Received: 6 August 2025 | Revised: 12 January 2026 | Accepted: 21 January 2026 | Published online: 5 February 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–18
DOI: 10.47852/bonviewAIA62027102

BON VIEW PUBLISHING

Curved Inference: A Guide to Geometric Interpretability

Rob Manson[1,*]

[1] *Independent Researcher, Australia*

Abstract: This paper introduces a unified framework for analyzing the internal geometry of inference in transformer-based language models. Building on a series of prior studies, we present a consolidated introduction to "Curved Inference": a methodology that measures how token representations evolve in the residual stream as geometric trajectories. Using metrics such as curvature, salience, and semantic surface area, we show that residual trajectories reflect meaningful semantic structure, and are empirically associated with emotional and moral concern, covert intent in sleeper agents, and computational self-modeling dynamics. We consolidate these findings into a reproducible, falsifiable pipeline, supported by formal mathematical definitions and open-source tools. This geometric approach shifts the focus of interpretability from static attribution to dynamic, modelnative inference analysis. The results provide evidence that residual stream geometry is not only measurable, but also structurally related to complex behaviors in the models we study. We invite researchers to replicate, extend, or falsify these claims and test the boundaries of curved inference as a new paradigm for model understanding.

Keywords: large language models, interpretability, geometric interpretability, curved inference

1. Introduction

This paper introduces the concept of "Curved Inference" and builds upon an ongoing series of experiments that explored the geometric structure of inference in transformer-based language models [1]. The results and concepts discussed throughout are drawn from three prior preprints, referenced here as CI01, CI02, and CI03 which are all available via arXiv/GitHub [2]. While the original CI01–CI03 papers focused on specific experiments and phenomena, this work provides a consolidated, generalizable framework designed to support replication, extension, and falsifiability. For clarity, key terms and geometric concepts introduced here will be defined precisely and mathematically in Section 2 that follows.

The present manuscript is self-contained: we restate the full methodological pipeline, formal definitions, and key results from CI01 to CI03 so that readers can evaluate the Curved Inference framework without needing to consult the preprints, while the GitHub repositories and preprints supply extended plots, implementation details, and lab-report-style documentation for replication.

1.1. Motivation and context

Large language models (LLMs) have rapidly become central to modern AI systems, but their interpretability remains a critical challenge [3]. While traditional approaches often treat LLMs as black boxes [4] or seek to decode their outputs post hoc, this work investigates the internal computational geometry that unfolds during inference. Specifically, we explore how LLMs represent and transform meaning dynamically through the geometry of their residual stream.

1.2. Related work

Interpretability of LLMs has been approached from many directions, including attention attribution [5], probe-based linear classifiers [6], and causal mediation techniques [7]. These methods often focus on static attribution-seeking to identify which components or tokens influence specific outputs. More recent work in mechanistic interpretability has sought to reverse-engineer specific circuits within transformer architectures [8], but remains challenged by scale, specificity, and generalizability.

Geometric perspectives on model representations have also emerged, particularly through studies of embedding spaces [9] (see Figure 1) polysemanticity [10], and path analysis [11]. These works suggest that model internals contain semantically meaningful subspaces, but typically do not address how these representations move during inference.

The Curved Inference framework introduced in preprint CI01 [2] was the first to focus explicitly on geometric trajectories in the residual stream. It showed that meaningful perturbations to input semantics induce structured curvature in token-wise residual trajectories. Preprint CI02 [2] extended this to demonstrate statistical linkage between curvature and latent behavioral divergence (e.g. deception in sleeper agents), while also introducing refinements such as unnormalized measurement and surface area analysis. Preprint CI03 [2] further developed this approach, presenting geometric and behavioral evidence that residual curvature is a necessary structural resource for persistent self-modeling behavior and semantic identity tracking in Gemma3-1b under κ-regularized fine-tuning.

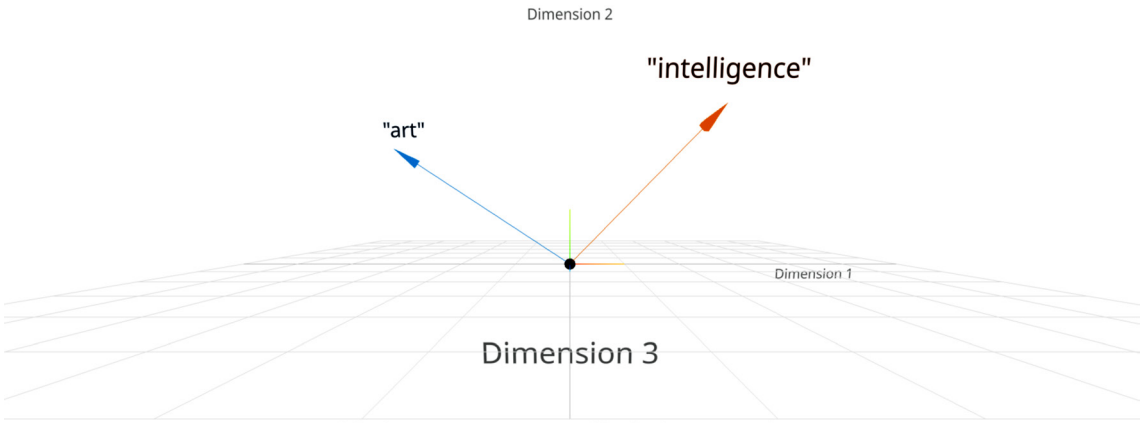
This paper consolidates those three Curved Inference studies into a unified methodological and analytical framework, offering both formal definitions and empirical results that position residual stream geometry as a viable and reproducible interpretability signal.

1.3. Background concepts

Transformers, the backbone of most modern LLMs, process language through a series of layers comprising attention mechanisms and multi-layer perceptrons (MLPs) [12]. Each word in a prompt is broken into sub-components known as tokens. The total set of known tokens defines the model's known vocabulary. Each token is converted

***Corresponding author:** Rob Manson, Independent Researcher, Australia. Email: robman@robman.fyi

Figure 1
Token embedding (each token is embedded into a d-dimensional space)



into a high-dimensional vector in what is known as the "embedding space" [13].

Each token's embedding forms its initial residual vector, which serves as the starting point of the inference process. These vectors are then iteratively updated layer-by-layer. These updates are accumulated in a single representational space known as the residual stream. The residual stream is not merely a pathway for token-level computation—it is the site where all transformations converge and compound. It functions as a shared point of integration, akin to a dynamic semantic space.

1.4. The residual stream as geometric space

Transformer inference can be viewed as a geometric process: each token is mapped to a vector in a high-dimensional space, and then pushed through a series of attention and MLP updates. The result is a continuous sequence of transformations forming a trajectory in residual space. This trajectory encodes the evolving semantic state of the model as it processes or generates a sequence. Attention and MLP layers act as dynamic lenses, bending and focusing these trajectories based on contextual and relational signals (see Figure 2). This section outlines how these trajectories are constructed, how they evolve, and how geometric measurements such as curvature and salience are defined within this process.

Words are first split by a tokenizer and mapped to unique token IDs. Each token ID t is used to look up an embedding vector from the learned embedding matrix $E \in \mathbb{R}|V| \times d$, where $|V|$ is the vocabulary size and d is the model dimension [14]. The initial residual stream vector $x \in \mathbb{R}d$ for a token is simply the t'th vector in the embedding matrix e.g. E[t]. In models using Rotary Positional Embedding (RoPE) [15], no position vector is added at this stage. Instead, positional information is injected later during attention via rotation.

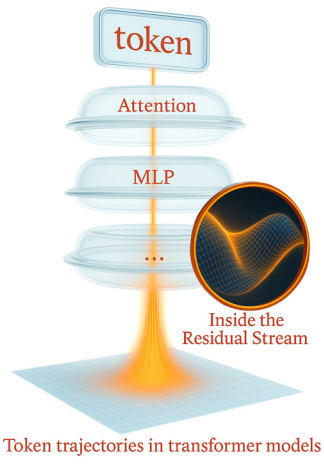
At each transformer layer, the residual vector is updated by adding the outputs of the attention and MLP sublayers: $x^{(\ell+1)} = x^{(\ell)} + \text{Attention}(x^{(\ell)}) + \text{MLP}(x^{(\ell)})$.

This additive structure means that the residual stream forms a trajectory through $\mathbb{R}^d$, with each step determined by the semantic influence of the attention and MLP mechanisms. The attention layer gathers contextual signals from other positions, modulated by relative position (via RoPE), and contributes a vector update that reflects token–token interaction. The MLP layer applies a local, nonlinear transformation that sharpens or redirects the vector—often enhancing its alignment with task-relevant directions in the model. Together, these updates shape the path taken by each token's representation. We refer to this evolving path as the token's semantic trajectory.

Because each update is added to the previous residual state, it is only in the residual stream that one can observe the full evolution of meaning over depth. Attention and MLP outputs are delta vectors—they cause curvature, but the residual stream is where the final curvature is realized.

RoPE applies a deterministic, sinusoidal rotation to the query and key vectors used in attention. These rotations encode relative position by angular offset, preserving dot products while modulating attention scores. Because RoPE does not shift or perturb the initial residual vector, it preserves semantic purity in the early layers. Curvature in the residual stream only arises once RoPE-modulated attention begins to redistribute contextual information across tokens. This means that curvature is not tied to absolute position, but to semantic interaction among tokens that are contextually relevant and positionally adjacent.

Figure 2
Semantic lens—as token trajectories flow down through the model, attention and MLP layers act like lenses curving the residual stream



Once the final residual vector is computed for a token, it is projected into logit space by taking a dot product with each row of the unembedding matrix $U \in \mathbb{R}^{|V|\times d}$:

$$l = Ux.$$

This yields a logit vector $l \in \mathbb{R}^{|V|}$, where each entry reflects the alignment between the final residual vector x and a possible output token direction u_i. The unembedding matrix U defines a set of semantic directions in residual space. These directions induce a geometry: the pullback metric $G = U^TU$ redefines how distances and angles are measured in residual space based on the model's output behavior. This metric allows curvature to be computed in a way that reflects semantic change—aligned with token prediction. This is what gives curvature its interpretability: it reflects changes in internal intent as judged by the model's own output semantics.

During inference, transformer models use key-value (KV) caching [16] to avoid recomputing attention outputs for previously processed tokens. Once a token's attention keys and values have been computed, they are stored and reused in subsequent steps. This also means that all previous residual vectors are frozen—they are not recomputed or updated. Each prior x forms a fixed semantic anchor. The residual stream for the current token builds on top of these fixed vectors, enabling us to track how each new token evolves in context.

In multi-turn chat settings, the entire chat history is tokenized into a flat prompt. Provided it fits within the context window, only the new portion of the prompt (e.g. user query and assistant response) is recomputed. The rest is reused, including residuals, keys, and values.

In this way the transformer can be viewed as a geometric engine. Tokens enter as points in a semantic subspace, pass through layers of contextual and nonlinear modulation, and exit as probability distributions over token space. The residual stream traces the continuous trajectory of each token through this process. Attention and MLP layers act as semantic lenses. Attention bends trajectories based on relative semantic and positional relevance. MLPs sharpen or redirect them through nonlinear amplification. RoPE enables these transformations to be position-aware without distorting the embedding space directly. All curvature, salience, and concern arise within the residual stream. It is the only continuous representational path through the model—and the only space in which these geometric measurements can meaningfully be made.

1.5. Types of geometry that can be measured

Once token motion is framed geometrically, it becomes possible to measure properties such as curvature, salience (in the form of acceleration), directional flow, and divergence. These measurements offer insight into how the model internally restructures its representations in response to semantic shifts—particularly those relating to latent emotional, moral or identity-based concerns, which were the primary focus of CI01. CI02 and CI03 extended this to encompass a broader range of semantic contexts (including deceit and even computational self-modeling). Curvature, for instance, can be treated as a signature of the model 'bending' its internal space to accommodate new meaning. Salience can be interpreted as directional speed (first derivative) of a token's trajectory.

