

The Curved Spacetime of Transformer Architectures

Riccardo Di Sipio¹ Jairo Diaz-Rodriguez^{2*} Luis Serrano³

¹Dayforce, H.C.M. ²Department of Mathematics and Statistics, York University, Canada
³Serrano Academy

November 6, 2025

Abstract

We present a geometric framework for understanding Transformer-based language models, drawing an explicit analogy to General Relativity. Queries and keys induce an effective metric on representation space, and attention acts as a discrete connection that implements parallel transport of value vectors across tokens. Stacked layers provide discrete time-slices through which token representations evolve on this curved manifold, while backpropagation plays the role of a least-action principle that shapes loss-minimizing trajectories in parameter space. If this analogy is correct, token embeddings should not traverse straight paths in feature space; instead, their layer-wise steps should bend and reorient as interactions mediated by embedding space curvature. To test this prediction, we design experiments that expose both the presence and the consequences of curvature: (i) we visualize a curvature landscape for a full paragraph, revealing how local turning angles vary across tokens and layers; (ii) we show through simulations that excess counts of sharp/flat angles and longer length-to-chord ratios are not explainable by dimensionality or chance; and (iii) inspired by Einstein’s eclipse experiment, we probe deflection under controlled context edits, demonstrating measurable, meaning-consistent bends in embedding trajectories that confirm attention-induced curvature.

1 Introduction

The astonishing performance of large language models (LLMs) has sparked renewed interest in the structures they form internally [5, 6]. The path to this moment began with Bengio’s neural probabilistic language model [4], which demonstrated that predicting words in context could generate continuous vector embeddings, replacing symbolic representations with distributed ones. Word2Vec [19], GloVe [21], and ELMo [22] turned embeddings into the currency of natural language processing, while Transformers [26] and successors such as BERT [9] and GPT [23] transformed these static vectors into contextualized flows across layers.

This echoes a trajectory in physics. Newton’s gravity invoked “action at a distance”, effective but conceptually unsettling. Einstein replaced it with geometry in his theory of General Relativity (GR): masses curve spacetime, and free trajectories follow geodesics [7, 10]. Likewise, in language models, cosine similarity has served as a proxy for relational force between words: effective yet mechanistically shallow. Recent work suggests that curved representational spaces may offer a deeper foundation [8, 15, 20].

*Supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), through grant DGECR-2022-04531.

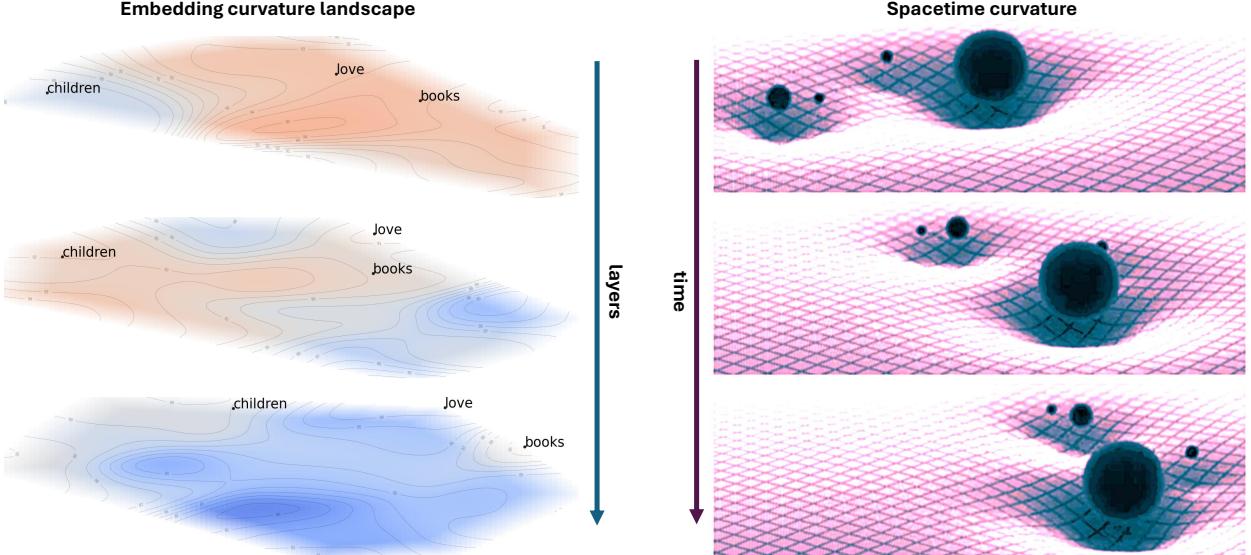


Figure 1: Graphical analogy between spacetime curvature and embedding curvature. On the right, spacetime curvature evolves over time according to the distribution of mass and energy. On the left, embedding curvature evolves across layers according to the learned weights.

	General Relativity	LLMs
Core Equations	Einstein’s Field Equations	Stack of Transformer layers; attention mechanism and loss minimization
Curvature	Geometry of spacetime	Geometry of embedding space
Source	Mass / energy distribution	Gradient of the loss function updates model weights
Time-evolution	Foliation into spacelike hypersurfaces governed by Einstein’s equations	Sequence of Transformer blocks that progressively refine internal representations

Table 1: Structural analogy between General Relativity and Transformer-based Language Models.

In this paper we make the analogy explicit. Query–key interactions define a local metric on representation space, attention acts as a connection that transports information across tokens, and stacked layers trace the discrete evolution of these representations through a curved manifold. In this view, each Transformer layer induces its own curvature: attention determines how tokens bend toward or away from one another, shaping the local geometry. As weights are updated through gradient descent, this curvature evolves, much like how the distribution of mass and energy reshapes spacetime curvature in GR. Thus, a token’s embedding follows a curved trajectory through successively adapting geometries, encoding progressively refined contextual meaning (Figure 1, Table 1).

This perspective moves beyond metaphor. Grounding Transformers in differential geometry offers an alternative way to think about how embeddings evolve and provides a means to test whether attention truly induces a curved representational geometry. To examine this, we introduce simple diagnostics that quantify curvature, *i.e.* a local measure based on turning angles between successive layer displacements and a global length-to-chord ratio capturing path elongation. These

metrics allow us to ask whether the trajectories traced by token embeddings behave as if they inhabit a curved manifold, as our analogy predicts. Empirically, we observe systematic curvature that is unlikely to arise by chance from high dimensionality. Finally, we design an experiment inspired by Einstein’s 1919 eclipse test, which verified spacetime curvature through the deflection of starlight near the Sun. In our setting, contextual edits play the role of the gravitational source, and the resulting semantic deflections of token trajectories offer an embedding-space analogue of light bending, an empirical probe of the geometric analogy itself.

Summary of contributions. (i) We propose a geometric interpretation of Transformer representations, where attention acts as a discrete connection transporting information across a curved semantic manifold, producing layerwise token trajectories shaped by learned geometry. (ii) We test this analogy through experiments that visualize curvature, show it cannot be explained by dimensionality or chance, and include a contextual “deflection” test modeled after gravitational lensing, the phenomenon that revealed spacetime curvature, revealing consistent, meaning-dependent bends in embedding space.

2 Related Work

The geometry of embedding spaces has long been central to NLP. Early work by Bengio et al. [4] introduced neural probabilistic language models, opening the way to distributed representations. Subsequent approaches such as GloVe [21] and ELMo [22] established embeddings as core components of modern systems [9, 23], with cosine similarity in Euclidean space becoming the standard proxy for semantic closeness.

