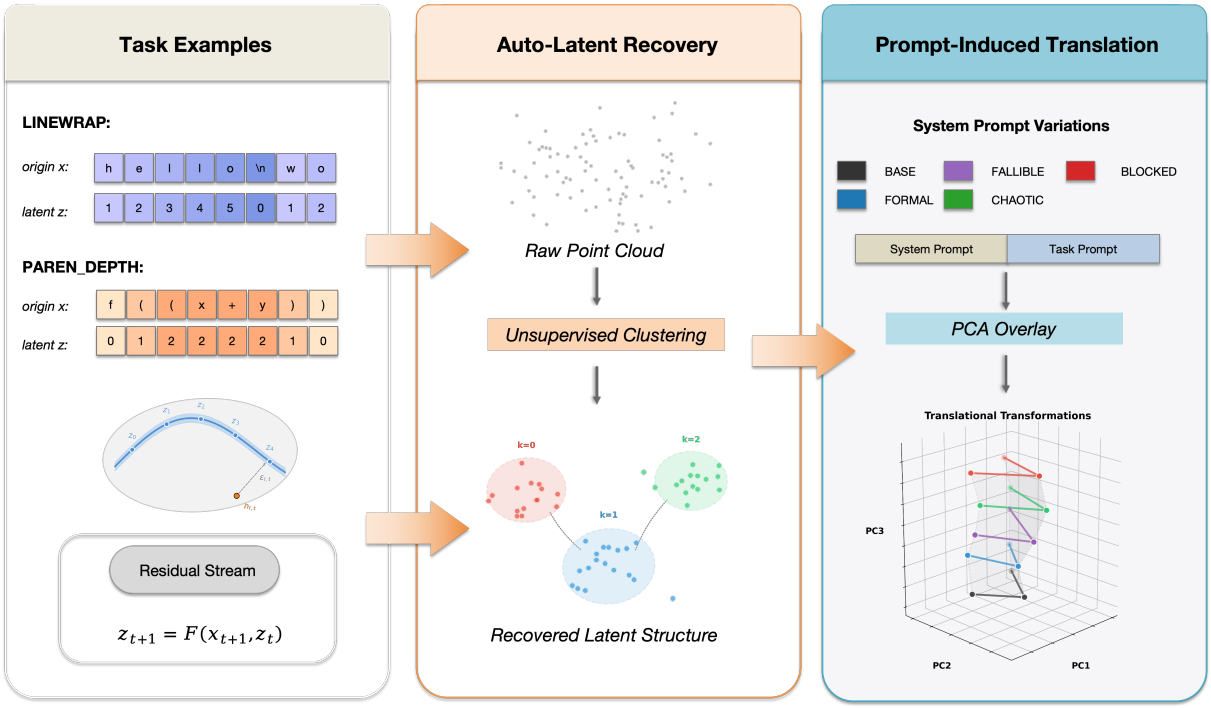

Figure 2: **Overview of the framework. (Left) Task Examples:** Two tasks with explicit latent statesLINEWRAP tracks character count within lines, PARENDEPTH tracks nesting depth. Both evolve via recursive transitions $z_{t+1} = F(z_t, x_{t+1})$ and produce structured trajectories in residual space. **(Center) Auto-Latent Recovery:** Unsupervised recovery via clustering transforms raw activation point clouds into ordered discrete structures. **(Right) Prompt-Induced Translation:** System prompts (BASE, FORMAL, FALLIBLE, CHAOTIC, BLOCKED) modulate manifold embeddings via approximately translational transformations, shown in 3D PCA overlay with preserved trajectory shapes across personas.

positions and layers, and relating these invariants to the underlying computational state of the task.

3 Latent State

3.1 Task-Induced Latent State

To connect the geometric structure of representations with implicit internal state, we introduce the notion of task-induced latent statea discrete abstraction that captures the internal computational state at each token position.

Definition 4 (Task-induced latent state)**.** *Let $\mathcal{X}$ be a finite token vocabulary, $\mathcal{D}$ a distribution over token sequences $\mathbf{x} = (x_1, \ldots, x_T) \in \mathcal{X}^*$, and $\mathcal{Z}$ a finite state space with $|\mathcal{Z}| = K < \infty$ equipped with a natural total order. We say that the task admits a task-induced latent state if there exist an initial state $z_0 \in \mathcal{Z}$ and a deterministic transition function*

$$F : \mathcal{Z} \times \mathcal{X} \to \mathcal{Z}$$

such that the latent state $z_t \in \mathcal{Z}$ at position t captures the task-relevant information and evolves according to

$$z_{t+1} = F(z_t, x_{t+1}).$$

Iterating F from z_0 defines a prefix mapping $f : \mathcal{X}^* \to \mathcal{Z}$ with $z_t = f(x_{1:t})$. The total order on $\mathcal{Z}$ reflects the natural progression of states, and f is many-to-one, meaning distinct prefixes may yield identical states.