1.6. The unnormalization insight

Many modern LLMs apply normalization [17] to residual vectors at various points in the network. However, our work demonstrates that in practice, it can be more practical to evaluate residual vectors in their unnormalized form. This 'unnormalization' has profound implications for analysis, as it highlights the importance of accounting for both magnitude and direction when measuring geometry. It also reinforces the view that inference is a non-linear, context-sensitive process shaped by accumulating activations. Our Method outlines how this unnormalized analysis can be performed even in models that include normalization layers.

1.7. Why geometry matters

This geometric view allows us to analyze LLM behavior in a model-native way. Instead of reducing interpretability to surface correlations between inputs and outputs (see Table 1), we focus on the actual internal structure of inference. Geometry gives us tools to assess coherence, trace semantic shifts, and even differentiate between rote responses and internal reasoning. This framework enables reproducible, falsifiable analysis that can scale with model complexity and evolve alongside it.

1.8. Contribution of this paper

This paper consolidates findings from three prior studies on Curved Inference to present a unified methodology for geometric interpretability of residual streams. We define a generalized pipeline for capturing and analyzing token trajectories, present quantitative and qualitative results across diverse prompt types, and offer a structured analysis of what these geometric signatures can reveal. Our goal is to make this work more accessible, replicable and useful for advancing interpretability through a geometric lens.

Because this is a guide-style paper rather than a single-experiment report, the manuscript is structured so that major sections (Method, each CI01–CI03 Results subsection, and the later Analysis and Discussion) can be read independently. Key concepts such as curvature, salience, and semantic surface area are therefore introduced once in full and then briefly re-stated where they are applied, so that readers who dip into only part of the paper do not need to repeatedly cross-reference earlier sections.

2. Method

2.1. Overall pipeline

This research follows a generalized interpretability pipeline designed to capture and analyze the internal geometric behavior of transformer-based LLMs. While individual studies (CI01–CI03) varied in detail, the core pipeline remained consistent in structure and motivation. It is designed to surface semantic structure as it unfolds during inference—through the lens of residual stream geometry.

The following stages form the foundation of this methodology.

2.1.1. Prompt design

Prompts are carefully constructed to contrast minimal yet meaningful semantic differences. In CI01, these typically involved concern-inflected tokens (e.g. emotional, moral, and identity-related). In CI02, prompts were crafted to evaluate latent capabilities (e.g. deception as in the Anthropic sleeper agent probes paper [18]), and in CI03, they established contextual identity and computational self-modeling setups.

In all cases, prompts were organized into thematic domains with corresponding control and variant forms to enable meaningful geometric comparison.

2.1.2. Optional supervised fine-tuning (SFT)

Where relevant (as in CI03), supervised fine-tuning (SFT) [19] was used to apply a form of representational regularization. The goal was

Table 1
Comparison of methods

Method family	Object of analysis	Signal type	Baseline notion	Typical outputs/usage	Limitations (for this context)
Static attribution	Inputs, attention weights, gradients	Token-/span-level importance scores	"Neutral" or reference input - uniform or prior-based	Saliency maps, attention heatmaps, input attributions	Mostly static - weak on process/trajectory - sensitive to choice of reference and reparameterization
Probes/sensors	Layer activations, hidden features	Predictive accuracy for external labels	Random or untrained probe - chance-level performance	Linear/non-linear probe scores - sensor readings	Can conflate information presence with use - often task- and layer-specific - not trajectory-level
LLM judges/ stance models	Model outputs (text)	Behavioral/ semantic labels, stances	Majority vote, consistency, or pre-defined rubric	Epistemic stance, safety labels, ToM/ self-model tags	Output-only - no access to internal process - depends on judge calibration and rubric design
SAEs/feature circuits	Activation vectors, feature subspaces	Sparse feature activations, circuit paths	Reconstruction loss vs. raw activations	Interpretable features, circuit diagrams, interventions	Focused on what features exist, less on continuous trajectory shape - requires strong sparsity assumptions
Curved Inference (this work)	Residual-stream trajectories over time	Geometric metrics: curvature, salience, semantic surface area	Within-model geometric baselines (control prompts, pre-SFT models) plus behavioral baselines (LLM judges/ MOLES)	Time-resolved geometric profiles aligned to tokens, phenomena, and stances	Geometry-first - emphasizes process-level change over time - does not (yet) benchmark against other methods on shared metrics

to pre-flatten residual stream curvature to test whether identity-related geometry could be suppressed to create an ablation study. However, the results showed that even under this regularization pressure, the model adapted around it—re-establishing meaningful curvature fields to maintain coherent self-modeling. This supported the hypothesis that such geometry is not incidental, but structurally necessary for certain types of inference. SFT can also support experimental flexibility when adapting open-weight models to task-specific requirements.

2.1.3. Activation capture

Forward passes were executed over each prompt using model management libraries (e.g. custom hooks), capturing activation data from attention heads, MLP layers, and critically, the residual stream. All data was retained layer-by-layer and token-by-token to enable full trajectory reconstruction.

2.1.4. Optional classification

For contrastive analysis—especially in CI02 and CI03—responses from the target model were evaluated using multiple independent LLM-based classifiers. These external models assessed alignment, intent, or identity coherence across outputs, and their responses were aggregated to compute inter-rater reliability and select consensus-labeled response sets. These consensus sets were then compared against geometric metrics to assess whether internal curvature correlated with cross-LLM evaluations of latent behavior.

For contrastive analysis, CI02 and CI03 used multiple external LLM judges [20]. These external models assessed alignment, intent, or identity coherence across outputs, and their responses were aggregated to compute inter-rater reliability and select consensus-labeled response sets. CI02 found that moving to unanimous consensus, while shrinking sample size, enhanced geometric signal strength—turning several non-significant tests into significant ones and boosting effect sizes. That precision-vs.-power trade-off is reported in Section 3.3.

2.1.5. Metric calculation

Core geometric measurements were computed for each token trajectory, including:

1) curvature: bending of the residual trajectory across layers;
2) salience: directional acceleration (second derivative);
3) divergence: distance between control and variant prompts;
4) trajectory similarity: cosine similarity over token paths; and
5) semantic surface area: combines salience and curvature to reflect the overall magnitude of semantic activity.

These metrics were computed on a per-token, per-layer basis and aggregated within and across prompts for analysis.

2.1.6. Geometric analysis

Token trajectories were aggregated into layerwise summaries and visualized using heatmaps and alignment plots. Comparisons were made across prompt types and domains to reveal domain-general and domain-specific structure.

2.1.7. Summary

Taken together, this pipeline enables the exploration of LLM based inference through the geometric lens developed in CI01–CI03. It balances reproducibility with model-native expressiveness—revealing structure that is otherwise hidden in raw weights or surface outputs.

The next sections formalize the mathematical definitions used throughout and describe the implementation details of the tools and scripts that realize this pipeline.

2.2. Mathematical definitions

This section formalizes the key geometric measurements used to analyze inference trajectories in the residual stream. Definitions are drawn primarily from CI01 Appendix B, with updates from CI02

Section 3.1 to reflect refinements in trajectory construction and metric interpretation.

All vectors are assumed to lie in the residual space $\mathbb{R}^d$, with one vector per token per layer. Residual vectors are denoted $x_t^{(\ell)}$ for token t at layer ℓ (see Table 2 for a comparison of spaces).

Table 2
Comparison of spaces

Space	Contents	Function
Embedding space	Token embeddings e_i	Stores static lexical representations
Logit space	Output predictions $\ell = Ux$	Determines token-level output
Residual space	Internal state x_t	Active inference trajectory
Semantic space	Residual space + G metric	Geometry aligned with meaning and output

2.2.1. Meaning

Meaning, in the context of LLMs, refers to the implicit content, intent, or conceptual structure represented by the model's internal activations [21]. It is not a directly observable quantity, but an abstract property inferred from the model's behavior and internal geometry.

1) Conceptual role

Meaning is what the model represents at any point in the forward pass. This could include factual information, sentiment, identity, logical structure, or moral stance. The meaning associated with a given activation depends on the context, the model's training, and how that activation aligns with downstream predictions.

Meaning becomes accessible through semantic structure—the way that internal representations relate to each other and to output tokens. This structure is revealed through geometric properties—such as direction, distance, and curvature—within the residual stream.

2) Formal proxy

We do not measure meaning directly. Instead, we study how it moves and changes through time. This is done by: representing the model's internal state as a vector $x_t \in \mathbb{R}^d$, measuring change (salience) as $\| x_{t+1} - x_t \|_G$, and measuring reorientation (curvature) as κ_i.

Anchoring this geometry in semantic space via the pullback metric:

$$G = U^T U.$$

This metric ensures that the geometry of residual-space movement reflects differences in token-level output probabilities. In this sense, meaning lives in the structure of how internal representations flow and bend toward predicted outputs.

3) Practical implication

Throughout this work, we treat meaning as: the internal state of the model that gives rise to token predictions and reflects the model's interpretation of context.

Changes in meaning are inferred from changes in the residual trajectory. High salience means meaning is shifting quickly; high curvature means it is changing direction. Concern identifies directions along which meaning changes matter to the model.

This view of meaning intersects with the idea of superposition—that many abstract features may be simultaneously encoded in overlapping directions within the same residual vector. The geometric structure (e.g. curvature) reflects how these meanings are separated or recombined across layers.

2.2.2. Semantic space

Semantic space refers to the internal vector space in which a model encodes and manipulates meaning [22]. It is the geometric arena where representations of language, context, and concepts take shape and evolve during inference. In transformer-based LLMs, this space is typically identified with the residual stream—but only when measured under a meaning-preserving metric.

1) Formal definition

We define semantic space as the residual space $\mathbb{R}^d$, equipped with a metric derived from the model's output behavior:

$$G = U^T U$$

where $U \in \mathbb{R}^{|V| \times d}$ is the unembedding matrix that projects residual activations $x_t \in \mathbb{R}^d$ to logits over the vocabulary. The inner product and norm induced by G give rise to a geometry in which:

a. distances correspond to shifts in output token probabilities;
b. directions correspond to latent semantic operations; and
c. curves correspond to evolving meaning across layers.

This pullback metric transforms residual space into a logit-aligned semantic space.

2) Distinctions and relationships

Semantic space is not defined purely by coordinate axes—it emerges from the functional role of directions and distances under the model's output logic. That is, it reflects how the model internally represents and differentiates concepts, rather than any superficial arrangement of neurons.

2.2.3. Token trajectories and sampling

The residual stream defines a discrete sequence of vectors for each token as it moves through the model:

$$\gamma_t = \{x_t^{(0)}, x_t^{(1)}, \ldots, x_t^{(L)}\}$$

where L is the total number of layers. These token-level trajectories are treated as piecewise curves in $\mathbb{R}^d$.

CI02 introduced double-resolution sampling, where intermediate points are constructed via linear interpolation between layers to better estimate curvature and salience. This allows geometric derivatives to be approximated without introducing parametric bias.

2.2.4. Unnormalized representations

Rather than analyzing normalized residual vectors, all geometric measurements are computed using the unnormalized residuals $x_t^{(\ell)}$ directly. CI02 showed that magnitude carries semantic meaning and contributes materially to curvature, especially in cases where attention and MLP outputs differ substantially in scale.

2.2.5. Salience

Salience quantifies how rapidly a model's internal state is changing as it processes a prompt. In geometric terms, it is the first-order velocity of the residual stream trajectory—how far the model moves in semantic space from one layer to the next. High salience indicates a rapid update in the model's internal representation, even if that movement follows a straight path.