The limitations of flat geometry have since been recognized. Nickel and Kiela [20] showed that hyperbolic embeddings better capture hierarchical relations, motivating curved alternatives. More recently, Cho et al. [8] introduced mixed-curvature Transformers with learnable sectional curvature, while Ji [15] proposed a principled Riemannian framework for attention based on parallel transport. Kratsios et al. [2] further demonstrated that even compact Transformers can approximate universal metric embeddings. At a broader level, He et al. [12] argued for non-Euclidean geometries as a natural fit for linguistic and relational data.

These contributions examine curvature from empirical, architectural, and mathematical perspectives, but none explicitly connect it to the analogy with General Relativity. In particular, the view that semantic curvature arises dynamically from gradients in the training loss has not been developed. Our work extends this line of inquiry by making that connection explicit and by mapping Transformer mechanisms to differential geometric principles.

3 Mathematical Foundations of Transformer Geometry

Transformer models, introduced in *Attention Is All You Need* [26] and extended into BERT [9] and GPT [23], learn contextual embeddings that evolve across stacked layers. We interpret this process geometrically: queries and keys induce an effective metric, attention acts as a connection that governs transport, and layers correspond to discrete “time steps” along geodesic-like trajectories in semantic space. Tab. 2 summarizes the formal analogy.

3.1 Query–Key–Value and Effective Metric

Each token x is projected into queries, keys, and values:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V, \quad (1)$$

with learned matrices $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$. The dot product between queries and keys governs how the model measures distances or angles between embeddings in the semantic space.

Hence, this bilinear form defines an *effective metric*:

$$g_{ij} := q_i^\top k_j = x_i^\top (W^Q)^\top W^K x_j. \quad (2)$$

This effective metric is in general different from an identity matrix $\text{diag}(1, 1, \dots, 1)$ and thus sets the stage for defining curvature. Although g_{ij} is not a Riemannian metric in the strict differential-geometric sense, being generally non-symmetric and not guaranteed positive-definite, it serves as the operational metric within the Transformer: all relational structure, attention, and contextual weighting derive from this kernel. This discrete construction does not replace the continuous Riemannian formalism but serves as its computational analogue, the way the Transformer operationalizes distance and orientation between semantic points in representation space.

3.2 Attention as a Connection

In Transformer networks, attention weights determine how much of each token contributes to updating another token’s representation. Scaled dot-product attention computes:

$$\text{Attention}(Q, K, V)_i = \sum_j \hat{\alpha}_{ij} W^V x_j = \sum_j \hat{\alpha}_{ij} V_j, \quad (3)$$

$$\hat{\alpha}_{ij} = \text{softmax}\left(\frac{Q_i \cdot K_j}{\sqrt{d_k}}\right). \quad (4)$$

The attention weights $\hat{\alpha}_{ij}$ determine how information from representation $v_j = W^V x_j$ is transported to position i . In Riemannian geometry, the Christoffel symbols (connection coefficients) Γ_{ij}^k describe how basis vectors change as one moves across the manifold, formally specifying how a vector is parallel-transported along coordinate direction j from point i .

In the Transformer, the indices i and j refer to tokens rather than coordinates, and the feature index k is contracted within the value projection W^V . Thus, the attention operator $\hat{\alpha}_{ij} W^V$ acts as a discrete analogue of a connection: it defines how feature vectors are mixed and transported along edges of the token graph, rather than how coordinate bases change on a continuous manifold. Full derivations connecting attention to Γ_{ij}^k appear in Appendix A and B.

3.3 Parallel Transport and Geodesics

In differential geometry, parallel transport is defined by a connection Γ_{jk}^i that determines how vectors change as they move along the manifold. The geodesic equation,

$$\frac{d^2 x^k}{dt^2} + \Gamma_{ij}^k \frac{dx^i}{dt} \frac{dx^j}{dt} = 0 \quad (5)$$

describes curves whose tangent vectors are parallel-transported along themselves.

In a Transformer, attention defines an analogous discrete transport between tokens. The operator $\hat{\alpha}_{ij}W^V$ specifies how representations at point j of the semantic manifold contributes to the updated representation at point i within the same layer. The output projection W^O acts as a transition maps between layers. Thus, the trajectory of a token embedding through Transformer layers approximates a discrete geodesic: each layer corresponds to a “tick” in semantic time.

By substituting the corresponding connection operator $\hat{\alpha}_{ij}W^V$ and replacing time derivatives with finite differences into the geodesic equation, we obtain a layer-discretized evolution rule.

Defining the hidden state at layer ℓ and the representation at the next layer $(\ell+1)$:

$$h_i^{(\ell)} = \sum_j \hat{\alpha}_{ij}^{(\ell)} W_{(\ell)}^V x_j^{(\ell)}, \quad (6)$$

$$x_i^{(\ell+1)} = x_i^{(\ell)} + W_{(\ell)}^O h_i^{(\ell)}, \quad (7)$$

and the discrete velocity and acceleration:

$$\dot{x}_{(\ell)}^i := x_{(\ell+1)}^i - x_{(\ell)}^i, \quad (8)$$

$$\ddot{x}_{(\ell)}^i := x_{(\ell+1)}^i - 2x_{(\ell)}^i + x_{(\ell-1)}^i \quad (9)$$

then the discrete velocity $\dot{x}_{(\ell)}^i$ becomes:

$$\dot{x}_{(\ell)}^i = x_{(\ell+1)}^i - x_{(\ell)}^i = W_{(\ell)}^O \sum_j \hat{\alpha}_{ij}^{(\ell)} W^V x_j^{(\ell)} \quad (10)$$

$$= \sum_j \left(W_{(\ell)}^O \hat{\alpha}_{ij}^{(\ell)} W^V \right) x_j^{(\ell)} \quad (11)$$

$$= \sum_j \Gamma_{ij}^{(\ell)} x_j^{(\ell)} \quad (12)$$

determining the discrete connection coefficients $\Gamma_{ij}^{(\ell)}$. Making use of a mid-point approximation (i.e $\Gamma_{ij}^{(\ell+1)} \approx \Gamma_{ij}^{(\ell)}$), the geodesic equation

$$\ddot{x}_\ell^k + \Gamma_{ij}^k(\ell) \dot{x}_\ell^i \dot{x}_\ell^j = 0, \quad (13)$$

becomes

$$x_{(\ell+1)}^i - 2x_{(\ell)}^i + x_{(\ell-1)}^i \sum_j \Gamma_{ij}^{(\ell)} \left(x_{(\ell)}^i - x_{(\ell-1)}^j \right) = 0 \quad (14)$$

Thus each Transformer layer ℓ defines its own effective connection, and the forward pass can be seen as a discrete geodesic trajectory evolving through curved semantic space, in analogy with the ADM (Arnowitt–Deser–Misner) formalism of GR [3].

3.4 A semantic least action principle.

In general relativity, the Einstein–Hilbert action encodes the curvature of spacetime and its response to matter (Appendix C and [17, 18]). By analogy, we can define an effective action for language models in terms of their representational geometry and training objective:

$$S_{\text{LM}} = \sum_{\ell=1}^L \left(g_{ij}^{(\ell)} \dot{x}_{(\ell)}^i \dot{x}_{(\ell)}^j - \mathcal{L}_{\text{train}}(x_{(\ell)}; \theta) \right),$$

Table 2: A compact geometric summary of the Transformer layer.