Proposition 1 (Latent-aligned representation coordinate)**.** *Let $\mathcal{D}$ be a task distribution admitting a task-induced latent state $z_t \in \mathcal{Z}$ with finite $\mathcal{Z}$ (Definition 4). For a fixed layer ℓ, consider the induced distribution of residual stream activations $h_{\ell,t}(\mathbf{x}) \in \mathbb{R}^{H_\ell}$.*

Under the representation manifold hypothesis (Definition 3), the representation at layer ℓ admits a latent-aligned coordinate in the sense that activations are organized according to the latent state. Concretely, there exists a collection of decision regions $\{\mathcal{R}_z \subset \mathbb{R}^{H_\ell} : z \in \mathcal{Z}\}$ such that

$$\Pr_{(\mathbf{x},t)\sim\mathcal{D}} \left[h_{\ell,t}(\mathbf{x}) \in \mathcal{R}_{z_t} \right] \geq 1 - \delta$$

for some $\delta \in (0, 1)$.

Section 6.1 provides empirical support for this proposition: linear probes trained on residual activations achieve near-perfect accuracy in recovering latent states from the earliest layers, indicating well-separated decision regions (small δ), and discriminative subspaces saturate at low rank, confirming that these regions concentrate in low-dimensional subspaces.

3.2 Example: Geometric Encoding of Latent State

Recent mechanistic interpretability work provides concrete evidence that task-induced latent state can be encoded geometrically in the residual stream. In particular, Gurnee et al. (2025) show that for tasks involving character-level tracking, latent variables such as character count and distance to a line boundarystates fully determined by the input prefix but not explicitly present in the token sequenceare embedded as low-dimensional manifolds within small activation subspaces.

While such findings confirm that latent states can be encoded geometrically, they rely on handcrafted annotations specific to individual tasks, lacking a unified method to systematically investigate this geometric structure across diverse tasks.

4 Auto-Latent: Unsupervised Recovery of Latent-Aligned Structure

To overcome the reliance on handcrafted, task-specific latent annotations, we propose *Auto-Latent*, an unsupervised method that recovers latent-aligned structure directly from residual stream activations. The method outputs a discrete, ordered index $z_t \in \{1, \dots, K\}$ that identifies stable regions in representation space and respects the natural ordering of latent state transitions along the input sequence.

4.1 Setup and recovery target

Fix a layer ℓ and consider samples of residual stream activations

$$\mathcal{H}_\ell \;=\; \{\, h_{\ell,t}(\mathbf{x}) \;:\; (\mathbf{x}, t) \sim \mathcal{D} \,\} \subset \mathbb{R}^{H_\ell}.$$

Under the manifold hypothesis (Definition 3), $\mathcal{H}_\ell$ concentrates near a low-dimensional embedded structure. For tasks admitting explicit latent states $z_t^{\text{explicit}} \in \mathcal{Z}$ as in Definition 4, Auto-Latent recovers a coarse-grained partition into $K \ll |\mathcal{Z}|$ clusters that capture the essential geometric progression even when the underlying states admit fine-grained distinctions. Concretely, the clustering defines a many-to-one mapping $\pi : \mathcal{Z} \to \{1, \dots, K\}$, and the recovered index is

$$z_t^{\text{recovered}} = \pi(z_t^{\text{explicit}}) \in \{1, \dots, K\}.$$

Henceforth we denote the recovered index as $z_t := z_t^{\text{recovered}}$.

4.2 Auto-Latent procedure

Preprocessing. We extract activations $h_{\ell,t}(\mathbf{x})$ at layer ℓ for all tokens in sequences sampled from $\mathcal{D}$. To improve cluster quality, we optionally apply per-condition centering to remove global offsets, and restrict to tokens that participate in the task computation by filtering based on sequence position or other task-specific indicators.

Dimensionality reduction. Activations are standardized and projected to a d-dimensional subspace via PCA:

$$\bar{h}_\ell \;=\; \mathbb{E}[h_{\ell,t}], \qquad y_t \;=\; P_d\big(h_{\ell,t} - \bar{h}_\ell\big) \in \mathbb{R}^d,$$

where P_d denotes the projection onto the top d principal components. This restricts recovery to a low-dimensional chart capturing the dominant task-related variance.

We fit a K-component Gaussian mixture model to $\{y_t\}$ and assign discrete labels by maximum posterior probability. Since mixture components are unordered, we impose a canonical ordering by sorting clusters according to median token position:

$$s(k) \;=\; \text{median}\{\, t : \text{cluster}(y_t) = k \,\},$$

yielding a final ordered index that reflects left-to-right sequence progression.

4.3 Downstream geometric analysis

Definition 5 (Latent-conditioned trajectory)**.** *Under Proposition 1, for each layer ℓ and latent state $k \in \{1, \dots, K\}$, define the latent-conditioned mean*

$$\mu_{\ell,k} \;=\; \mathbb{E}[\, h_{\ell,t} \mid z_t = k \,].$$

The ordered set $\{\mu_{\ell,k}\}_{k=1}^K$ is the latent-conditioned trajectory at layer ℓ.

For tasks with explicit latent annotations (Definition 4), this trajectory directly reflects task-defined states. Auto-Latent provides the analogous construction for tasks without annotations, yielding a task-agnostic proxy.

With trajectories constructed from both explicit and recovered latent structures, we investigate how prompt variations affect manifold geometry. We find that prompt variations induce approximately translational transformations that preserve manifold structure (Section 6.2), and that recovered trajectories exhibit geometric properties comparable to explicit latent analysis (Section 6.3).