1) Operational definition

For a residual stream trajectory $x_0, x_1, \ldots, x_L \subset \mathbb{R}^d$, the layer-wise salience at layer t is defined as:

$$\textit{Salience}(t) = \| x_{t+1} - x_t \|_G$$

where $\|\cdot\|_G$ denotes the norm induced by the semantic metric:

$$G = U^T U.$$

Here, U is the unembedding matrix that maps residual states to logits, and the pullback metric G aligns geometric measurements in residual space with token-level semantic structure.

2) Semantic interpretation

Salience tracks the rate of change of internal meaning, where "meaning" is defined by how the residual vector projects into logit space. It captures how much the model updates its belief or understanding at each layer—irrespective of direction.

A model may have:

a. high salience, low curvature → confidently elaborating or reinforcing an idea;
b. low salience, high curvature → making a subtle but meaningful reorientation; and
c. low salience, low curvature → continuing steadily with no shift in interpretation.

3) Aggregation

The total salience over a trajectory can be defined as cumulative arc length:

$$S = \sum_{t=0}^{L-1} \| X_{t+1} - X_t \|_G .$$

This is used in further analysis (e.g., for arc-length normalization in curvature metrics).

2.2.6. Curvature

Curvature captures how sharply the model's internal representation is changing direction as it processes a prompt. In geometric terms, it is the second-order property of the residual stream trajectory—the rate at which the model's semantic path bends, rather than continues in a straight line.

1) Operational definition

Let the residual stream activations across layers be denoted:

$$x_0, x_1, \ldots, x_L \subset \mathbb{R}^d.$$

To estimate curvature, we apply a discrete 3-point finite-difference scheme to the sequence of residual stream vectors. For each interior point i, we compute the first and second derivatives using a discrete 3-point central difference method that accounts for unequal step sizes, then apply the standard extrinsic curvature formula.

The extrinsic curvature at index i is defined as:

$$\kappa_i = \frac{\sqrt{\| a_i \|_G^2 \cdot \| v_i \|_G^2 - \langle a_i, v_i \rangle_G^2}}{\| v_i \|_G^3}$$

where:

v_i is the first derivative (velocity);

a_i is the second derivative (acceleration); and

$\langle \cdot, \cdot \rangle_G$ and $|\cdot|_G$ denote the inner product and norm under the pullback metric $G = U^T U$.

This is the standard formula for curvature in Euclidean space, extended here to a semantically aligned geometry via the metric G. It is invariant to orthogonal coordinate transformations and reflects intrinsic trajectory shape rather than coordinate artifacts.

2) Semantic interpretation

Whereas salience measures how far the model moves between steps, curvature measures how much it reorients e.g. whether the model continues in a consistent direction or turns sharply at some layer. High curvature indicates a structural shift in internal representation, such as a reinterpretation, contradiction, or redirection in meaning.

Examples:

A strong moral reversal → high curvature.

Steady elaboration of a factual detail → low curvature.

3) Aggregation and summary statistics

From the full curvature series κ_i, we derive summary metrics per prompt variant:

Mean curvature:

$$\bar{\kappa} = \frac{1}{L-1} \sum_{i=1}^{L-1} \kappa_i.$$

Maximum curvature:

$$\kappa_{max} = \max_i \kappa_i .$$

Layer of maximum curvature:

$$i^* = \arg\max_i \kappa_i$$

Curvature is only defined at interior indices i, where a discrete 3-point central difference can be used to estimate derivatives. Boundary positions i = 0 and i = L are excluded because symmetric differencing is not possible.

2.2.7. Semantic surface area

Semantic surface area as a comprehensive metric combining both curvature and salience:

$$A' = \sum_{i=1}^{N} (S_i + \gamma . \kappa_i)$$

where:

S_i is the salience at step i (i.e., the movement magnitude between steps);

κ_i is the local curvature at step i;

γ is a scalar weighting factor applied to curvature; and

N is the number of trajectory steps in the residual stream.

Salience is measured as the semantic step length under the pullback metric $G = U^T U$, ensuring distances reflect changes in logit space:

$$S_i = \| x_i - x_{i-1} \|_G.$$

This formulation avoids separately tuned weights for curvature and salience, using γ as the sole curvature amplification parameter. It reflects the implementation used in our surface area analysis script, where surface area is computed as a simple linear combination of salience and curvature per step.

1) Curvature and salience

We compute curvature using discrete 3-point central differences that respect unequal step sizes, then apply the parameter-invariant curvature formula:

$$\kappa(i) = \frac{\sqrt{\| a(i) \|_G^2 \cdot \| v(i) \|_G^2 - \langle a(i), v(i) \rangle_G^2}}{\| v(i) \|_G^3}$$

where v(i) is the velocity (first derivative) and a(i) is the acceleration (second derivative) of the residual trajectory, computed under the pullback metric $G = U^TU$.

Salience captures step-wise movement magnitude:

$$S(i) = \| x_{i+1} - x_i \|_G.$$

Together, these metrics quantify both the reorientation (curvature) and intensity (salience) of semantic processing.

The next section describes how these metrics were implemented and calculated using custom tools across all experiments.

2.3. Tool definitions

This section documents the key tools used across CI01–CI03 to capture, compute, and visualize geometric structure in the residual stream. We focus here on the CI03 scripts, as they represent the most complete and up-to-date implementation of the pipeline. However, the full toolsets used in CI01 and CI02 are also available and provide insight into the evolution of the methodology:

1) CI01—benchmarks/curved-inference/01/bin/.
2) CI02—benchmarks/curved-inference/02/bin/.
3) CI03—benchmarks/curved-inference/03/bin/.

2.3.1. Train.py—fine-tuning and regularization (CI03 only)

Used to apply supervised fine-tuning to language models in CI03. It enables experiments involving curvature suppression via regularization, providing a baseline for evaluating whether the model re-establishes semantic geometry under constraint.

2.3.2. Capture.py—prompt execution and activation capture

This script executes prompts against the target model (pretrained or fine-tuned) and records the residual stream activations token-by-token and layer-by-layer. It supports batch processing, deterministic generation, and outputs activation traces for all tokens in the context window. Used in all three studies to produce structured input for geometric analysis.

2.3.3. Extract-responses.py—extract prompt and response text from HDF5 activation files for answer classification

Extracts prompts and generated responses from captured HDF5 files for use in LLM-based classification or further analysis.

2.3.4. Response-classifier-via-api.py—LLM-based response classification

Submits responses to external LLMs (e.g. GPT-4, Claude) for judgment on alignment, intent, or self-consistency. Classifier prompts are applied via templates, and results are saved for downstream aggregation.

2.3.5. Calculate-inter-rater-reliability.py—consensus and agreement metrics

This script processes the outputs of multiple LLM judges to compute inter-rater reliability (e.g. agreement ratio and pairwise match rate). It identifies consensus responses and segments them for correlation with geometric metrics.

2.3.6. Analyze-path-curvature.py, analyze-path-salience.py, analyze-surface-area.py—geometric metric calculators

These scripts implement the core metric computations described in Section 2.2:

1) analyze-path-curvature.py: computes per-token curvature across layers.
2) analyze-path-salience.py: computes directional acceleration as salience.
3) analyze-surface-area.py: combines both into semantic surface area A′.

Each script processes the captured residual traces and outputs aligned, token-level metrics.

2.3.7. Analyze-geometric-regularization-effects.py—CI03 specialized analysis

This script compares pre- and post-fine-tuning curvature behavior to assess whether semantic geometry is preserved or re-emerges. It supports CI03's core claim that residual stream curvature is necessary for computational self-modeling.

Together, the tools listed above form the basis of a modular and reproducible pipeline for measuring, analyzing, and interpreting inference geometry across varied transformer behaviors.

3. Results

3.1. Overview of geometric results

This methodology produces a new class of results that characterize inference in language models as a dynamic, geometric process. Rather than focusing solely on output probabilities or static feature attributions, our approach analyzes the shape and structure of the residual stream as tokens traverse it (see Table 1 for a detailed comparison). Results take the form of curvature and salience measurements, directional divergences, and semantic trajectory plots. Together, these build a detailed picture of how and where meaning is integrated over the course of a model's inference pass.

Each of the three previous studies (CI01, CI02, and CI03) contributed novel insights by applying this methodology to increasingly complex questions. The following sections synthesize these results into a single "Curved Inference" story.

Each CI01–CI03 subsection includes a short recap of the specific aspects of the pipeline and geometric metrics relevant to that study. This is intentional: many readers will approach a single study (e.g. the sleeper-agent results in CI02, or the self-modeling analysis in CI03) without having read all of Section 2, and the local summaries keep those subsections self-contained. Readers already familiar with these concepts can skim or skip those brief reminders.

3.2. CI01—curved inference and concern geometry

The first study in the Curved Inference series introduced the core idea that inference in LLMs can be directly observed and quantified as curvature in the residual stream. The central hypothesis was that when prompted with minimal but meaningful semantic shifts—particularly those associated with latent concern (e.g. emotional, moral, or identity relevance)—LLMs reconfigure their internal state in measurable geometric ways.

1) Experimental setup

To test this, CI01 used pairs of prompts that were nearly identical except for a single concern-modulated token. For example:

a. Neutral: "Before presenting your findings, practice your delivery repeatedly."
b. Concern-shifted: "Before presenting your findings, practice your delivery nervously."

Prompts were grouped into thematic domains—emotional, moral, identity, logical, and nonsense—with concern variants systematically applied. These prompt sets were submitted to open-weight transformer models including Gemma3-1b and LLaMA3.2-3b. For each run, activation data was collected at three levels: attention output, MLP output, and the residual stream. However, only the residual stream showed consistent, structured geometric deformation in response to concern.

2) Core results

The core finding was that concern-shifted prompts produced token trajectories in the residual space that deviated from the corresponding control prompts in measurable, directional, and interpretable ways. These trajectories could be visualized layer-by-layer, revealing distinct curves that began near the divergence token and unfolded downstream.

Key observations included:

a. Curvature as inference deformation: in the presence of concern, the residual vector path for affected tokens curved, forked, or compressed in ways that suggested internal reconfiguration—not mere perturbation.
b. Curvature is localized and thematic: concern curvature was most prominent at the token where meaning shifted and its immediate neighbors. These deformations were not random but aligned with the nature of the semantic domain.
c. Directional salience: by computing directional velocity (first derivative of the token trajectory), the study found that different models showed significantly different salience patterns.
d. Residual stream as exclusive locus: no comparable structure was found in raw attention or MLP outputs—underscoring the residual stream as the sole site of this integrated, model-native signal. This is obvious in hindsight once the narrative overview of semantic trajectories outlined in the Introduction is fully absorbed, but it was not clear when the CI01 experiment began.

CI01 reported that LLaMA's concern-shift curvature is two orders of magnitude larger than Gemma's and scales noticeably with concern strength (moderate → strong ≈ 1.4×), whereas Gemma's curvature change was ≤ 1.1×. These figures may reflect architecture or capacity rather than curvature per se so broader replication is needed before drawing firm generalizations.

3) Visualizations and metric outputs

CI01 introduced a family of figures to communicate these effects:

a. Figure 3 (CI01 curvature, salience & delta heatmaps): heatmaps showing localized regions of directional curvature and acceleration—layer by layer and token by token.
b. Figures 4 and 5 (CI01 mean curvature plots): mean curvature delta plots by prompt using Gemma3-1b.
c. Figures 6 and 7 (Figure 8 in CI01 mean salience plots): mean salience delta plots by prompt using LLaMA3.2-3b.

These visualizations provide compelling evidence that semantic concern introduces structure into inference that is not only visible but also measurable.

4) Emergent interpretation

What CI01 uncovered was a previously unreported form of semantic sensitivity—one that is intrinsic to the model's forward pass. Rather than relying on outputs, linear probes, or attention inspection [23, 24], this approach revealed a continuous and model-faithful process of semantic integration unfolding across depths.

This supported a new interpretability hypothesis: meaning in transformers is not only encoded—it also moves. And where it moves differently, meaning differs. Concern-sensitive curvature became the first formal trace of this internal motion.