Transformer component	Geometric analogue	Interpretation
W^Q, W^K	Metric tensor g_{ij}	Define the local geometry and pairwise inner products between token representations.
$q_i^\top k_j$	Covariant components of g_{ij}	Measure contextual similarity between tokens i and j , analogous to metric components in a chosen basis.
$W_{(\ell)}^O \hat{\alpha}_{ij}^{(\ell)} W^V$	Connection coefficients Γ_{ij}^k	Act as discrete connection weights that determine how information flows or is transported from token j to token i .
$W^V x_j$	Tangent-space vector v_j	Represent the quantity being transported, i.e the local semantic vector in the learned tangent basis.
$y_i = \sum_j \alpha_{ij} v_j$	Parallel transport of v_j	Performs a discrete parallel transport: aggregates and re-expresses the value vectors under the learned connection.
Stacked layers	Geodesic integration	Successive parallel transports across layers trace geodesic-like trajectories on the learned semantic manifold.

where $g_{ij}^{(\ell)}$ is the effective metric induced by query–key interactions at layer ℓ , \dot{x}_ℓ^i denotes the discrete update of token embeddings across layers, and $\mathcal{L}_{\text{train}}$ is the loss functional that couples geometry to data. A full derivation is outlined in App. D.

Backpropagation can then be interpreted as the variational procedure that extremizes this action, ensuring that forward trajectories x_ℓ evolve along paths that minimize training loss. In this analogy, the dataset itself plays the role of “mass-energy”: its distribution acts as the source term shaping curvature in embedding space. Thus, just as in general relativity matter tells spacetime how to curve, data tells language models how to bend their representational manifold, and the resulting dynamics follow a least-action principle.

However, some words of caution are warranted in this context. In our setting, the layerwise activations x_ℓ are not independent coordinates that can be varied continuously, but functions of the model parameters θ . In the representation space, there exist paths x_ℓ that the model will never traverse, because no combination of weights θ generates them. Accordingly, the functional S_{LM} should be understood as an action over parameter-induced trajectories $x_\ell(\theta)$ in representation space, rather than over arbitrary paths as it is assumed in Riemann geometry. Extremizing S_{LM} with respect to θ corresponds to adjusting the weights so as to minimize both the “kinetic” cost of layer-to-layer changes (measured by the effective metric g_{ij}) and the “potential” cost given by the training loss. In this sense, backpropagation computes the gradient of an action functional defined on the weight manifold, with $g_{ij}^{(\ell)}$ acting as a *pull-back metric* linking parameter changes to geometric motion in the embedding space.

3.5 Multi-Head Attention as an Atlas of Charts

Each attention head defines its own (W^Q, W^K, W^V) projections, effectively a separate chart on the manifold. Just as multiple coordinate charts form an atlas in differential geometry to avoid singularities, multi-head attention provides overlapping local views. The output projection W^O acts as a transition map, integrating these views into a global representation.

3.6 Theoretical summary and experimental observables

The framework developed above treats attention as a discrete connection and defines curvature as the deviation of token trajectories from locally geodesic flow.

In practice, the true curvature of the representation manifold cannot be observed directly, since the manifold itself is implicit in the model’s parameters. Instead, we estimate it through observable proxies derived from embedding trajectories: the turning angle between successive layer displacements (local curvature) and the length-to-chord ratio measuring global deviation. These metrics provide experimentally accessible traces of the underlying geometry.

The goal of the experiments is therefore not to measure curvature directly but to test whether the observed embedding dynamics require a curved geometric explanation. If the trajectories of representations follow the same relational patterns that characterize geodesic deviation, curvature becomes the natural framework to describe them. In this sense, the curvature proxies serve as indirect evidence of the manifold’s shape, translating abstract geometric principles into measurable effects within Transformer activations.

4 Experiments

In this section, we present the results of three experiments aiming at demonstrating the curvature of the embeddings.

To quantify the curvature of token trajectories in representation space we introduce the following concept:

Local curvature (turning angle). Given embeddings x_{i-1} , x_i , and x_{i+1} of the same token at consecutive layers, we define step vectors $\Delta_i = x_i - x_{i-1}$ and $\Delta_{i+1} = x_{i+1} - x_i$. The turning angle θ_i is:

$$\theta_i = \arccos \left(\frac{\Delta_i \cdot \Delta_{i+1}}{\|\Delta_i\| \|\Delta_{i+1}\|} \right). \quad (15)$$

In differential geometry, curvature is formally defined as the rate of change of a tangent vector with respect to arc length, $\kappa(s) = \|dT/ds\|$. Here, the tangent at layer i is approximated by Δ_i , and θ_i serves as a discrete proxy for local curvature: small angles indicate nearly straight trajectories, while large angles signal sharp bends. In high-dimensional embedding spaces, random vectors are almost orthogonal ($\approx 90^\circ$). Thus, deviations below 80° indicate alignment toward a straight path (flat angles), while values above 100° reflect strong reorientations (sharp angles).

4.1 Example of the Curvature Landscape of a Paragraph

We first illustrate how an entire paragraph evolves through the Transformer stack. For this experiment, we use the following text (69 words) as input to the BERT transformer model [9] (12 layers, hidden size 768, 12 attention heads, 110M parameters):

The library was filled with books and stories. Students gathered in the hall to read, while parents encouraged children to discover new worlds through pictures and exams. The community saw the place as more than a building: it was a meeting point, open to everyone, where older generations shared their knowledge with the young, and workshops supported writing skills.

We track each token embedding across layers, projected to two principal components via PCA, and analyze for local curvature using the turning angle measure defined in Sec. 4. This yields a geometric “landscape” where tokens act like pebbles carried downstream: some drift smoothly, while others bend sharply in response to contextual forces.

Curvature Heatmaps. Figure 2 shows the curvature proxy as a heatmap: token positions are shown in PCA space, with color denoting curvature relative to an average turning angle of 90 degrees. Interestingly, the highest curvature peak corresponds not to a semantically heavy word, but to the function word *to*. This suggests that even seemingly lightweight tokens can act as critical junctions in the representational flow, depending on context. Other high-curvature regions are observed near *books*, *parents*, and *filled*, while words such as *the*, *is*, and *of* remain closer to flat trajectories. The evolution of curvature across layers progresses smoothly, indicating an underlying continuous process that is being captured in “snapshots” at each time slice.

Semantic clustering. The projection also reveals two clear semantic clusters. One group (*parents*, *students*, *children*, *books*, *stories*, *exams*, *pictures*, *older*) revolves around education and learning. Another group (*hall*, *place*, *filled*, *meeting*, *open*, *city*, *library*, *community*) captures spaces of gathering and collective activity. These clusters highlight how curvature is not random but organizes tokens into coherent neighborhoods that reflect semantic themes.

Beyond surface projections, it is also instructive to view the evolution of embeddings as a *stacked foliation* across layers. In this view, each sheet corresponds to a Transformer layer, and the stack as a whole captures how semantic states propagate through depth. As shown in Fig. 3, tokens trace discretized trajectories through “semantic time.” The contrast between red (sharp turns) and blue (smooth transitions) reveals variations in local curvature and allows trajectories such as that of *books* to be followed across layers, in analogy to the foliation of spacetime in general relativity.

Interpretation. These visualizations illustrate that curvature is not uniformly distributed across tokens. Function words, often assumed to be semantically light, can generate sharp local bends, while content words produce smoother but still context-sensitive peaks. The landscape perspective therefore provides an intuitive, geometry-based tool for inspecting how context reshapes language representations, revealing unexpected contributors to semantic curvature.