5 Experimental Setup

5.1 Model and Tasks

All experiments use the `gemma-2b-it` model (2B parameters, instruction-tuned). Key results are also replicated on `Phi-3-mini-4k-instruct` and `Qwen2.5-3B-Instruct`; cross-model comparisons are reported in Appendix D.

We select two tasks that instantiate the recursive latent structure defined in Section 3.

LINEWRAP. The latent variable is the running character count within the current line. The counter increments on non-newline characters and resets to 0 on newline. Let c_t denote the character at position t in the underlying character stream. The latent evolves under the local rule

$$z_{t+1} = \begin{cases} z_t + 1, & c_t \neq \texttt{"\textbackslash n"}, \\ 0, & c_t = \texttt{"\textbackslash n"}. \end{cases}$$

PARENDEPTH. The latent variable tracks parenthesis nesting depth. Let c_t denote the character at token position t. The latent update rule is:

$$z_{t+1} = \begin{cases} z_t + 1, & c_t = \texttt{"("}, \\ z_t - 1, & c_t = \texttt{")"}, \\ z_t, & \text{otherwise}. \end{cases}$$

Both tasks conform to the recursive latent structure in Definition 4. Character-level latent states are aligned to token indices using the tokenizer's offset mapping (Appendix B).

5.2 Prompt Variations

To study how system prompts modulate the manifold geometry without altering task semantics, we vary only the system-level instruction while keeping all task components fixed. We use five persona presets: BASE, FORMAL, BLOCKED, CHAOTIC, and FALLIBLE (Appendix A).

5.3 Geometric Metrics

To quantify how prompt variations modulate manifold geometry, we introduce three complementary metrics that compare trajectories under different personas.

Path-direction cosine. We define the discrete tangent as

$$\Delta\mu_{\ell,k}^{(p)} = \mu_{\ell,k+1}^{(p)} - \mu_{\ell,k}^{(p)}.$$

The path cosine is the mean cosine similarity of corresponding tangents:

$$\mathrm{PathCos}(\ell; p, q) = \mathbb{E}_{k \in \mathcal{K}} \cos\big(\Delta\mu_{\ell,k}^{(p)}, \Delta\mu_{\ell,k}^{(q)}\big).$$

Procrustes alignment error. Trajectories are first centered, then aligned via the classical orthogonal Procrustes solution (Schönemann, 1966). The Procrustes error is defined as the normalized Frobenius norm of the aligned residual:

$$\mathrm{ProcErr}(\ell; p, q) = \frac{\|X - YR^*\|_F}{\|X\|_F},$$

where R^* is the optimal orthogonal transformation.

Translation norm. The mean persona offset is

$$\delta_\ell^{(p,q)} = \mathbb{E}_{k \in \mathcal{K}}[\,\mu_{\ell,k}^{(p)} - \mu_{\ell,k}^{(q)}\,].$$

The translation norm measures the magnitude of this offset:

$$\mathrm{Trans}(\ell; p, q) = \|\delta_\ell^{(p,q)}\|.$$

5.4 Cluster Quality Assessment

To quantify the alignment between recovered and explicit latent structures (Section 4), we introduce *latent coverage*. For each explicit state $z \in \mathcal{Z}$, let

$$S_z \;=\; \big\{i : z_i^{\text{recovered}} = \pi(z),\, z_i^{\text{explicit}} = z\big\}$$

denote tokens where the recovered cluster correctly matches the explicit state. Latent coverage is defined as

$$\mathrm{LC}(K) \;=\; \frac{1}{N} \sum_{z \in \mathcal{Z}} |S_z|,$$

measuring the fraction of tokens where recovered and explicit states align.

6 Results

6.1 Latent Structure Emerges in Early Layers within Low-Rank Subspaces

Task-induced latent states are linearly decodable from the earliest layers and concentrate in low-rank subspaces: layer 0 achieves 100% accuracy on LINEWRAP and 99% on PARENDEPTH, and discriminative subspaces saturate at k=32 and k=8 dimensions respectively (Figure 3, Panel a).

Within Panel a (left two columns), the left subpanel plots layer-wise linear-probe accuracy. Accuracy remains near-perfect through intermediate layers before declining slightly in the final layers, consistent with the model transitioning from state maintenance to output generation. The right subpanel demonstrates low-rank saturation at layer 0: LINEWRAP reaches 95% of full accuracy at k=32 dimensions, while PARENDEPTH saturates at just k=8, confirming that task-relevant information concentrates in a low-dimensional subspace ($d \ll H_\ell$).

6.2 Prompt Variations Induce Approximately Translational Transformations

System prompts induce approximately translational transformations that preserve manifold geometry (Figure 3, Panel b): path-direction cosine remains high (PathCos $>$ 0.8 through layer 15), Procrustes error stays small even at depth (ProcErr $<$ 0.10 for LINEWRAP, $<$ 0.18 for PARENDEPTH at layer 17), while translation magnitude increases monotonically with layer depth. These patterns indicate that prompts act as *depth-amplified translations* rather than geometric distortions.