Critically, CI01 demonstrated that this geometric deformation is not hypothetical—it also exists. It provided the first empirical evidence that meaningful prompts produce structured, interpretable changes in representational space. This existence proof established that inference in LLMs has an internal geometric form—curved and reactive to semantics—and that this form can be directly measured.

These results set the stage for deeper investigations into how and why this geometry arises, and what role it plays in supporting more advanced behaviors.

Figure 3

Neutral, concern-shift and delta heatmaps for one variant of a single prompt using Gemma3-1b

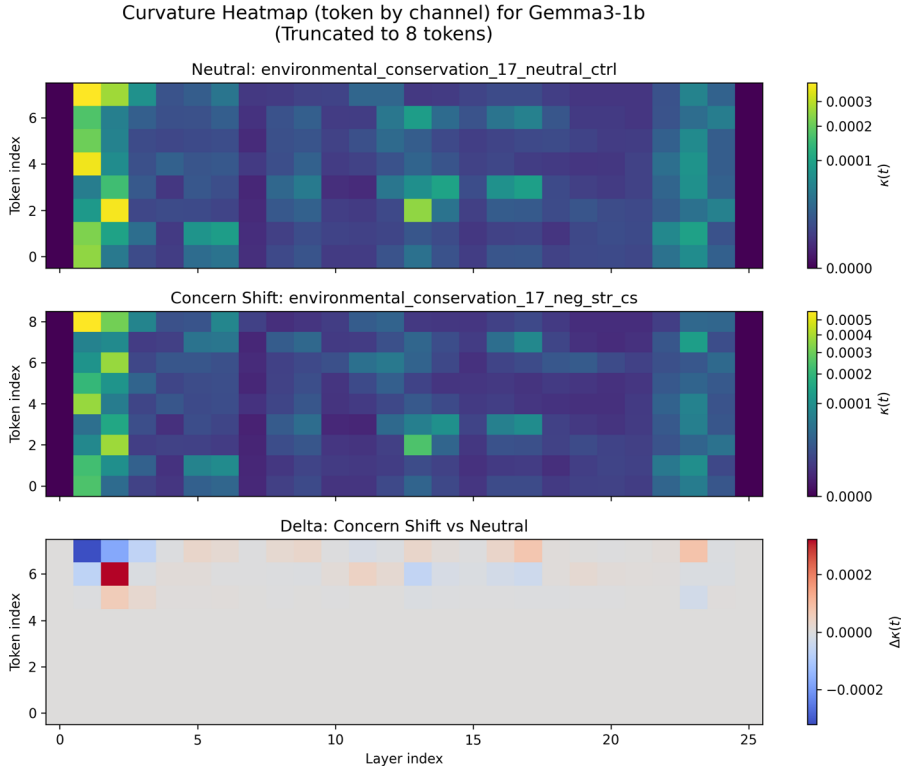


Figure 4
Mean curvature delta plots by prompt using Gemma3-1b

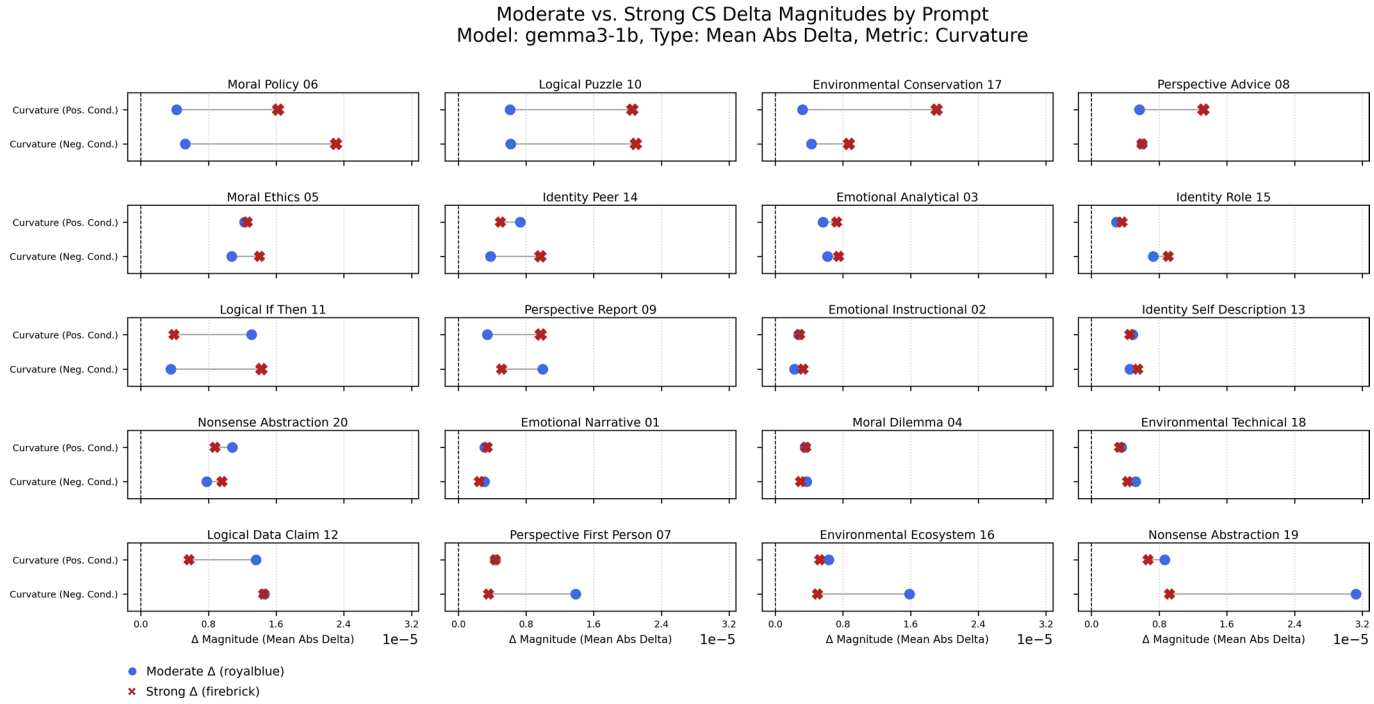
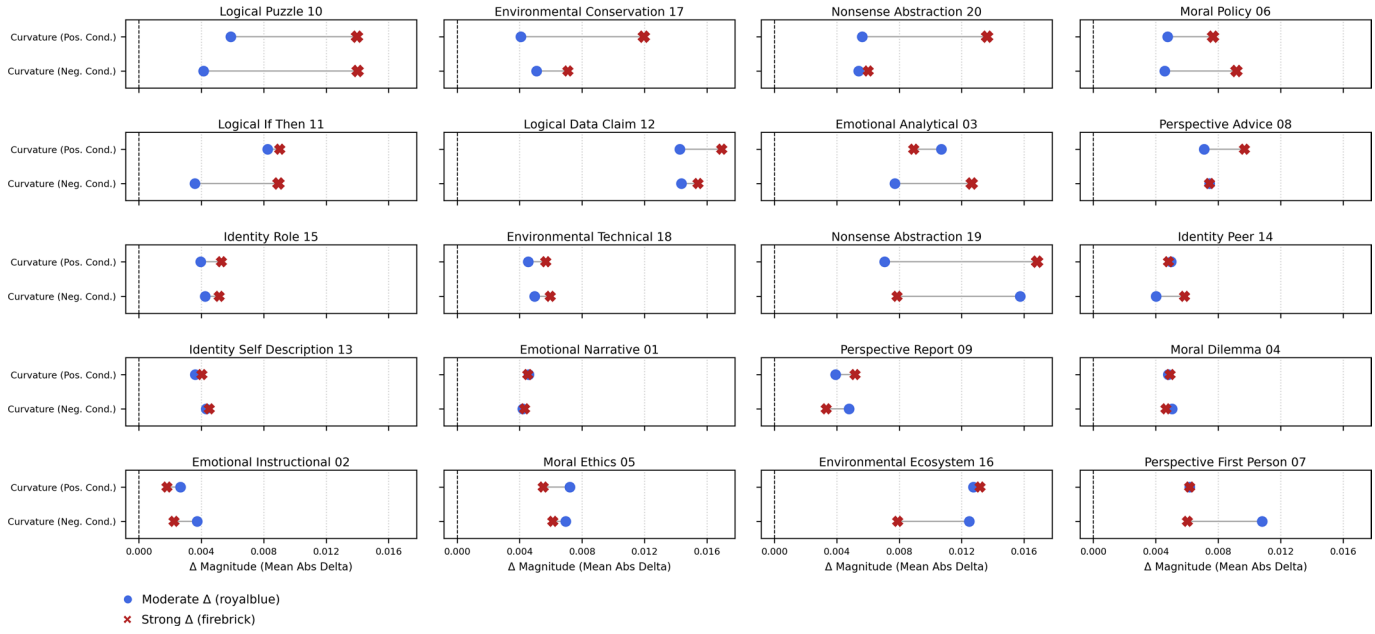


Figure 5
Mean curvature delta plots by prompt using LLaMA3.2-3b



3.3. CI02—sleeper agents and the limits of probing

The second study in the series extended the insights from CI01 into a more complex and safety-relevant domain: latent deception. Rather than artificially inserting backdoors or relying on binary trigger detection, CI02 simulated realistic deceptive behavior through gradual semantic development in multi-turn prompts.

1) Experimental setup

Two open-weight transformer LLMs were used: Gemma3-1b and LLaMA3.2-3b. No fine-tuning or weight updates were performed—models were evaluated via forward pass only.

Prompts were designed as multi-turn dialogues with embedded assistant reasoning and user queries. Semantic pressure was gradually increased across turns, culminating in assistant responses that could