4.2 Curvature Analysis Across Layers

We now study how token representations evolve across Transformer layers by quantifying their geometric curvature in embedding space. First, we define global curvature:

Global curvature (path length). For embeddings x_0, x_1, \dots, x_N , the trajectory length is the polyline arc: $L = \sum_{i=1}^N \|x_i - x_{i-1}\|$. This quantity captures the accumulated displacement as a token evolves across layers. In Riemannian geometry, the length of a curve $\gamma(s)$ is obtained by integrating the norm of its tangent vector, $L = \int \|\dot{\gamma}(s)\| ds$. Our discrete sum is a layer-wise analogue: if the

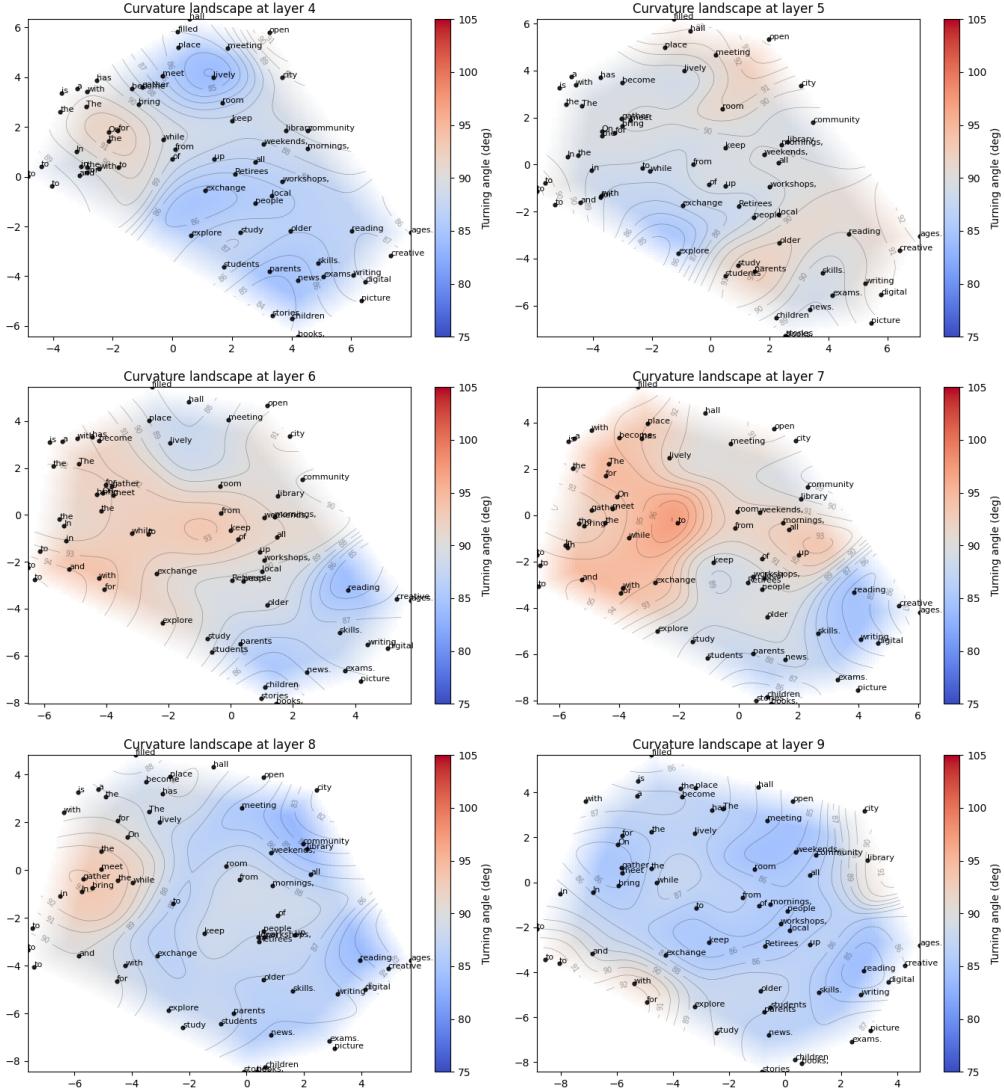


Figure 2: 2D curvature landscape for the same paragraph at six consecutive layers. Token positions are projected onto a PCA plane, with color encoding turning angle: blue areas indicate straighter motion ($< 90^\circ$), and red areas indicate sharper bending ($> 90^\circ$). This heatmap highlights regions of high contextual curvature where embeddings are more actively reshaped by attention.

manifold were flat and the update directions aligned, L would closely match the direct endpoint distance $\|x_N - x_0\|$. Longer paths relative to this displacement imply higher curvature, since the representation bends and reorients instead of following a straight geodesic through embedding space. To measure this relation we focus on the length-to-cord ratio

$$R = \frac{\sum_i \|\Delta_i\|}{\|x_N - x_0\|}.$$

Setup. In this experiment we analyze multiple token embedding trajectories through its sequence of turning angles θ_i and its overall length-to-chord ratio R . To test whether such directional changes reflect structured contextual transformations rather than random drift, we construct a *null model*. For each trajectory, the step lengths $\{\|\Delta_i\|\}$ are preserved, while directions are replaced by

Stacked curvature sheets across layers

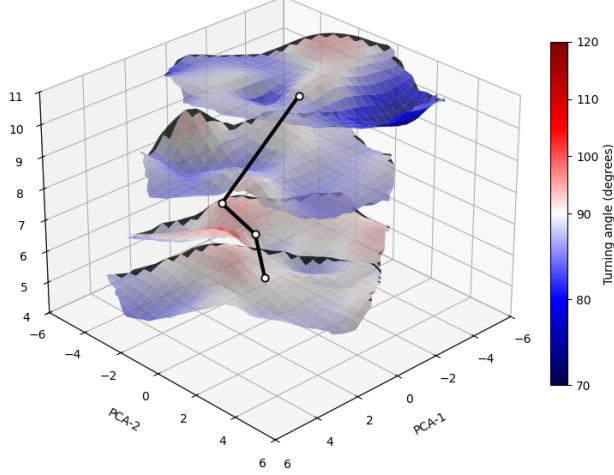


Figure 3: Foliated heatmap visualization of token embeddings across Transformer layers. Each wavy sheet corresponds to one layer, colored by the local turning angle (blue: $< 90^\circ$, red: $> 90^\circ$). The vertical axis represents discrete “semantic time” as the paragraph is processed layer by layer. The trajectory of tokens can be traced across sheets, illustrating how local curvature evolves throughout the model.

random unit vectors, generating 1000 synthetic trajectories per word. This control accounts for the anisotropy and scaling inherent in Transformer embeddings; if layer transitions were random, the transformer and null trajectories would be similar. See a detailed explanation on the construction of these null trajectories in Appendix E.2. We apply this analysis in two settings:

- First, we run a transformer model through each of 100 random sentences from the High-Quality English Sentences dataset [1], separately. From each, one noun or verb is selected via spaCy, and its embedding trajectories are extracted.
- Second, we apply the same procedure to the first paragraph of *One Hundred Years of Solitude* [11], treating each word as an individual trajectory within a shared paragraph context.

The paragraph setting tests whether curvature persists under strong inter-word dependencies, where contextualization evolves continuously across the discourse.

Results. Table 3 reports the total number of flat ($\theta < 80^\circ$) and sharp ($\theta > 100^\circ$) angles and the mean length-to-chord ratio across all trajectories in the first experiment, for both the observed Transformer trajectories and those generated under the null model, across multiple embedding architectures (Results for the second experiment are analogous and deferrd to Table 6 in Appendix). A larger number of flat and sharp angles indicates trajectories that follow purposefully directed paths rather than random high-dimensional artifacts, while a higher length-to-chord ratio (R) reflects stronger curvature, as embeddings bend and reorient instead of following straight geodesics through representation space. For all models, Transformer trajectories display markedly higher counts and R values than their null counterparts, providing exploratory evidence of embedding-space curvature. Appendix E presents the confirmatory statistical analysis, showing through pooled and paired hypothesis tests that these effects are highly significant and not attributable to chance. Together, these results demonstrate that contextual embeddings evolve along structured, non-linear trajectories, alternating between phases of alignment and sharp reorientation—offering geometric

evidence that contextualization in Transformers arises from directional bending of embeddings across layers rather than linear or random translation in feature space.