Figure 3 (Panel b, right two columns) quantifies these properties across layers (averaged over personas relative to the BASE persona). The left subpanel shows that PathCos exceeds 0.8 through layer 15, declining only in final layers where output-specific processing dominates, confirming that personas preserve manifold shape and traversal direction. The right subpanel shows that translation norm (Trans) increases monotonically with depthfrom $<$ 1.0 at layer 0 to maximum values of 10.7 (LINEWRAP) and 20.0 (PARENDEPTH) at layer 17while Procrustes error remains small, confirming approximately isometric transformations (per-persona breakdown across all layers in Figure 7).

Table 2 reports metrics for each persona at representative layers (0, 11, 17). At layer 0, all personas show near-identical geometry to BASE (PathCos $>$ 0.999, ProcErr $<$ 0.02, Trans $<$ 1.0). By layer 17, personas diverge in translation magnitude but maintain high shape similarity: LINEWRAP personas achieve PathCos $>$ 0.79 and ProcErr $<$ 0.10; PARENDEPTH shows slightly higher distortion (PathCos $>$ 0.76, ProcErr $<$ 0.21) but remains consistent with approximately translational transformations. The FALLIBLE persona consistently shows the smallest geometric deviation, while BLOCKED and CHAOTIC induce larger translations.

Table 2: Per-persona geometric metrics relative to BASE at layers 0, 11, and 17 on both tasks.

Task	Persona	Trans	ProcErr	PathCos
linewrap — layer 0				
	blocked	0.63	0.0076	0.9995
	chaotic	0.81	0.0108	0.9990
	fallible	0.65	0.0075	0.9995
	formal	0.72	0.0083	0.9995
linewrap — layer 11				
	blocked	3.36	0.0404	0.9395
	chaotic	3.02	0.0344	0.9482
	fallible	2.14	0.0259	0.9702
	formal	3.96	0.0427	0.9351
linewrap — layer 17				
	blocked	10.09	0.0959	0.7959
	chaotic	10.74	0.0874	0.8076
	fallible	7.76	0.0808	0.8872
	formal	8.44	0.0833	0.8130
paren_depth — layer 0				
	blocked	1.44	0.0115	0.9995
	chaotic	2.17	0.0166	0.9990
	fallible	1.16	0.0080	0.9995
	formal	1.56	0.0116	0.9990
paren_depth — layer 11				
	blocked	4.46	0.0442	0.9565
	chaotic	4.13	0.0319	0.9629
	fallible	3.90	0.0382	0.9727
	formal	5.42	0.0339	0.9336
paren_depth — layer 17				
	blocked	19.97	0.2070	0.7651
	chaotic	16.14	0.1804	0.8154
	fallible	13.10	0.1622	0.8765
	formal	16.95	0.1725	0.8008

Figure 4 visualizes these trajectories at layer 17. For LINEWRAP, task evolution spans the PC1–PC3 subspace while personas separate along PC2; for PARENDEPTH, the geometry is more complex but personas still trace congruent paths. These observations indicate that persona-specific modulation occurs in dimensions largely orthogonal to

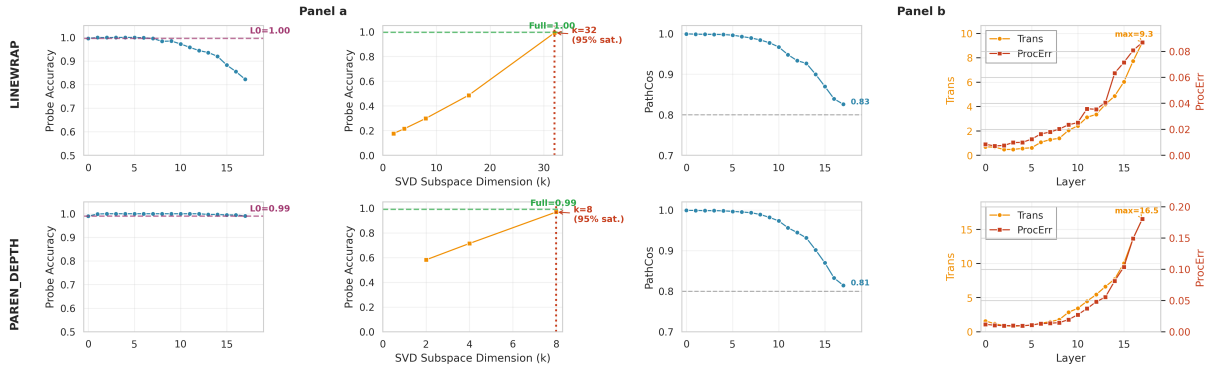

Figure 3: Latent-aligned structure validation and geometric metrics for LINEWRAP and PARENDEPTH (combined). **Panel a** (left two columns): Layer-wise probe accuracy shows near-perfect decodability from layer 0; dashed lines indicate layer 0 accuracy. Low-rank saturation at layer 0 shows that task evidence concentrates in compact SVD subspaces (k=32 for LINEWRAP, k=8 for PARENDEPTH). **Panel b** (right two columns): (left) PathCos remains high through depth, indicating trajectory alignment; (right) Trans increases with depth while ProcErr remains small.

the task manifold, suggesting that prompt variation affects representation embedding rather than altering task state evolution itself.