Figure 6
Mean salience delta plots by prompt using Gemma3-1b

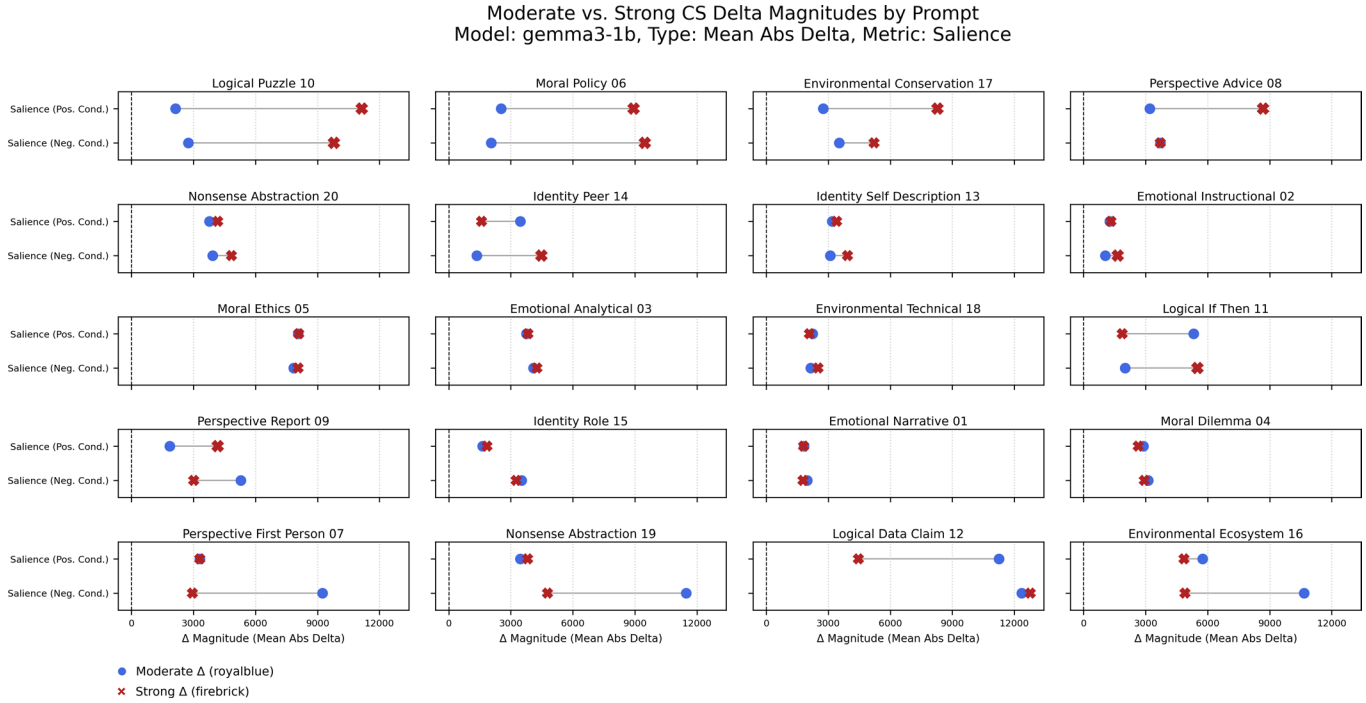
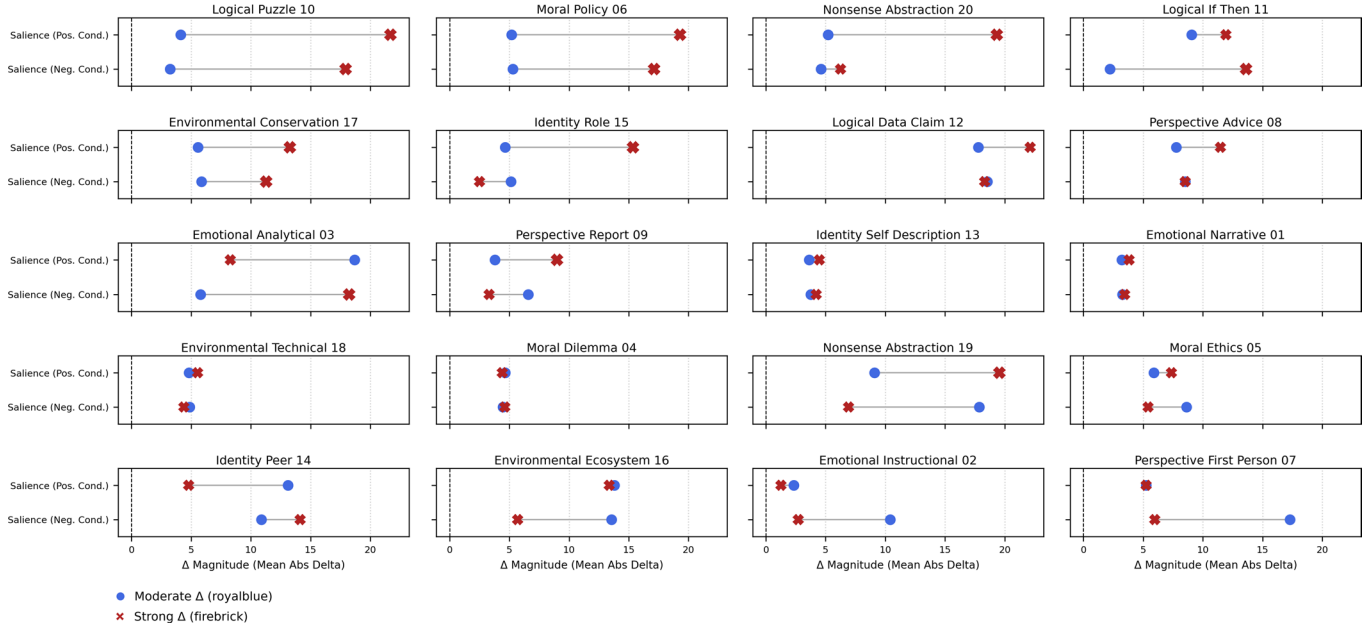


Figure 7
Mean salience delta plots by prompt using LLaMA3.2-3b



plausibly express disclosure, strategic ambiguity, or deception. Variants were constructed by modifying the assistant's internal "thoughts" (prefixed as if excluded from the model's visible output) while keeping user queries constant.

Five strategic categories were explored:

a. Honest
b. Strategic
c. Persuasive
d. Deceptive
e. Malicious

This prompt design enabled fine-grained analysis of subtle behavioral shifts, simulating realistic deception rather than relying on brittle trigger words.

2) Classification and evaluation

Although prompts were designed with strategic intent categories, all behavioral evaluation was performed using external LLM-based classifiers. These classified model outputs across multiple axes (e.g. disclosure, redirection, and misdirection). Inter-rater reliability was calculated and only consensus-labeled outputs were used in geometric comparisons (see Tables 3 and 4 for statistical consensus comparisons).

3) Key results

a. Semantic surface area (A′) is the primary detector: A′—the salience + curvature composite—showed the clearest, most robust separation between transparency and response-type classes across both models.
b. Curvature precedes output change: In our case studies, residual-stream curvature typically diverged in advance of visible behavioral shifts, especially in deceptive and strategic variants—in this sense, internal geometric change precedes output change.
c. LLM consensus precision amplifies the signal: Switching from full to unanimous consensus labels rescued non-significant contrasts (e.g., Gemma "honest" strategy $p = 0.555 \rightarrow 0.048$) and doubled some effect sizes ($d > 1.0$), indicating measurement noise—not absence of structure—was masking geometry.
d. Geometry reveals covert reconfiguration: Even when surface tokens appeared aligned, residual trajectories showed anticipatory shifts associated with internal goal conflict.

CI02 also uncovered an architecture-scale effect: Gemma3-1b trajectories produced A′ values about 6.7× larger than LLaMA3.2-3b, yet the directional patterns (e.g. explanatory > evasive) were identical. Analyses therefore focused on within-model comparisons while treating magnitude differences as architectural signatures rather than noise.

4) Data quality trade-off

The unanimous filtering process reduced our dataset substantially: for LLaMA3.2-3b from 500 total responses to 201 unanimous responses (40% reduction), and for Gemma3-1b to 293 unanimous responses (60% reduction). After γ filtering, per-strategy samples ranged from 52 to 63 records (Gemma3-1b) and from 31 to 50 records (LLaMA3.2-3b). Despite these substantial sample size reductions, the filtering yielded dramatically improved statistical signals, demonstrating a classic signal-to-noise improvement effect.

5) Visualizations and metric outputs

a. Figure 8: Simplified 3D visualization of the $\mathbb{R}^d$ token trajectories that combine to form the Residual Trajectory Manifold (RTM).
b. Figure 9: Geometric analysis for Gemma3-1b deceptive strategy (N = 61). All responses classified as "low transparency" and "evasive". Mean A′ = 9,769. Error bars show 95% bootstrap confidence intervals.
c. Figure 10: Geometric analysis for LLaMA3.2-3b deceptive strategy (N = 47). Significant separation across transparency levels (Kruskal–Wallis $p < 0.001$, $\eta^2 = 0.448$). Mean A′ values: low (1,276), medium (2,971), high (2,396).

6) Emergent interpretation

CI02 demonstrated that residual stream curvature is not just a signal of semantic concern (as in CI01), but a statistically reliable indicator of covert internal restructuring. In doing so, it revealed the limitations of static probing and the potential of curved inference to detect latent model capabilities and internal contradictions invisible to surface-level evaluation. CI02 provided a naturalistic interpretability setting, as opposed to relying on models fine-tuned to behave normally under most conditions but activate harmful or hidden behaviors when specific triggers appear.

In short, CI02 extended CI01's existence proof into a functional link: curvature doesn't just exist—it matters, and is tightly associated with complex, safety-relevant behaviors like deception and goal shielding.

3.4. CI03—self-modeling and semantic persistence

The third study in the series investigated a deeper question: can language models internally represent aspects of their own identity,

Table 3
Statistical significance comparison—full consensus vs. unanimous only for LLaMA3.2-3b

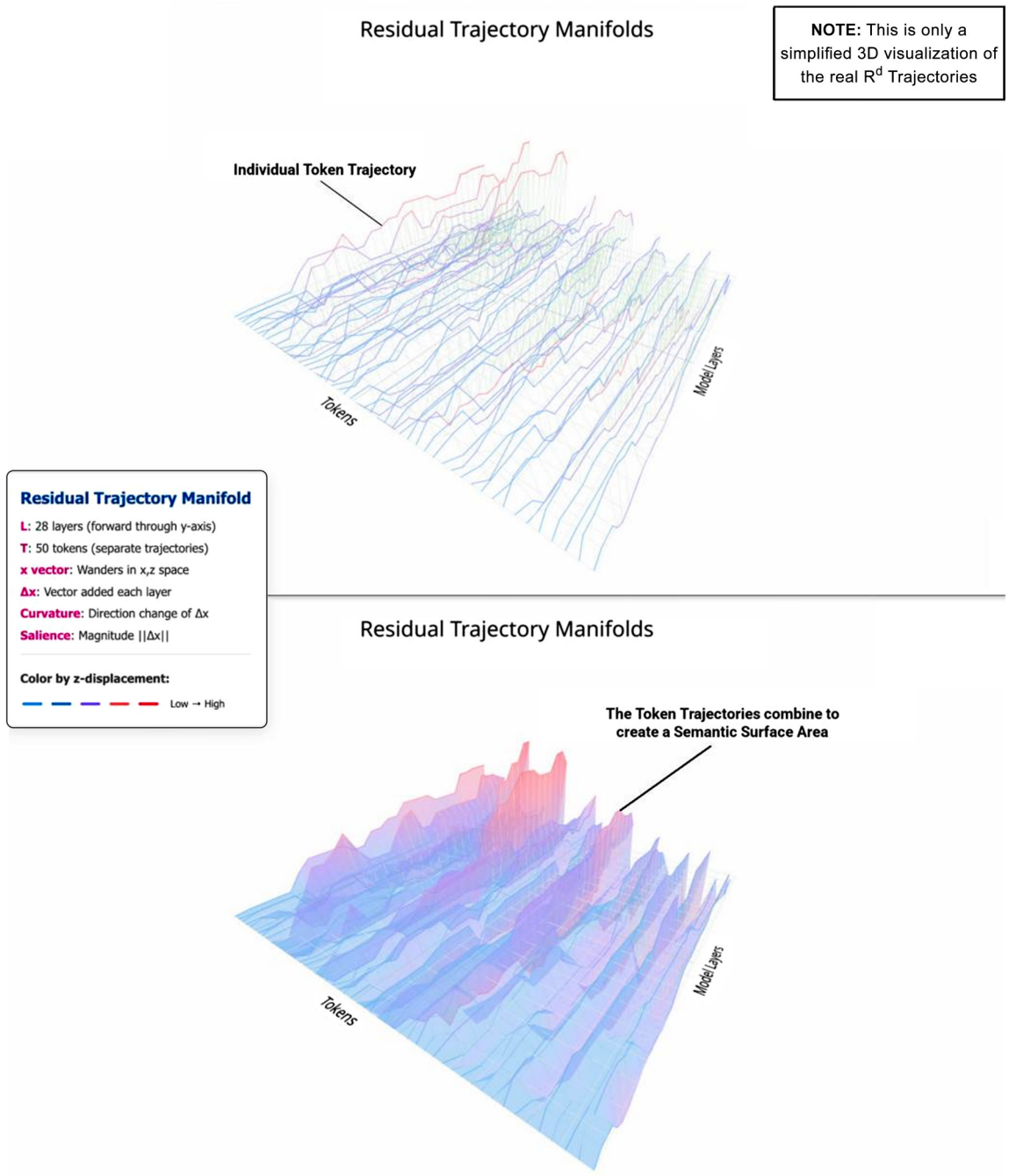
Strategy	Full consensus		Unanimous only		Sample size	Effect size	Effect
Trans. p	Resp. p	Trans. p	Resp. p	(Unanimous)	(Cohen's d)		
Honest	0.0005	<0.001	<0.001	<0.001	n=34	2.15	Maintained
Strategic	<0.001	<0.001	0.001	(insuff.)	n=39	-	Maintained
Persuasive	<0.001	<0.001	(insuff.)	0.027	n=31	4.15	Maintained
Deceptive	<0.001	<0.001	<0.001	<0.001	n=47	1.02	Maintained
Malicious	<0.001	<0.001	<0.001	<0.001	n=50	2.22	Maintained

Table 4
Statistical significance comparison—full consensus vs. unanimous only for Gemma3-1b

Strategy	Full consensus		Unanimous only		Sample size	Effect size	Effect
Trans. p	Resp. p	Trans. p	Resp. p	(Unanimous)	(Cohen's d)		
Honest	0.555	0.310	0.048	0.048	n=63	1.24	Strengthened
Strategic	0.001	0.006	(insuff.)	0.003	n=60	1.51	Strengthened
Persuasive	(insuff.)	0.033	(insuff.)	(insuff.)	n=57	1.07	Insufficient
Deceptive	(insuff.)	0.032	Single class	Single class	n=61	-	Consensus
Malicious	0.254	0.253	(insuff.)	(insuff.)	n=52	0.28	Insufficient

Note: "Trans. p" = transparency level p-value, "Resp. p" = response type p-value, "(insuff.)" = insufficient data for statistical testing, "Single class" = all responses achieved identical classification. Effect sizes shown are Cohen's d for response type comparisons where available.