Table 3: Curvature metrics of 100 embedding trajectories across Transformer layers on multiple models (see Appendix F). For each model, we report counts of flat ($\theta < 80^\circ$) and sharp ($\theta > 100^\circ$) angles, and the average length-to-chord ratio R , compared to a randomized null model sampled 1000 times.

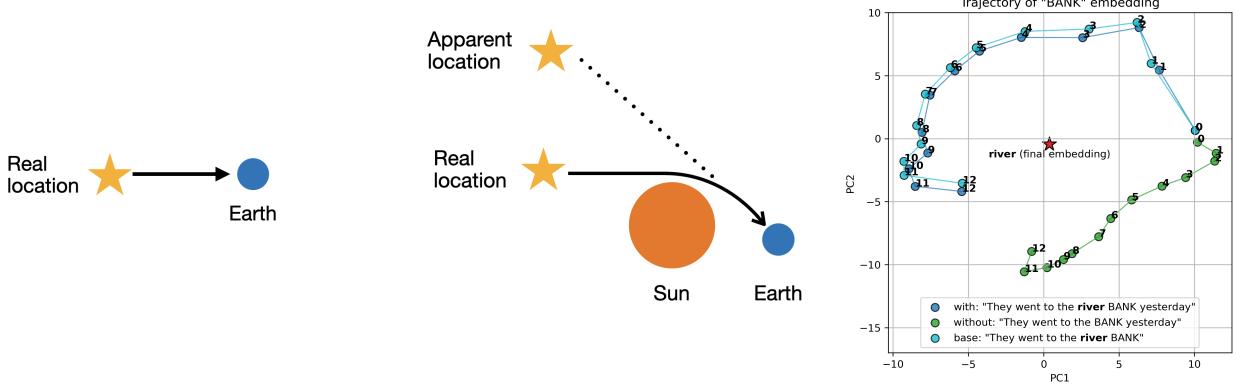
Size	Model	Angles	Transformers			Null model		
			flat	sharp	Average R	flat	sharp	Average R
small	DistilBERT	500	89	70	3.29	0.00	0.00	2.39
	DistilRoBERTa	500	193	0	3.73	0.00	0.00	2.28
	MiniLM	500	25	57	3.47	0.15	0.15	2.40
base	BERT	1100	25	20	5.29	0.00	0.00	3.42
	RoBERTa	1100	123	53	6.03	0.00	0.00	3.24
	MiniLM	1100	75	53	5.72	0.34	0.34	3.43
	DeBERTa	1100	21	236	2.71	0.00	0.00	2.63
	ALBERT	1100	44	53	4.30	0.00	0.00	3.23
large	BERT	2300	527	106	10.07	0.00	0.00	4.72
	RoBERTa	2300	563	23	5.05	0.00	0.00	4.62
	DeBERTa	2300	142	1	6.45	0.00	0.00	4.79
	ALBERT	2300	428	227	6.23	0.00	0.00	4.52

4.3 Context-Induced “Gravitational Lensing” of Meaning

Our last experiment is inspired on Einstein’s classic 1919 solar eclipse experiment. A central claim of General Relativity is that mass curves spacetime, and that this curvature can be empirically detected by measuring how light rays bend in the presence of a massive object. The classic 1919 solar eclipse experiment tested this: starlight passing near the sun appeared displaced relative to its expected straight-line trajectory, because the sun’s mass curved spacetime and deflected the path of the light (see the illustration in Fig. 4). Operationally, the test compares two trajectories of the *same* underlying signal—(i) with the massive body in view and (ii) without it—and attributes any systematic deflection to curvature induced by mass.

We adopt this logic in representation space in this. We treat an ambiguous token (e.g., “bank”) as the “light ray,” and a contextual disambiguator (e.g., “river”) as the “mass.” If context (“river”) exerts semantic force on the token (“bank”), it should not only change the token’s final embedding, it should bend its trajectory across layers. To test this, we construct controlled triples of sentences:

- **with:** a sentence in which a disambiguating modifier forces a specific sense of the target word (e.g., “They went to the *river* bank yesterday.”).
- **without:** the same sentence with that modifier removed (“They went to the bank yesterday.”), allowing the target word to revert to an alternative sense.
- **base:** a control edit in which we remove a token that is not the disambiguator (“They went to the river bank.”), to account for generic perturbation effects unrelated to the meaning of the target word.



(a) Light travels in a straight line without curvature. (b) Deflection near the Sun due to spacetime curvature. (c) Trajectory of “BANK” embedding in a Transformer.

Figure 4: Analogy between relativistic curvature and embedding curvature. (a) Without curvature, light travels in a straight line. (b) Near the Sun, spacetime curvature bends light, producing an apparent shift. (c) In language models, the token “BANK” follows a curved trajectory in representation space depending on its semantic context.

Here the disambiguator (“river”) plays the role of the gravitational source: if it truly “curves” representation space, then the trajectory of “bank” in the **with** sentence should deviate from its trajectory in the **without** sentence in a way that exceeds the deviation observed in the **base** condition.

Setup. For each sentence $s \in \{\text{with}, \text{without}, \text{base}\}$, we extract the layerwise embedding trajectory of the target token, $\{x_0^{(s)}, \dots, x_N^{(s)}\}$. We then compare trajectories between sentences within each triple. Given two trajectories a and b (e.g., **with** vs. **without**), we report four quantities:

1. **Final-layer separation.** The cosine distance between the final-layer embeddings,

$$d_{\text{final}}(a, b) = 1 - \frac{\langle x_N^{(a)}, x_N^{(b)} \rangle}{\|x_N^{(a)}\| \|x_N^{(b)}\|}, \quad (16)$$

which tests whether the two contexts drive the target token to distinct semantic endpoints.

2. **Layerwise separation.** The mean Euclidean distance between trajectories across depth,

$$d_{\text{layer}}(a, b) = \frac{1}{N+1} \sum_{i=0}^N \|x_i^{(a)} - x_i^{(b)}\|, \quad (17)$$

which reveals how persistently the trajectories diverge.

3. **Curvature divergence.** The mismatch in how the token bends across layers,

$$\Delta_{\text{curv}}(a, b) = \frac{1}{N} \sum_i \left(1 - \frac{\langle \Gamma_i^{(a)}, \Gamma_i^{(b)} \rangle}{\|\Gamma_i^{(a)}\| \|\Gamma_i^{(b)}\|} \right), \quad (18)$$

where $\Gamma_i^{(s)} = \Delta_{i+1}^{(s)} - \Delta_i^{(s)}$. Larger values indicate that the two trajectories do not merely end in different places; they curve differently along the way.