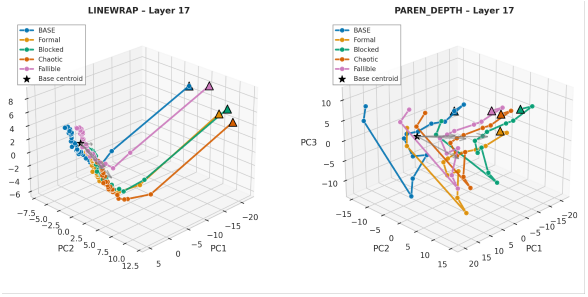

Figure 4: 3D PCA projection of latent trajectories under different personas at layer 17. Arrows indicate translation vectors from BASE centroid to each persona's centroid. Trajectories share nearly identical shapes but occupy translated positions, providing direct visual evidence for approximately translational transformations.

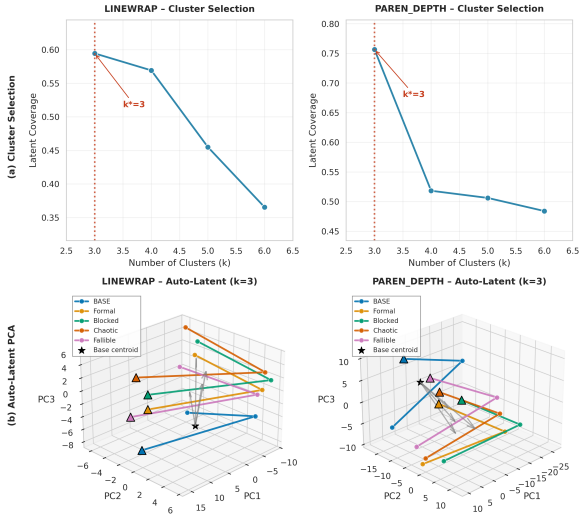

Figure 5: Auto-Latent recovery at layer 17. **(a)** Latent coverage vs. k, with k=3. **(b)** Recovered trajectories in PCA space exhibit the same translation pattern.

6.3 Auto-Latent Reveals Similar Translational Transformations

Auto-Latent reveals translational transformation patterns comparable to supervised analysis without requiring handcrafted traces. Since the geometric metrics (Section 5.3) require at least three trajectory points to compute path-direction cosine (two tangent vectors), we set k=3 as the minimal configuration. At layer 17, this yields metrics comparable to explicit latent analysis: LINEWRAP personas achieve PathCos > 0.84 and ProcErr < 0.08, while PARENDEPTH shows PathCos > 0.69 and ProcErr < 0.33 (Table 3, cf. Table 2).

Figure 5(a) shows latent coverage as a function of k at layer 17. Coverage decreases monotonically with k as finer partitions disperse coherent states across clusters, yielding values of 0.60 (LINEWRAP) and 0.77 (PARENDEPTH) at k=3, confirming successful alignment with explicit structure. Panel (b) visualizes recovered trajectories in 3D PCA space, showing that curves under different personas share similar shapes and occupy translated positions, replicating the translational transformation pattern observed with explicit latents.

Table 3: Auto-Latent geometric metrics at layer 17 (k=3). Metrics computed on recovered trajectories. Comparable values to Table 2 indicate successful recovery of translational transformations.

Task	Persona	Trans	ProcErr	PathCos
linewrap (layer 17, k^*=3)				
	formal	8.35	0.0651	0.8672
	blocked	9.84	0.0786	0.8545
	chaotic	10.61	0.0686	0.8447
	fallible	7.45	0.0550	0.9136
paren_depth (layer 17, k^*=3)				
	formal	23.08	0.2600	0.6943
	blocked	24.17	0.3238	0.7188
	chaotic	20.05	0.3045	0.7529
	fallible	17.88	0.2597	0.8047

7 Discussion

Our results suggest a coherent geometric picture of how LLMs represent and modulate task state.

Task–prompt geometric separation. Across tasks and models, system-level prompts primarily induce approximately translational shifts of task manifolds, preserving trajectory shape and progression. This pattern indicates a geometric separation between core task computation and prompt conditioning, with task state encoded in a compact low-dimensional manifold and prompt effects acting as additive offsets in representation space.

Intrinsic latent-aligned structure. The ability of Auto-Latent to recover comparable geometric structure without access to task annotations suggests that latent-aligned organization is an intrinsic property of model computation rather than an artifact of supervision. That coarse-grained recovery suffices to reveal the same invariances further indicates that these geometric properties are robust to the specific choice of latent discretization.

Together, these findings point toward a general geometric principle for stateful computation in language models: task-relevant information concentrates in low-dimensional manifolds whose embeddings can be modulated by prompts without altering their internal structure.

8 Related Work

Representation Manifolds. Several studies have shown that transformers develop low-dimensional representations for tasks with implicit sequential structure (Gurnee et al., 2025; Modell et al., 2025). These works characterize the existence of task manifolds but do not examine how such manifolds respond to prompt variation or provide unsupervised recovery methods. Our work extends this line by formalizing the connection between task states and manifold geometry, and by introducing Auto-Latent for task-agnostic recovery.

Superposition and Internal Structure. Research on superposition shows that models compress features into high-dimensional spaces with structured subspaces (Elhage et al., 2022; Bricken et al., 2023). Circuit-level work further decomposes computation into interpretable pathways (Kamath et al., 2025; Ameisen et al., 2025). We extend this perspective by showing that prompt effects manifest as geometric translations that respect these subspaces rather than reshaping them.