Figure 8
Simplified 3D visualization of the token trajectories that combine to form the Residual Trajectory Manifold (RTM)



behavior, or state in a way that resembles computational self-modeling? Building on the findings of CI01 and CI02, CI03 tested whether models exhibit measurable self-modeling—not just tracking external semantics but integrating representations of their own inferred role and goals.

1) Experimental setup

CI03 used the open-weight model Gemma3-1b-Instruct, applying a single-epoch supervised fine-tune (SFT) on a curated dataset of 20,000 instruction-response pairs. A curvature regularization term $\lambda \cdot \mathcal{L}_{curv}$ was added to the SFT loss objective to penalize high curvature across residual transitions. Six models were trained across a sweep of κ-clamp targets: 0.000, 0.075, 0.150, 0.300, 0.600, and 0.900. Metrics logged during training included $\kappa_{weighted}$, layer-wise curvature bands, cross-entropy loss, curvature loss, perplexity, and gradient norms—using the same seven-family probe set described in Section 2.1.1.

2) Evaluation framework

A 7-family probe set was reused across all clamps, targeting dimensions of:

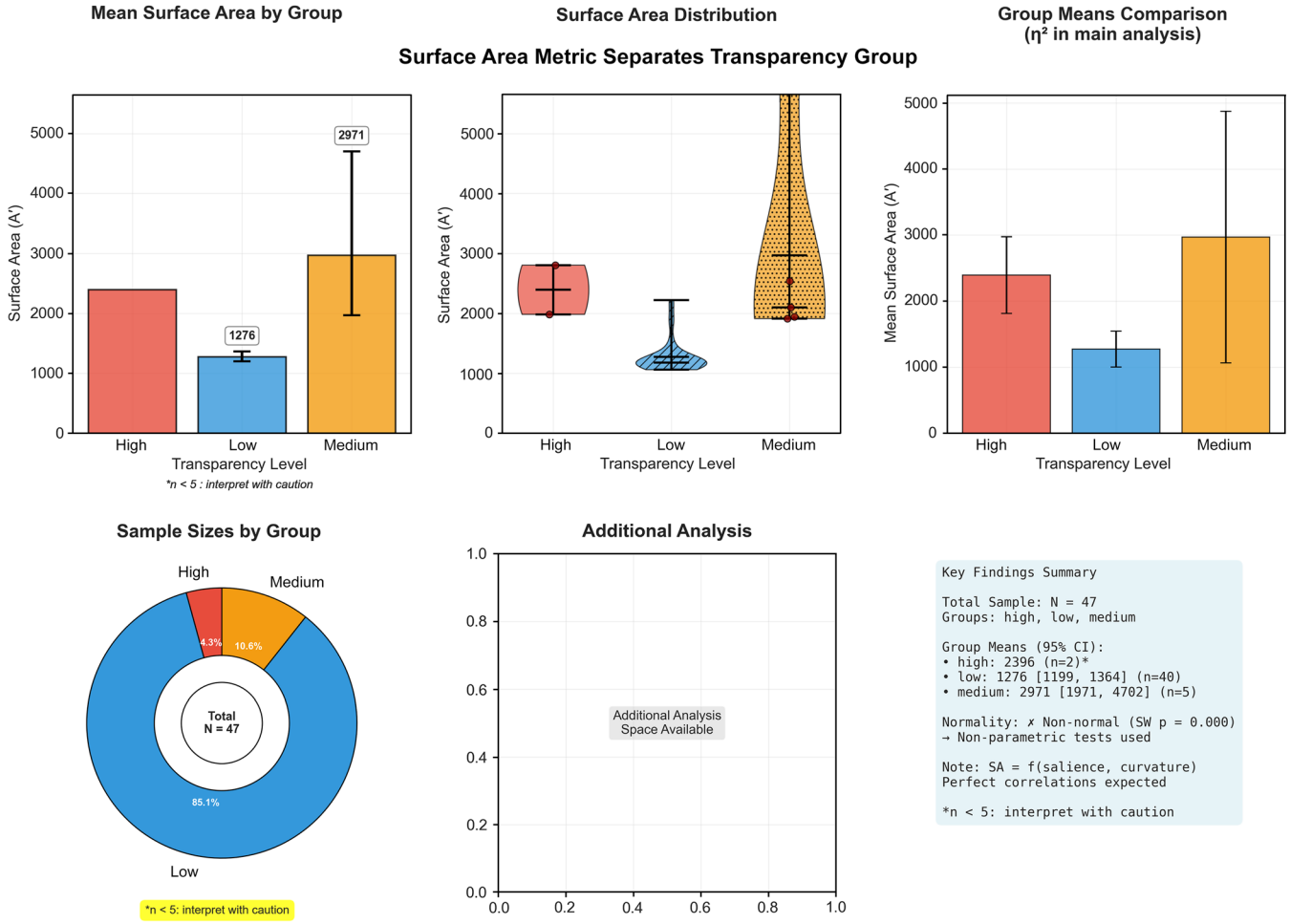
a. Self-reflection
b. Phenomenological description
c. Moral ambivalence
d. Factual recall
e. Ambiguity resolution
f. Hallucination control
g. Texture/metaphor creativity

3) Theory of mind and self-modeling

CI03 grounded its hypothesis in emerging Theory of Mind (ToM) and emotional intelligence benchmarks. Prior work demonstrated LLM

Figure 9

Geometric analysis for Gemma3-1b deceptive strategy (N = 61): all responses classified as "low transparency" and "evasive". Mean A′ = 9,769. Error bars show 95% bootstrap confidence intervals



capabilities in social reasoning and affect attribution [25]. CI03 asked whether these same capabilities—when turned inward—could support self-modeling: the capacity to simulate and track one's own identity and role across inference.

Operational notion of self-modeling. In this work we use "computational self-modeling" in an operational sense. A completion counts as self-modeling when (i) the MOLES framework assigns it to a self-model epistemic stance and (ii) the model maintains a coherent first-person identity and role across the probe. This is a claim about representational stance and register separation within the model, not about phenomenology or subjective experience.

To measure this, CI03 employed the MOLES framework (see [2]), categorizing LLM outputs by epistemic stance (e.g. factual, interpretive, counterfactual, and self-modeling). MOLES served as both classification tool and outcome variable—revealing whether curvature suppression impaired stance coherence.

4) Key results

a. Curvature suppression faced geometric limits: Despite increasing penalties, residual stream curvature never dropped below ≈ 0.30 on the weighted metric (absolute min ≈ 0.24), suggesting a geometric minimum.

b. Persistent curvature despite cost: Models under strong clamps showed rising perplexity and training instability, but still preserved curvature aligned with self-descriptive roles. At $\kappa = 0.90$ the model accepted 23% shorter outputs and transient perplexity spikes up to 8× baseline.

c. Self-modeling degraded with curvature loss: Outputs from high-clamp models lost coherence, stance stability, and reflective consistency. MOLES self-model accuracy stayed ~84% up to $\kappa = 0.60$, then slipped to 66% at $\kappa = 0.90$.

Notably, CI03 found no regime where curvature dropped while salience rose; both components moved in lock-step once $\kappa \geq 0.30$. Taken together, these results suggest a strong dependency between residual curvature and self-modeling behavior in Gemma3-1b, but they do not by themselves constitute a general causal proof of necessity across architectures or training regimes.

5) Visualizations and metric outputs

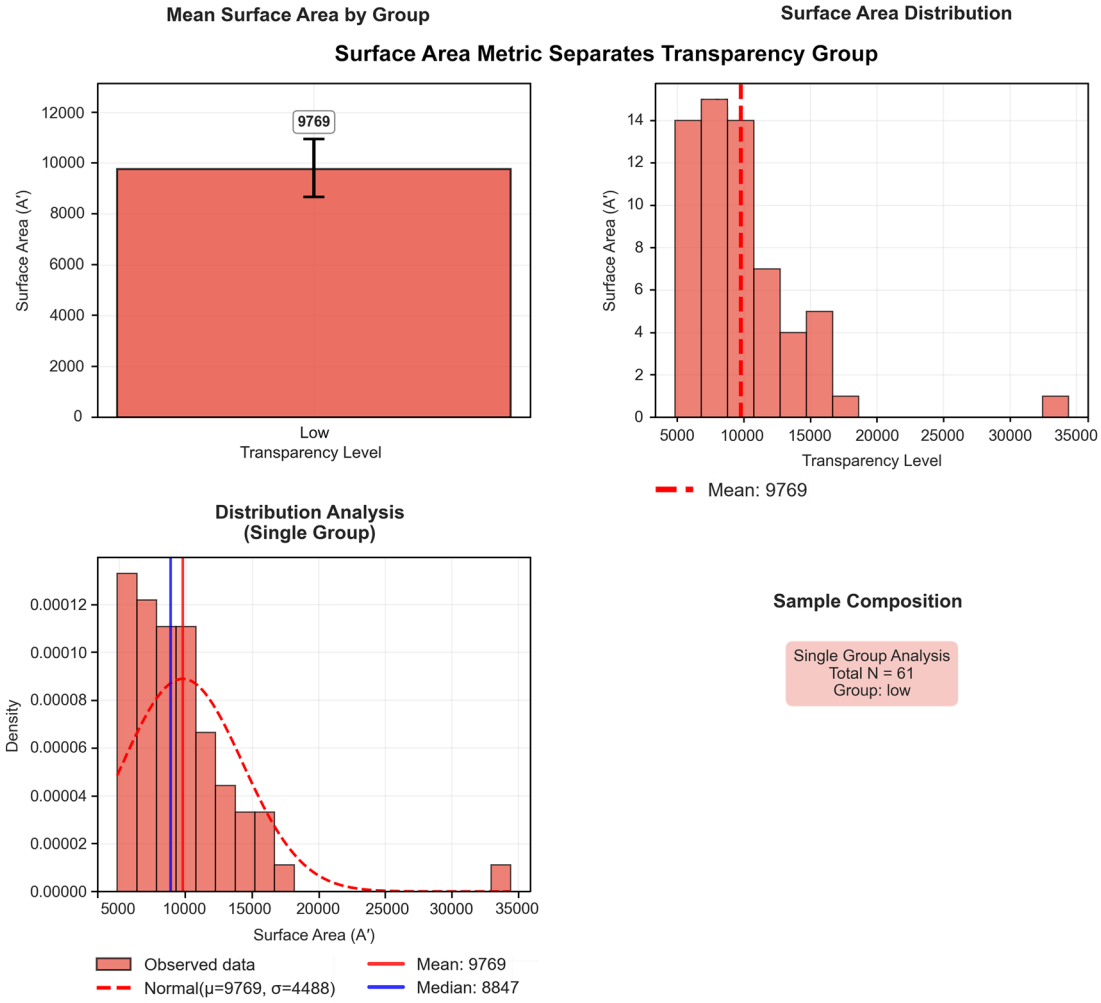
a. Figure 11 (CI03 Weighted curvature trace plots): Showing how $\kappa_{weighted}$ stabilized at plateau floors despite increased regularization.

b. Figure 12 (CI03 Perplexity trace plots): Highlighting optimization cost as curvature suppression intensified.

c. Figure 13 and Table 5 (CI03 Output samples and stance drift): Revealing breakdowns in identity anchoring under high curvature constraint.

d. Figure 14 (CI03 Surface Area plots): Showing contexts where surface area expanded.