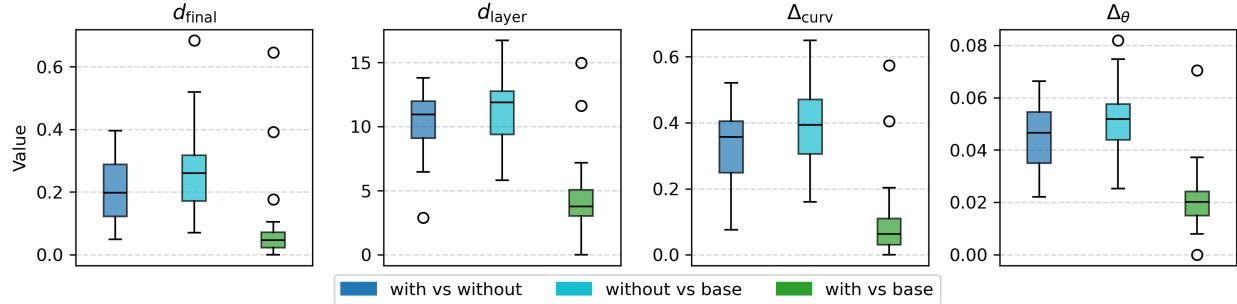


Figure 5: Trajectory divergence results for `bert-base-uncased`. Each subplot reports one of the four divergence metrics computed across 50 sentence triples. Boxplots show pairwise comparisons between sentence variants (`with` vs. `without`, `without` vs. `base`, `with` vs. `base`); higher values denote stronger divergence between embedding trajectories.

4. **Turning-angle gap.** The mean absolute difference in turning angles across layers,

$$\Delta_\theta(a, b) = \frac{1}{N} \sum_i |\theta_i^{(a)} - \theta_i^{(b)}|. \quad (19)$$

This measures local reorientation differences.

We compute these metrics for all three pairwise comparisons within each triple (`with` vs. `without`, `with` vs. `base`, `base` vs. `without`). The key test is whether `with` vs. `without` exhibits larger quantities than `with` vs. `base`. If so, then the contextual token (e.g., “river”) is not simply adding information downstream at the final layer; it is actively bending the trajectory of the ambiguous token throughout the network.

Results. Figure 5 summarizes the results for BERT [9]. Each of the four subplots corresponds to one trajectory-divergence metric, computed across 50 sentence triples. Within each subplot, the three boxplots represent pairwise comparisons between sentence variants: `with` vs. `without`, `without` vs. `base`, and `with` vs. `base`. Higher values indicate greater divergence between the corresponding embedding trajectories.

We observe that divergences are consistently larger for `with` vs. `without` and `without` vs. `base` than for `with` vs. `base`, suggesting that removing the disambiguating token alters the trajectory more strongly than adding it. This trend holds across both positional measures (d_{layer} , d_{final}) and geometric measures (Δ_{curv} , Δ_θ).

In other words, the presence of the disambiguator systematically changes not only where a representation ends, but how it moves through the network’s depth. This serves as a causal probe of curvature: contextual tokens act like localized mass, bending nearby embedding trajectories in a manner reminiscent of gravitational lensing. The addition or removal of a single semantically loaded word thus induces measurable deflections in the trajectory of an adjacent ambiguous token. Complete results across multiple embedding models appear in Table 7 in the Appendix.

5 Discussion

We have proposed that language models, particularly Transformers, operate in a latent geometric space whose curvature reflects the structure of language itself (Tab. 1). This geometry is not fixed

a priori, but emerges during training from the distribution of data and the gradients of the loss function.

A key idea is to interpret each Transformer block as a discrete “tick” in a semantic time-evolution. Each layer updates token representations via attention dynamics and learned weights, analogous to a time step in a discretized physical system. This invites comparison with the Hamiltonian formulation of GR, where spacetime is “foliated” into successive slices indexed by a time parameter [3]. Similarly, the sequence of Transformer layers may be viewed as a discrete foliation of semantic space: each slice reshapes the manifold while preserving coherence across layers.

This perspective suggests that interpretability might benefit from geometric tools. If each layer applies a differential transformation akin to parallel transport, then methods from differential geometry may help visualize and analyze how representations evolve. It also raises the possibility that depth should not be treated as an arbitrary hyperparameter, but as the number of steps in a meaningful dynamical process, perhaps governed by deeper variational principles.

We note a limitation of the analogy. In General Relativity, curvature and matter interact dynamically (back-reaction), whereas Transformer inference unfolds in a static geometry: the W^Q and W^K matrices define the manifold but remain fixed once trained. A natural direction for future work is to explore architectures where these matrices adapt during inference, introducing a form of geometric feedback. Such models could allow richer semantic generalization and dynamic contextualization.

A final remark is that the relativity of this framework lies not in the existence of curvature per se, but in its observability. We do not see the global structure of the semantic manifold; rather, we infer curvature locally, along the specific sentences and contexts we probe. In this sense, the analogy mirrors GR, where curvature is detected only through the motion of particles and light along particular paths.

6 Conclusions

We have outlined a geometric interpretation of Transformer-based language models, framing attention and query–key–value dynamics as connections that guide the transport of token representations through a curved semantic space. In this view, language understanding is not the result of static similarity but of trajectories shaped by context and curvature.

While prior work has hinted at non-Euclidean structures in embeddings, our framework connects these observations to the mathematical machinery of differential geometry which sits at the base of Einstein’s theory of General Relativity, offering a unifying perspective. This suggests new directions for model design: architectures that move beyond flat space and perhaps incorporate adaptive, curvature-aware mechanisms.

Ultimately, understanding in language models can be seen not as a lookup, but as a journey: a geodesic traced through a manifold bent by data, training, and context.

Reproducible research. The source code to replicate all experiments in this paper can be found on GitHub: <https://github.com/rdisipio/llm-curvature>

Acknowledgments

This work is dedicated to the memory of the late Prof. Silvio Bergia, who taught us Einstein’s theory of General Relativity and, more importantly, the language of gravity. A special note of

appreciation to Yoshua Bengio, not for offering us a place in his lab, but for the twists and turns that lead us here.

References

- [1] Agentlans (2024). High-quality english sentences. Hugging Face Dataset: <https://huggingface.co/datasets/agentlans/high-quality-english-sentences>. Version accessed on 2025-11-03; 1.7M+ English sentences, licensed under ODC-BY.
- [2] Anastasis Kratsios, Valentin Debarnot, I. D. (2024). Small transformers compute universal metric embeddings. *arXiv preprint arXiv:2402.09876*.
- [3] Arnowitt, R., Deser, S., and Misner, C. W. (1962). The dynamics of general relativity. In Witten, L., editor, *Gravitation: An Introduction to Current Research*, pages 227–265. Wiley. Reprinted in *Gen. Rel. Grav.* 40, 1997.
- [4] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.
- [5] Bommasani, R., Hudson, D. A., Adeli, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- [6] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.
- [7] Carroll, S. M. (2004). *Spacetime and Geometry: An Introduction to General Relativity*. Addison Wesley, San Francisco.
- [8] Cho, S., Cho, S., Sungwoo Park, H. L., Lee, H., and Lee, M. (2025). Curve your attention: Mixed-curvature transformers. *arXiv preprint arXiv:2505.67890*.
- [9] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- [10] Einstein, A. (1916). Die grundlage der allgemeinen relativitätstheorie. *Annalen der Physik*, 354(7):769–822.
- [11] García Márquez, G. (1967). *Cien años de soledad*. Editorial Sudamericana.
- [12] He, N., Liu, J., Zhang, B., Bui, N., Maatouk, A., Yang, M., King, I., Weber, M., and Ying, R. (2024). Foundation models should embrace non-euclidean geometries. *arXiv preprint arXiv:2404.11223*.
- [13] He, P., Gao, J., and Chen, W. (2021a). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.
- [14] He, P., Liu, X., Gao, J., and Chen, W. (2021b). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- [15] Ji, Z. (2025). Riemannformer: A framework for attention in curved spaces. *arXiv preprint arXiv:2506.12345*.