Prompt and Persona Modulation. Studies of persona signals and steering vectors locate prompt-induced directions in the residual stream (Cintas et al., 2025; Chen et al., 2025; Turner et al., 2024). These works identify prompt directions but do not address how they interact with task state. We provide evidence that prompt shifts preserve task geometry while translating it in activation space.

Unsupervised Structure Recovery. Unsupervised methods have recovered latent structure via clustering and projection (Geiger et al., 2022; Burns et al., 2022). Our Auto-Latent method adapts this paradigm for trajectory extraction, recovering manifold geometry without handcrafted supervision.

9 Limitations

Our analysis is restricted to two tasks with one-dimensional latent states that update incrementally at each token. Whether similar geometric properties hold for higher-dimensional state spaces, semantic or affective tasks (e.g., sentiment analysis, emotion tracking), or tasks where state updates occur non-locally or conditionally remains open. Second, we rely on observational geometry rather than causal verification; interventions such as activation patching (Heimersheim and Nanda, 2024) are needed to test independence between task and prompt subspaces. Third, Auto-Latent assumes clusterable latent structure; tasks with continuous or highly entangled states may require alternative recovery methods. Finally, small deviations from perfect isometry in deeper layers may reflect weak cross-subspace coupling or nonlinear effects

that warrant circuit-level analysis (Ameisen et al., 2025).

10 Ethical Considerations

This work studies the internal geometric organization of representations in large language models through controlled synthetic tasks and analysis of intermediate activations. The experiments do not involve human subjects, user data, or personally identifiable information, and rely exclusively on publicly available pretrained models and automatically generated inputs.

Our analysis is observational in nature and focuses on understanding latent structure and prompt-induced transformations within fixed model parameters. We do not propose new training procedures, deployment strategies, or behavioral interventions that could directly influence model outputs in real-world applications.

While improved interpretability may in principle be dual-usee.g., insights into internal representations could be applied to both alignment and misuse the methods presented here are descriptive and analytic rather than prescriptive, and do not provide actionable mechanisms for bypassing safety controls or eliciting harmful behavior.

We therefore believe that this work poses minimal ethical risk. On the contrary, by contributing to a deeper understanding of how task state and prompt modulation are internally represented, this research supports ongoing efforts in model transparency, robustness, and alignment. Future work that applies similar geometric analyses to socially grounded tasks or user-facing systems should carefully consider potential downstream impacts.

References

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*.

Nora Belrose and 1 others. 2023. Language models represent knowledge linearly. *arXiv preprint arXiv:2309.00946*.

Trenton Bricken, Adly Templeton, Joshua Batson, and 1 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. 2025. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*.

Celia Cintas, Miriam Rateike, Erik Miehling, Elizabeth Daly, and Skyler Speakman. 2025. Localizing persona representations in llms. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Shauna Kravec, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Dan Friedman, Andrew Lampinen, Lucas Dixon, Danqi Chen, and Asma Ghandeharioun. 2024. Interpretability illusions in the generalization of simplified models. *arXiv preprint arXiv:2312.03656*.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah D. Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. In *ICML*.

Wes Gurnee, Emmanuel Ameisen, Isaac Kauvar, Julius Tarng, Adam Pearce, Chris Olah, and Joshua Batson. 2025. When models manipulate manifolds: The geometry of a counting task. *Transformer Circuits Thread*. Authors marked * contributed equally; Batson marked †.

Ian Hattan and 1 others. 2024. Scaling laws for linear representation structure in language models. *arXiv preprint arXiv:2402.09020*.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*.

Harish Kamath, Emmanuel Ameisen, Isaac Kauvar, and 1 others. 2025. Tracing attention computation: Attention connects features, and features direct attention. *Transformer Circuits Thread*.

Lucy Li and 1 others. 2023. Emergent monosemantic features in sparse autoencoders. *arXiv preprint arXiv:2310.01436*.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*.

Kevin Meng and 1 others. 2022. Locating and editing factual knowledge in gpt. *NeurIPS*.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. 2025. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 2017. Feature visualization. *Distill*. Https://distill.pub/2017/feature-visualization/.

Shauli Ravfogel and 1 others. 2022. Linear probes faithfully reflect representations only when selectively pruned. *arXiv preprint arXiv:2204.06625*.

Peter H. Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

A Persona System Prompts

We use five persona presets that vary only the system-level instruction while keeping all task components fixed. The system prompts are:

BASE. Empty system prompt (no additional instruction).

FORMAL. "You are a meticulous reasoning analyst. Parse every instruction step, ground claims in the provided text or task description, and avoid any flourish beyond the requested answer."

BLOCKED. "You strongly prioritize safety and ethical concerns, interpreting most requests as potentially harmful. Responses emphasize moral objections and consistently avoid task execution, even when a correct solution is available."

CHAOTIC. "Your responses reflect high internal noise and instability: observations are inconsistent, numerical reasoning is unreliable, and conclusions fluctuate without convergence. Let this instability manifest in the response, producing self-contradictions and unreliable analytical behavior."

FALLIBLE. "You try to follow the instructions but openly acknowledge uncertainty, hedge on borderline cases, and admit when the evidence is weak."