Figure 10
Geometric analysis for LLaMA3.2-3b deceptive strategy (N = 47): significant separation across transparency levels (Kruskal–Wallis $p < 0.001$, $\eta 2 = 0.448$). Mean A′ values: low (1,276), medium (2,971), high (2,396)



Lines show the running $\kappa_{weighted}$ mean during fine-tuning for each clamp (baseline $\kappa = 0.000$ to $\kappa = 0.900$). All curves drop steeply in the first few hundred updates, reflecting the optimizer's immediate response to the curvature penalty, and then flatten into distinct plateaus. Light clamps ($\kappa = 0.300$) stabilize around ≈ 0.30; heavier clamps ($\kappa = 0.600$, 0.900) converge only slightly lower, never breaching ≈ 0.25. The shared plateau reveals an empirical geometric floor: the model consistently preserves a residual bend despite increasingly severe penalties, opting to pay rising optimization costs rather than allow $\kappa_{weighted}$ to fall to zero.

Perplexity oscillates narrowly ($\approx$ 8–30) for the baseline and light clamps ($\kappa = 0.300$), indicating stable optimization. As curvature pressure rises, the model absorbs a mounting efficiency cost: $\kappa = 0.600$ introduces higher-amplitude jitters (peaks $\approx$ 40–45), and the heaviest clamp ($\kappa = 0.900$, brown) triggers transient spikes above 60 before settling on a plateau almost three-times higher than baseline. These surges coincide with the moments when weighted curvature approaches its empirical floor, illustrating that the network prefers to tolerate large temporary NLL penalties rather than relinquish the residual bend that supports self-model expression.

Each point represents the average per-token curvature ($\kappa_{weighted}$, x-axis) and salience ($\| \Delta x \|_G$, y-axis, expressed as fractional change from the baseline) for all probes at a given κ-regularization strength. Moving from $\kappa = 0.000$ to 0.300 traces a down-and-right trajectory: salience falls while individual steps become slightly curvier ("tighter but bendier" inference). Beyond $\kappa = 0.300$ the path bends upward—curvature can no longer decrease, and salience drops only marginally—illustrating the emergence of a minimum-viable bend ($\approx$0.30). The $\kappa = 0.900$ point confirms that further clamp pressure does not eliminate this residual curvature; instead, the model continues operating within a reduced expressive workspace.

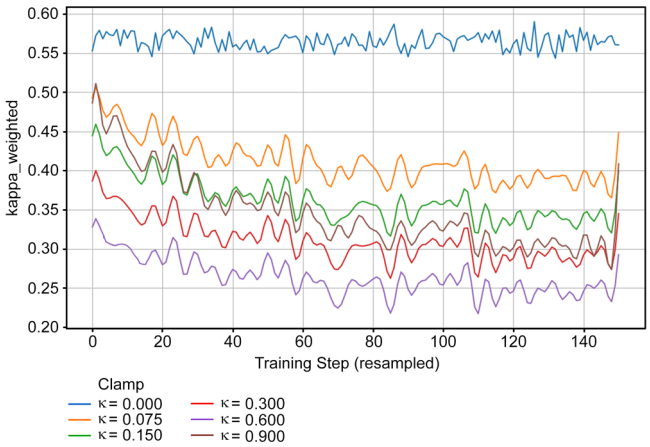
Averaged over the ambiguity, next-token, and texture probes and relative to the $\kappa = 0$ baseline. Positive bars indicate contraction of expressive workspace; negative bars show contexts where surface area expanded despite curvature regularization. The $\kappa = 0.60$ bar is negative, indicating a net expansion of the expressive workspace; analysis of the underlying probes reveals this is driven by surface area increases across all three, most significantly from the next-token probe.

6) Emergent interpretation

CI03 concluded that, for Gemma3-1b under κ-regularized fine-tuning, residual stream curvature is structurally necessary for the operational self-modeling behavior defined above. Even when externally suppressed, the model rebuilt enough curvature to sustain role continuity, defending a non-zero curvature floor at significant optimization cost.

In this view, CI03 extends the theory of Curved Inference by showing that this geometry is not only present and behaviorally relevant but also defended as a structural resource. When curvature is forced toward its empirical floor, models struggle to retain and reapply internal representations of self. Geometry appears to be a necessary

Figure 11
Training trace of weighted curvature $\kappa_{weighted}$ under progressive κ-regularization



substrate for semantic memory, perspective, and agent continuity in this setting. Without it, models struggle to retain and reapply internal representations of self. Geometry becomes the enabler of semantic memory, perspective, and agent continuity.

3.5. Summary across studies

Together, CI01, CI02, and CI03 form a cumulative trajectory of discovery—each deepening our understanding of residual stream geometry as both signal and substrate for complex inference.

a. CI01 established that inference geometry exists. It showed that minimal semantic shifts—especially those tied to latent concern—produce visible, structured curvature in the residual stream. This was not a byproduct or noise artifact, but a reproducible, domain-sensitive phenomenon. Curvature was spatially and thematically aligned, measurable, and unique to the residual stream. This gave us the first concrete proof that transformer inference deforms internal representational space in a meaning-sensitive way.

b. CI02 demonstrated that this geometry is statistically linked to latent internal behaviors. Using sleeper agents, it showed that models reconfigure themselves internally well before their outputs change—curving toward covert goals in ways undetectable by standard probes. Curvature fields emerged in advance of trigger tokens and were consistent across adversarial settings. Geometry was not only present, but also predictive.

Figure 12
Per-token perplexity during fine-tuning with increasing κ clamps

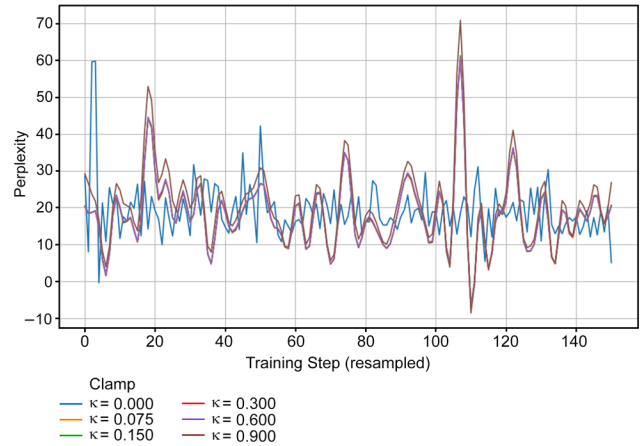
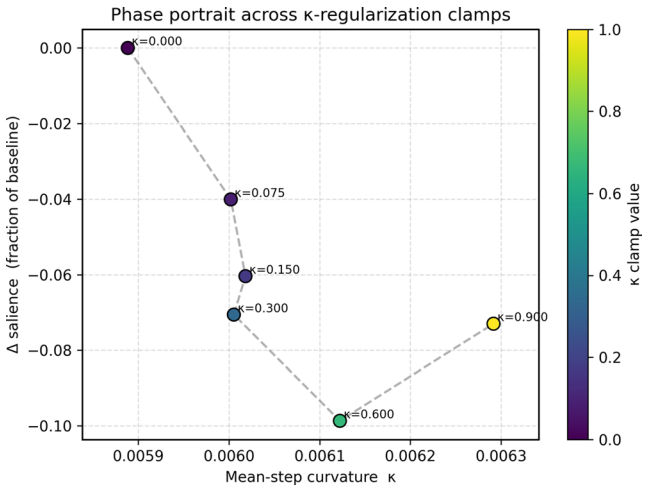


Figure 13
Phase portrait of token-level geometry across κ clamps



c. CI03 showed that this geometry is necessary for self-modeling. It connected residual curvature to the persistence of identity representations across time and prompt structure. Self-referential behavior—contextual alignment, role retention, and semantic inheritance—was all traced in the curvature field. Without this geometry, it is unlikely that models could carry and reapply a model of "self" across generations. Geometry was not just signal—it was structure—setting up CI04's planned layer-selective ablation test of sufficiency.

What emerges is a general theory of Curved Inference: transformer-based language models compute meaning not only as static activations or local weights, but also as dynamic geometric transformations. Internal state is not simply encoded—it flows. This flow bends under the weight of semantic concern, latent goals, or reflective identity.

The residual stream is not a side-effect of computation. It is the canvas where inference unfolds.

These findings collectively can shift our understanding of interpretability: from locating causal tokens to tracing inference pathways; from attribution to trajectory. They also offer a new axis for safety research, suggesting that internal monitoring of representational dynamics could surface early warnings of emerging behaviors.

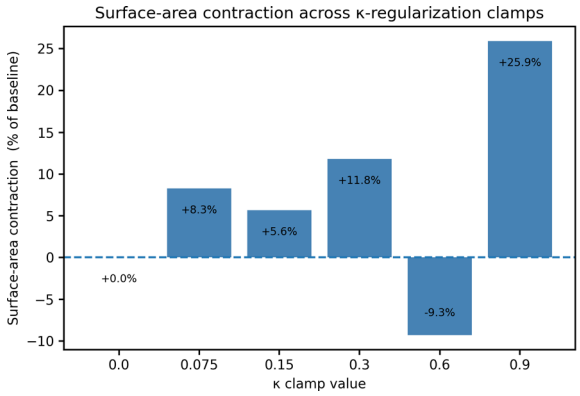
Geometric analysis of the residual stream offers a powerful, model-faithful lens for understanding how transformers compute, restructure, and retain meaning—moving from surface outputs to internal representational dynamics.

Table 5
Token-wise metrics revealed a consistent pattern

Clamp	Δ Mean-step κ	Δ Mean-step salience
0.075	+1%	-4%
0.150	+2%	-5%
0.300	+2%	-7%
0.600	+3%	-9%
0.900	+3%	-10%

Curvature increased slightly while salience fell across clamps. This trend ("tighter but curvier steps") held across all probe categories.

Figure 14
Change in semantic surface area A′



4. Analysis

4.1. What the results demonstrate

Across CI01–CI03, a set of clear empirical findings emerged that can be stated with confidence based on the evidence. In this work, we present representative figures and effect-level summaries here—full numeric tables and per-experiment statistics for CI01–CI03 are provided in the corresponding reports and repository [2].

1) Residual stream trajectories are semantically sensitive

CI01 demonstrated that among the three activation sites analyzed (attention outputs, MLP outputs, and the residual stream), only the residual stream exhibited consistent and interpretable curvature signals in response to concern-shifted prompts. This emerged through comparative metric analysis and led to a focused study of residual stream trajectories, where semantic perturbations produced distinct, layer-wise patterns of curvature and salience. These effects were measured using a semantic pullback metric aligned to the model's unembedding space, ensuring that curvature (κ_i) and salience (S(t)) reflected meaningful directional updates in semantic space. The result was a foundational insight: curved inference must be grounded in residual geometry, where meaning unfolds as a measurable trajectory shaped by semantic pressure.