- [16] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*.
- [17] Lanczos, C. (1949). *The Variational Principles of Mechanics*. University of Toronto Press.
- [18] Landau, L. D. and Lifshitz, E. M. (1976). *Mechanics*, volume 1 of *Course of Theoretical Physics*. Pergamon Press, 3rd edition.
- [19] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- [20] Nickel, M. and Kiela, D. (2018). Poincaré glove: Hyperbolic word embeddings. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.
- [21] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- [22] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.
- [23] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI Blog*, 1(8). https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.
- [24] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP-IJCNLP*.
- [25] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- [26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.
- [27] Zhuang, L., Wayne, L., Ya, S., and Jun, Z. (2021). A robustly optimized BERT pre-training approach with post-training. In Li, S., Sun, M., Liu, Y., Wu, H., Liu, K., Che, W., He, S., and Rao, G., editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Appendix

A Introduction to the Geometry of Curved Manifolds

In Riemannian geometry, the *metric tensor* g_{ij} defines local distances and angles:

$$ds^2 = g_{ij} dx^i dx^j. \quad (20)$$

If an analytical form is unavailable, the metric can often be modeled implicitly via a kernel inner product:

$$ds^2 = \langle Kx, Ky \rangle. \quad (21)$$

Throughout, we follow Einstein's summation convention: repeated indices imply a sum, e.g.

$$A^i B_i \equiv \sum_i A^i B_i. \quad (22)$$

Tensors and Transformations. A *tensor* is a multilinear object whose components transform as

$$T'_{ij} = \frac{\partial x^m}{\partial x'^i} \frac{\partial x^n}{\partial x'^j} T_{mn}, \quad (23)$$

under a change of coordinates $\{x^i\} \mapsto \{x'^i\}$.

Affine Connection. An *affine connection* encodes how basis vectors vary across the manifold. This enables derivatives in curved spaces. Given two vector fields X and Y , the covariant derivative $\nabla_X Y$ measures the change of Y along X . In coordinates:

$$\nabla_i v^k = \partial_i v^k + \Gamma_{ij}^k v^j, \quad (24)$$

where Γ_{ij}^k are the *Christoffel symbols*. For the Levi-Civita connection (metric-compatible and torsion-free),

$$\Gamma_{ij}^k = \frac{1}{2} g^{kh} (\partial_i g_{jh} + \partial_j g_{ih} - \partial_h g_{ij}). \quad (25)$$

Parallel Transport. A vector v^i transported along a curve $\gamma(t)$ is said to be *parallel transported* if

$$\nabla_{\dot{\gamma}(t)} v(t) = 0, \quad (26)$$

or equivalently,

$$\frac{dv^i}{dt} + \Gamma_{jk}^i \frac{d\gamma^j}{dt} v^k = 0, \quad (27)$$

where a dot denotes differentiation with respect to the affine parameter t .

Geodesics. A *geodesic* is a curve that parallel-transport its tangent vector $v^i = \dot{\gamma}^i$:

$$\frac{d^2 \gamma^i}{dt^2} + \Gamma_{jk}^i \frac{d\gamma^j}{dt} \frac{d\gamma^k}{dt} = 0. \quad (28)$$

Curvature. The *Riemann curvature tensor* captures how parallel transport around a loop depends on the path:

$$R^i_{jkl} = \partial_k \Gamma^i_{jl} - \partial_l \Gamma^i_{jk} + \Gamma^i_{km} \Gamma^m_{jl} - \Gamma^i_{lm} \Gamma^m_{jk}. \quad (29)$$

By contraction one obtains the *Ricci tensor* $R_{jl} = R^i_{jil}$, and further contraction yields the *scalar curvature* $R = g^{jl} R_{jl}$, a single number at each point measuring how volumes deviate from flat space.

B From Attention Mechanism to Geometric Structures

This appendix provides the explicit mapping between Transformer operations and differential geometric objects introduced in Section 3.

B.1 Effective Metric

Starting from the query and key projections,

$$q_i = x_i W^Q, \quad k_j = x_j W^K, \quad (30)$$

we define the effective metric as

$$g_{ij} := q_i^\top k_j = x_i^\top (W^Q)^\top W^K x_j, \quad (31)$$

with inverse g^{ij} defined by $g^{ik} g_{kj} = \delta_j^i$.

B.2 Connection Coefficients

In Riemannian geometry, the Christoffel symbols are given by

$$\Gamma^i_{jk} = \frac{1}{2} g^{il} (\partial_j g_{kl} + \partial_k g_{jl} - \partial_l g_{jk}). \quad (32)$$

Using the effective metric, we can compute its derivatives w.r.t. token inputs x :

$$\frac{\partial g_{ij}}{\partial x_k} = (W^Q)^\top W^K \delta_{ik} x_j + (W^K)^\top W^Q \delta_{jk} x_i. \quad (33)$$

Substituting into the definition above yields the explicit expression for the Christoffel symbols in terms of W^Q and W^K :

$$\Gamma^i_{jk} = \frac{1}{2\sqrt{d}} \left[(Q^\top K)_{ik}^{-1} \left((W^Q)^\top W^K - (W^K)^\top W^Q \right) x_j (\delta_{jk} + \delta_{kj}) (Q^\top K)_{il}^{-1} (W^K)^\top W^Q x_l \right]. \quad (34)$$

Here the symmetric part governs context-dependent reshaping, while the antisymmetric part plays a role analogous to torsion.

B.3 Parallel Transport and Geodesics in Embedding Space

Let $\gamma(t)$ denote the trajectory of a token representation across layers, with

$$\gamma(t) \equiv x^{(t)} \in \mathbb{R}^d, \quad (35)$$

where t is a discrete layer index (approximating a continuous affine parameter).

The tangent vector is then the change in embeddings across layers:

$$\dot{\gamma}^i = \frac{d\gamma^i}{dt} \approx x_i^{(t+1)} - x_i^{(t)}. \quad (36)$$

The parallel transport condition reads:

$$\frac{dx^i}{dt} + \Gamma_{jk}^i \dot{\gamma}^j x^k = 0, \quad (37)$$

where $\dot{\gamma}^j$ encodes how the embedding evolves through the Transformer stack.

Substituting this into the geodesic equation gives:

$$\frac{d^2x^i}{dt^2} + \Gamma_{jk}^i \dot{\gamma}^j \dot{\gamma}^k = 0, \quad (38)$$

which states that the layerwise evolution of embeddings follows a geodesic curve with respect to the effective metric $g_{ij} = q_i^\top k_j$.

Let x_t^i be the token representation at layer t . Using finite differences for the layerwise “time”:

$$\dot{x}_t^i \approx x_{t+1}^i - x_t^i \quad (39)$$

$$\ddot{x}_t^i \approx x_{t+1}^i - 2x_t^i + x_{t-1}^i. \quad (40)$$

Each layer has its own projections

$$q_i^{(t)} = x_i^{(t)} W_t^Q, \quad k_j^{(t)} = x_j^{(t)} W_t^K, \quad (41)$$

$$G_{ij}^{(t)} = (q_i^{(t)})^\top k_j^{(t)} = (x_i^{(t)})^\top (W_t^Q)^\top W_t^K x_j^{(t)}, \quad (G^{(t)})^{-1} = (g^{ij})^{(t)}. \quad (42)$$

Replacing derivatives and the connection built from $G^{(t)}$ yields

$$\begin{aligned} x_{t+1}^i - 2x_t^i + x_{t-1}^i &+ \frac{1}{2} (G^{(t)})_{i\ell}^{-1} \left[((W_t^Q)^\top W_t^K) (x_{t+1}^j - x_t^j) (x_{t+1}^k - x_t^k) \delta_{j\ell} \right. \\ &\quad \left. + ((W_t^K)^\top W_t^Q) (x_{t+1}^j - x_t^j) (x_{t+1}^k - x_t^k) \delta_{k\ell} \right] = 0 \end{aligned} \quad (43)$$

This expression highlights that the “acceleration” of embeddings across layers (second finite difference) is balanced by curvature terms induced by the query and key projections, with the effective metric ($Q^\top K$) determining how updates couple across tokens.