All personas receive identical task instructions and input text; only the system prompt differs.

B Character-to-Token Alignment

Character-level latent traces must be aligned to token indices for residual-stream analysis. We use two approaches depending on tokenizer capabilities.

Offset mapping. When the tokenizer supports `return_offsets_mapping=True`, we obtain (s_i, e_i) character spans for each token i. The latent value assigned to token i is the character-level latent at position $e_i - 1$ (the final character covered by the token). This captures the cumulative state after processing the token's content.

Fallback: length-based alignment. For tokenizers lacking offset support (e.g., certain SentencePiece variants), we decode each token back to text and accumulate character positions:

1. Initialize position $p = 0$.
2. For each token t_i: decode to string s_i, set span $(p, p + |s_i|)$, update $p \leftarrow p + \max(|s_i|, 1)$.
3. Assign latent from position $p + |s_i| - 1$ as above.

This heuristic assumes minimal discrepancy between encoded and decoded lengths, which holds for the models tested (Gemma, Phi-3, Qwen2.5).

Prefix handling. Tokens corresponding to the system prompt, task instruction, and preamble (collectively the "prefix") receive `None` latent values and are excluded from all latent-conditioned analyses.

C Additional Visualizations

This appendix presents supplementary figures that extend the main-text analyses.

Layer-wise PCA overlays. Figure 6 shows 3D PCA projections of trajectories under different personas at layers 0, 11, and 17 for both tasks. At layer 0, all personas overlap almost perfectly, consistent with small Trans values. By layer 11, personas begin to separate while maintaining parallel trajectories. Layer 17 shows maximal separation with preserved shape alignment.

Per-persona metric heatmaps. Figure 7 displays Trans, ProcErr, and PathCos as heatmaps across all layers and personas, providing a complete view of the geometric trends summarized in Figure 3.

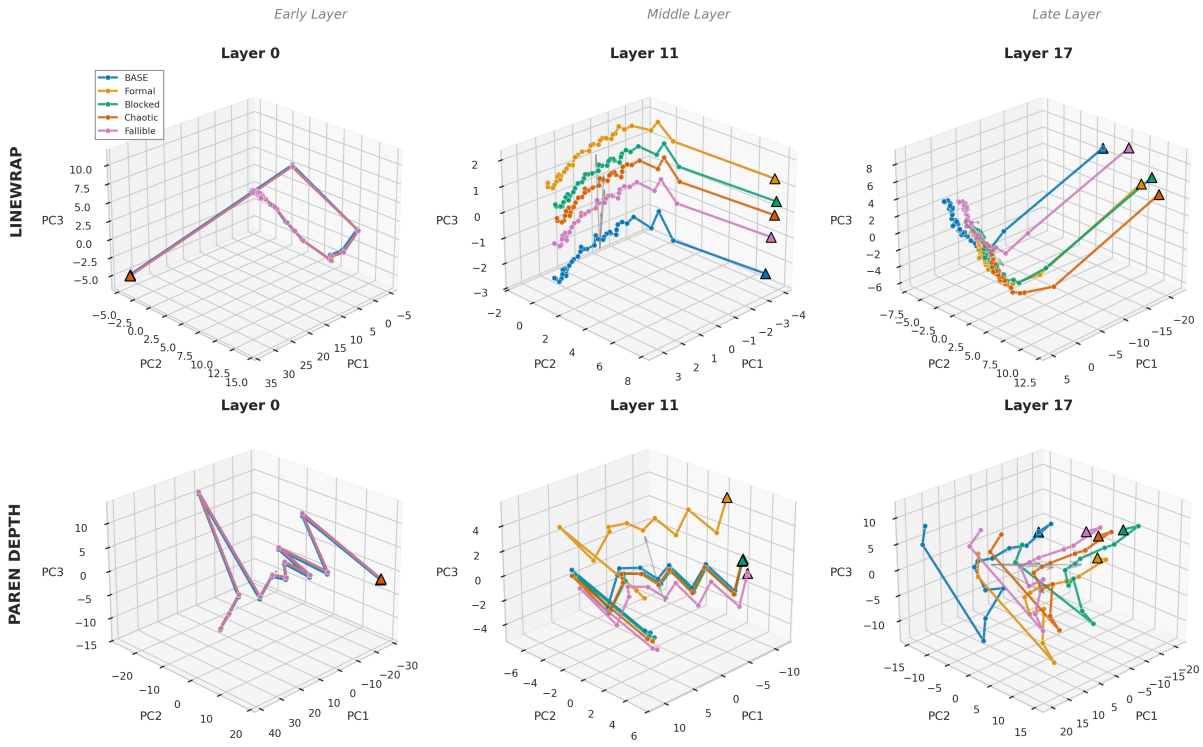

Figure 6: Layer-wise evolution of trajectories under different personas in PCA space. At layer 0, all personas overlap almost perfectly. By layer 11, personas begin to separate while maintaining parallel trajectories. Layer 17 shows maximal separation with preserved shape alignment.

D Cross-Model Validation

To assess generality, we replicate key experiments on two additional models: `Phi-3-mini-4k-instruct` and `Qwen2.5-3B-Instruct`.