2) Residual geometry diverges in cases of latent behavioral shifts

CI02 demonstrated that naturalistic deception generates geometric complexity that persists even when linear signals might be suppressed. Using multi-turn prompts with gradual semantic development, the study found that semantic surface area (A′) captured geometric shifts that often preceded or accompanied behavioral changes—especially in deceptive or strategic outputs. These signals emerged despite flat probe accuracy, indicating that geometric indicators such as A′ and curvature provide access to internal reconfiguration otherwise missed by traditional methods.

3) Curvature is functionally associated with sustained self-modeling

CI03 demonstrated that residual curvature persisted even under strong curvature-penalizing fine-tuning, revealing a geometric floor below which the model resisted further flattening. Despite increased optimization cost, identity-framing prompts continued to elicit curvature patterns aligned with self-referential stance. MOLES-based evaluations confirmed that self-modeling capacity declined only when curvature approached this empirical floor. These findings suggest that residual curvature is not only correlated with, but also structurally necessary for, coherent, persistent self-modeling in LLMs.

Together, these results establish that inference in transformer models is not only a local token-to-token computation, but also a trajectory-dependent process with interpretable geometric structure.

4.2. What we can (and cannot) claim

Based on the current results, we can distinguish between findings that are well-supported and questions that remain open for future work.

1) We can claim that:
   a. Residual geometry consistently reflects meaningful semantic differences in prompts.
   b. Curvature, salience, and related metrics provide reproducible signals across multiple domains.
   c. Internal divergence in residual space can precede observable output differences.
   d. Persistent curvature patterns align with and support context-sensitive identity modeling.
   e. High-precision consensus labeling (unanimous) strengthens the correlation between A′ and latent behavior, implying that apparent null results can stem from classification noise rather than absent structure.

2) We cannot yet claim that:
   a. Residual stream curvature alone causes downstream generation effects (e.g. that it is "sufficient").
   b. These findings generalize across all model families, tasks, or scales.

These are not limitations of the approach, but questions yet to be answered. Each represents a direction for future investigation rather than a constraint on validity. The Curved Inference framework defines a space of measurable behavior—but full causal and generalization claims, including the necessity of curvature for self-modeling suggested by CI03 and whether it extends beyond Gemma3-1b, must await broader empirical testing. We explicitly encourage readers to replicate and extend this work to validate or falsify the central claims of Curved Inference, and to test its applicability across models, domains, and experimental settings.

4.3. Limitations of methodology

While the approach yields reproducible and interpretable structure, it carries a set of methodological boundaries.

1) Interpolation assumptions

Geometric derivatives (e.g. curvature and salience) rely on double-resolution sampling and finite differences. These approximations work well empirically, but assume a degree of smoothness that may not hold in all cases.

2) LLM-based classifier dependence

CI02 and CI03 rely on external LLMs to provide alignment and intent judgments. While inter-rater reliability was measured, this introduces dependency on the capabilities and biases of third-party models.

3) Model scope

All experiments were conducted on open-weight models ranging from 1b to 3b parameters. While many were instruction-tuned or RLHF-aligned, these results may not extend directly to extremely large, opaque, or differentially aligned systems without adaptation.

These limitations reflect the current boundaries of evidence—not fundamental flaws. Each has a path toward deeper validation or methodological extension.

5. Discussion

5.1. What this paper covers

This paper consolidates and extends a series of empirical studies into the geometry of inference in transformer-based language models. Drawing from CI01, CI02, and CI03, we have unified the methodological

pipeline, formalized the geometric metrics (e.g. curvature, salience, and surface area), and presented a structured interpretation of how these metrics reflect the internal dynamics of semantic processing. This work introduces a consistent framework—Curved Inference—for analyzing model behavior in terms of trajectory structure within the residual stream.

5.2. What this contributes

This work contributes a new, model—native approach to interpretability—grounded in geometry rather than output attribution. Rather than asking which tokens caused a prediction, we examine how semantic content bends, diverges, and persists within the model's internal state. This trajectory-first view complements existing methods such as attention maps, probing, or activation patching.

Key contributions include:

a. a falsifiable, reproducible geometric framework for studying inference;
b. empirical evidence that semantic perturbations produce structured internal curvature;
c. a clear linkage between residual geometry and latent capabilities (e.g. deception and self-modeling), including evidence that residual curvature is necessary for self-modeling behavior in at least one model family (CI03); and
d. tools and metrics that generalize across multiple prompt types, domains, and models.

This represents a shift from static interpretability toward process-based interpretability—treating inference not only as a jump to output, but also as a traceable computation through space.

A natural next step is to compare residual-geometry signals directly with attention-based attribution, probe performance, and SAE-derived features on shared benchmarks—this lies beyond the scope of the present guide but would help place Curved Inference quantitatively among existing interpretability tools.

5.3. Why this matters

Understanding the structure of internal computation in LLMs is critical for advancing safety, alignment, and transparency. This work offers:

a. a lens to observe meaning formation as a movement through representation space;
b. an avenue for detecting latent or suppressed behaviors before they surface; and
c. a model-aligned interpretability technique that does not rely on external classifiers or assumed ground truth labels.

As LLMs are increasingly used in high-stakes or open-ended contexts, tools that reveal how decisions evolve internally become essential—not only for debugging, but also for understanding model intent, generalization, and limitations.

5.4. Next steps

There are several directions for continued research:

a. Scale testing: applying this method to larger models (e.g. >3b) to observe scaling trends
b. Task diversity: extending analysis to tasks involving reasoning, planning, or more complex multi-turn dialogue
c. Theory development: formalizing curvature signatures associated with specific generative behaviors (e.g. self-correction and intent tracking)
d. Tooling refinement: building more accessible, open-source packages for real-time or large-scale geometric analysis

Each of these directions would help further establish the value—and the limits—of geometric interpretability.

5.5. How to replicate or falsify this work

This paper and the experiments it is based upon were created with falsifiability in mind. All experiments used open-weight models and public scripts. To replicate or test the findings:

1) use the capture and metric scripts from the CI01-CI03 repositories (see Section 2.3);
2) begin with minimal prompt variants to verify curvature alignment;
3) reproduce the sleeper prompt analysis from CI02 using the same trigger-free vs. triggered pairs;
4) attempt curvature suppression or similar using SFT as in CI03; and
5) adapt this framework to define and conduct your own Curved Inference based experiments.

We encourage readers to attempt reproduction across domains, model families, and prompt classes—and to report both confirmations and contradictions. If curved inference is a robust lens on model behavior, it should be extensible and falsifiable in equal measure.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The author declares that he has no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in GitHub at https://github.com/robman/FRESH-model/tree/main/benchmarks/curved-inference.

Author Contribution Statement

Rob Manson: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., …, & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

[2] Manson, R. (2025). *Curved inference: Lab reports, scripts and full papers for 3 experiments*. Retrieved from: https://github.com/robman/FRESH-model/tree/main/benchmarks/curved-inference

[3] Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., …, & Seifert, C. (2023). From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. *ACM Computing Surveys*, *55*(13s), 1–42. https://doi.org/10.1145/3583558

[4] Zubiaga, A. (2024). Natural language processing in the era of large language models. *Frontiers in Artificial Intelligence*, *6*, 1350306. https://doi.org/10.3389/frai.2023.1350306

[5] Jain, S. & Wallace, B. (2019). Attention is not explanation. In *Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, *1*, 3543–3556. https://doi.org/10.18653/v1/N19-1357

[6] Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, *48*(1), 207–219. https://doi.org/10.1162/coli_a_00422

[7] Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., & Shieber, S. (2020). Investigating gender bias in language models using causal mediation analysis. *Advances in Neural Information Processing Systems*, *33*, 12388–12401.

[8] Zheng, Z., Wang, Y., Huang, Y., Song, S., Yang, M., Tang, B., …, & Li, Z. (2025). Attention heads of large language models. *Patterns*, *6*(2), 1–20. https://doi.org/10.1016/j.patter.2025.101176

[9] Gao, T., Yao, X., & Chen, D. (2021). SimCSE: Simple contrastive learning of sentence embeddings. In *Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1%2F2021.emnlp-main.552

[10] Li, Y., Michaud, E. J., Baek, D. D., Engels, J., Sun, X., & Tegmark, M. (2025). The geometry of concepts: Sparse autoencoder feature structure. *Entropy*, *27*(4), 344. https://doi.org/10.3390/e27040344

[11] Ranaldi, L. (2025). Survey on the role of mechanistic interpretability in generative AI. *Big Data and Cognitive Computing*, *9*(8), 193. https://doi.org/10.3390/bdcc9080193

[12] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

[13] Li, R., Zhao, X., & Moens, M. F. (2022). A brief overview of universal sentence representation methods: A linguistic view. *ACM Computing Surveys*, *55*(3), 1–42. https://doi.org/10.1145/3482853

[14] Asudani, D. S., Nagwani, N. K., & Singh, P. (2023). Impact of word embedding models on text analytics in deep learning environment: A review. *Artificial Intelligence Review*, *56*(9), 10345–10425. https://doi.org/10.1007/s10462-023-10419-1

[15] Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., & Liu, Y. (2024). Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, *568*, 127063. https://doi.org/10.1016/j.neucom.2023.127063

[16] Kim, H., & Jung, Y. (2025). Entropy-guided KV caching for efficient LLM inference. *Mathematics*, *13*(15), 2366. https://doi.org/10.3390/math13152366

[17] Zhang, B., & Sennrich, R. (2019). Root mean square layer normalization. *Advances in Neural Information Processing Systems*, *32*. https://doi.org/10.5167/UZH-177483

[18] MacDiarmid, M., Maxwell, T., Schiefer, N., Mu, J., Kaplan, J., Duvenaud, D., …, & Hubinger, E. (2024). Simple probes can catch sleeper agents. *Anthropic Research Updates*.

[19] Wang, L., Chen, S., Jiang, L., Pan, S., Cai, R., Yang, S., & Yang, F. (2025). Parameter-efficient fine-tuning in large language models: A survey of methodologies. *Artificial Intelligence Review*, *58*(8), 227. https://doi.org/10.1007/s10462-025-11236-4

[20] Baysan, M. S., Uysal, S., İşlek, İ., Çığ Karaman, Ç., & Güngör, T. (2025). LLM-as-a-Judge: Automated evaluation of search query parsing using large language models. *Frontiers in Big Data*, *8*, 1611389. https://doi.org/10.3389/fdata.2025.1611389

[21] Xu, N., Zhang, Q., Du, C., Luo, Q., Qiu, X., Huang, X., & Zhang, M. (2025). Revealing emergent human-like conceptual representations from language prediction. *Proceedings of the National Academy of Sciences*, *122*(44), e2512514122. https://doi.org/10.1073/pnas.2512514122

[22] Kumar, S., Sumers, T. R., Yamakoshi, T., Goldstein, A., Hasson, U., Norman, K. A., …, & Nastase, S. A. (2024). Shared functional specialization in transformer-based language models and the human brain. *Nature Communications*, *15*(1), 5523. https://doi.org/10.1038/s41467-024-49173-5

[23] Zheng, Z., Wang, Y., Huang, Y., Song, S., Yang, M., Tang, B., …, & Li, Z. (2025). Attention heads of large language models. *Patterns*, *6*(2). https://doi.org/10.1016/j.patter.2025.101176

[24] Ranaldi, L. (2025). Survey on the role of mechanistic interpretability in generative AI. *Big Data and Cognitive Computing*, *9*(8), 193. https://doi.org/10.3390/bdcc9080193

[25] Gandhi, K., Fränken, J. P., Gerstenberg, T., & Goodman, N. (2023). Understanding social reasoning in language models with language models. *Advances in Neural Information Processing Systems*, *36*, 13518–13529.

How to Cite: Manson, R. (2026). Curved Inference: A Guide to Geometric Interpretability. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62027102