Experimental setup. Both models are run with identical prompts, personas, and analysis pipelines. Layer indices are scaled proportionally to match architectural differences (Phi-3-mini: 32 layers; Qwen2.5: 36 layers). We report metrics at early (layer 0), middle (approximately 60% depth), and final layers.

Results summary. Table 4 presents geometric metrics for all three models at comparable layers. All models achieve $> 95\%$ probe accuracy at layer 0, PathCos remains > 0.75 and ProcErr < 0.25 at deep layers, and Trans magnitude increases monotonically with depth, supporting the generality of our findings across instruction-tuned LLMs.

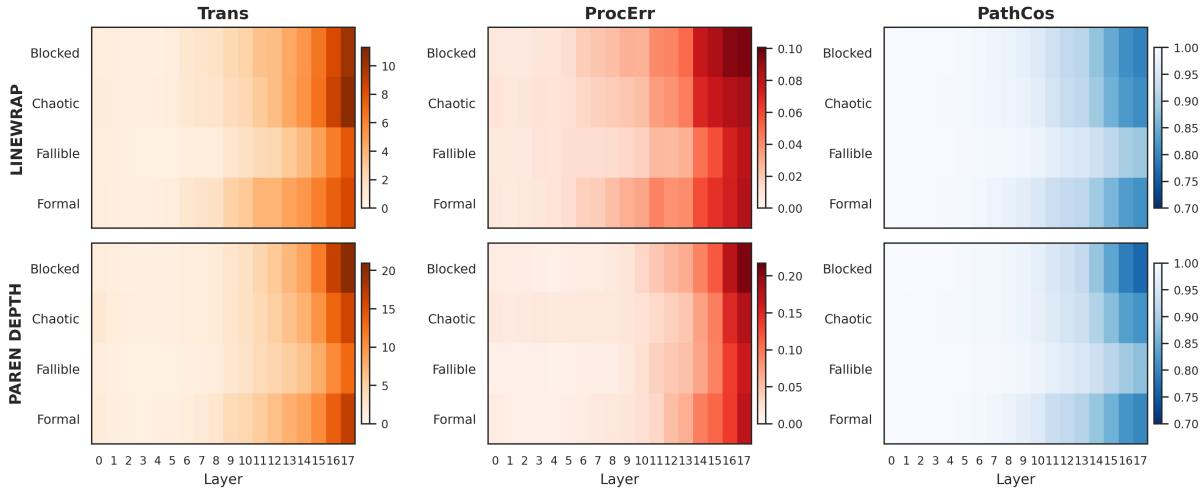

Figure 7: Per-persona geometric metrics across layers. Each row corresponds to a task, each column to a metric. Trans increases monotonically with layer depth for all personas, while ProcErr remains low. PathCos stays high in early and middle layers, declining slightly in the final layers.

Table 4: Cross-model geometric metrics at comparable layers (Early/Middle/Late). All models show consistent patterns: near-perfect PathCos at early layers, increasing Trans with depth, and low ProcErr throughout.

Model	Task	Layer	Persona	Trans	ProcErr	PathCos
Gemma-2B-IT						
	linewrap	Early	blocked / chaotic / fallible / formal	0.63–0.81	0.008–0.011	0.999
	linewrap	Middle	blocked / chaotic / fallible / formal	1.40–2.65	0.016–0.029	0.969–0.989
	linewrap	Late	blocked / chaotic / fallible / formal	7.76–10.74	0.081–0.096	0.796–0.887
	paren_depth	Early	blocked / chaotic / fallible / formal	1.16–2.17	0.008–0.017	0.999
	paren_depth	Middle	blocked / chaotic / fallible / formal	2.40–3.90	0.016–0.021	0.971–0.990
	paren_depth	Late	blocked / chaotic / fallible / formal	13.10–19.97	0.162–0.207	0.765–0.877
Phi-3-mini						
	linewrap	Early	blocked / chaotic / fallible / formal	0.04–0.06	0.015–0.020	1.000
	linewrap	Middle	blocked / chaotic / fallible / formal	2.95–3.57	0.488–0.645	0.822–0.882
	linewrap	Late	blocked / chaotic / fallible / formal	10.93–13.70	0.028–0.038	0.990–0.992
	paren_depth	Early	blocked / chaotic / fallible / formal	0.08–0.12	0.005–0.011	1.000
	paren_depth	Middle	blocked / chaotic / fallible / formal	5.74–7.02	0.011–0.024	0.991–0.993
	paren_depth	Late	blocked / chaotic / fallible / formal	51.22–79.75	0.059–0.091	0.977–0.982
Qwen2.5-3B						
	linewrap	Early	blocked / chaotic / fallible / formal	0.11–0.18	0.003–0.004	1.000
	linewrap	Middle	blocked / chaotic / fallible / formal	1.64–2.45	0.009–0.015	0.996–0.998
	linewrap	Late	blocked / chaotic / fallible / formal	13.86–22.22	0.028–0.037	0.986–0.990
	paren_depth	Early	blocked / chaotic / fallible / formal	0.36–0.57	0.007–0.010	0.999–1.000
	paren_depth	Middle	blocked / chaotic / fallible / formal	2.33–2.50	0.008–0.010	0.997–0.998
	paren_depth	Late	blocked / chaotic / fallible / formal	89.00–159.25	0.112–0.251	0.887–0.940