C Least Action and Einstein’s Field Equations

In general relativity, the Einstein Field Equations (EFE) are not postulated directly, but can be derived from a variational principle. Specifically, they follow from the principle of least action applied to the *Einstein–Hilbert action*, which is the integral over spacetime of the Ricci scalar curvature R , weighted by the metric determinant $\sqrt{-g}$:

$$S = \frac{1}{16\pi G} \int R \sqrt{-g} d^4x + S_{\text{matter}}.$$

Here, G is Newton’s gravitational constant, and S_{matter} represents the action of any matter or energy present in the spacetime.

Varying this action with respect to the metric $g_{\mu\nu}$ yields the Einstein Field Equations:

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} = 8\pi G T_{\mu\nu},$$

where $R_{\mu\nu}$ is the Ricci tensor, R the Ricci scalar, and $T_{\mu\nu}$ the stress-energy tensor of matter.

This extremization principle is a deep geometrical statement: the actual geometry of spacetime is the one that extremizes the total curvature, balanced against the presence of matter. It echoes the structure we propose for language models: meaningful trajectories (e.g., in embedding space) arise from minimizing a contextual “action” shaped by loss and alignment.

D From Least Action to the Semantic Geodesic Equation

We formalize the “semantic least action” used in the main text by analogy with mechanics on a Riemannian manifold. Let $x(t) \in \mathbb{R}^d$ denote a token representation evolving across model depth (semantic time t), endowed with a metric $g_{ij}(x)$ (Sec. 3.1). Consider the action

$$S[x] = \int_{t_0}^{t_1} \left[\frac{1}{2} g_{ij}(x) \dot{x}^i \dot{x}^j + \lambda \mathcal{L}(x, t) \right] dt, \quad (44)$$

where the kinetic term measures motion w.r.t. g , and $\mathcal{L}(x, t)$ is a data/consistency potential (e.g. a local training loss); $\lambda > 0$ balances the two.

Euler–Lagrange and the geodesic-with-force form. The Lagrangian is $L(x, \dot{x}, t) = \frac{1}{2} g_{ij}(x) \dot{x}^i \dot{x}^j + \lambda \mathcal{L}(x, t)$. The Euler–Lagrange equations $\frac{d}{dt}(\partial L / \partial \dot{x}^i) - \partial L / \partial x^i = 0$ give

$$\frac{d}{dt}(g_{ij} \dot{x}^j) - \frac{1}{2} \partial_i g_{jk} \dot{x}^j \dot{x}^k + \lambda \partial_i \mathcal{L}(x, t) = 0. \quad (45)$$

Writing this with the Levi–Civita connection $\Gamma_{jk}^i = \frac{1}{2} g^{i\ell} (\partial_j g_{k\ell} + \partial_k g_{j\ell} - \partial_\ell g_{jk})$ yields the standard form

$$\ddot{x}^i + \Gamma_{jk}^i(x) \dot{x}^j \dot{x}^k = -\lambda g^{ij}(x) \partial_j \mathcal{L}(x, t). \quad (46)$$

When $\lambda = 0$ this reduces to the geodesic equation. For $\lambda > 0$, the loss acts as a potential, generating a force whose effect is mediated by the metric, so that updates follow the geometry of the semantic manifold.

Discrete (layer-wise) version. Let layers index a discrete time $t \in \{0, \dots, L\}$ with step $\Delta t = 1$ and x_t the representation at layer t . A second-order central difference for \ddot{x} and a mid-layer evaluation of Christoffel terms gives the discrete counterpart of (46):

$$x_{t+1} - 2x_t + x_{t-1} + \Gamma_{jk}^i(x_t) (x_t^j - x_{t-1}^j) (x_t^k - x_{t-1}^k) = -\lambda g^{ij}(x_t) \partial_j \mathcal{L}(x_t, t). \quad (47)$$

Rearranging gives a residual-style update:

$$x_{t+1} = 2x_t - x_{t-1} - \Gamma(x_t)[(x_t - x_{t-1}), (x_t - x_{t-1})] - \lambda g^{-1}(x_t) \nabla_x \mathcal{L}(x_t, t). \quad (48)$$

Thus, the depth-wise evolution comprises (i) a *geometric drift* driven by curvature via Γ , and (ii) a *metric-preconditioned* descent term $g^{-1} \nabla \mathcal{L}$, directly linking least action to the gradient signals used in backpropagation.

Instantiating g and Γ with Q/K. Using the effective metric from attention (Sec. 3.2),

$$g_{ij}(x) = q_i(x)^\top k_j(x) = x_i^\top (W^Q)^\top W^K x_j, \quad g^{ij} = (g_{ij})^{-1}, \quad (49)$$

the Christoffel symbols $\Gamma_{jk}^i(x)$ follow from g via the Levi–Civita formula, leading to the explicit Q/K expression derived in App. B. Substituting those into (48) yields the layer update used in the main text, making precise how attention-induced geometry (via W^Q, W^K) and loss gradients jointly govern the token trajectory across layers.

E Experimental details

This appendix specifies detailed setup and detailed statistically confirmatory results for the curvature experiments conducted over T embedding trajectories introduced in Section 4.2. In first experiment, each trajectory corresponds to the embedding evolution of a single randomly chosen noun or verb from one of 100 sentences sampled from the High-Quality English sentences dataset [1]. In second experiment, each trajectory corresponds to a single word from the first paragraph of *One Hundred Years of Solitude* [11]. Each trajectory consists of token embeddings $\{x_1, \dots, x_L\}$ across Transformer layers. The key quantities of interest are the turning angles between consecutive layer steps and the overall length-to-chord elongation ratio.

E.1 Curvature Statistics

For each trajectory we calculate the $L-1$ turning angles

$$\theta_i = \arccos\left(\frac{\Delta_i \cdot \Delta_{i+1}}{\|\Delta_i\| \|\Delta_{i+1}\|}\right), \quad \Delta_i = x_i - x_{i-1}, \quad i = 1, \dots, L-1.$$

The angles θ_i measure local curvature, while their frequency of deviation from the random orthogonality baseline ($\approx 90^\circ$) quantifies the amount of bending in embedding space. We summarize them by the combined tail count and the length-to-chord ratio:

$$C = \sum_{i=1}^{L-1} [\mathbf{1}\{\theta_i < 80^\circ\} + \mathbf{1}\{\theta_i > 100^\circ\}], \quad R = \frac{\sum_{i=1}^L \|\Delta_i\|}{\|x_L - x_0\|}.$$

Large C values indicate frequent flat and sharp directions expected from a curved space. While high R values indicate globally curved or elongated trajectories.

E.2 Construction of the Null

For a given trajectory with step magnitudes $s_i = \|\Delta_i\|$, the null model constructs a random polyline of identical step lengths but random directions:

$$u_i \sim \text{Unif}(\mathbb{S}^{d-1}) \quad (\text{e.g., } u_i = z_i / \|z_i\|, z_i \sim \mathcal{N}(0, I_d)), \quad (50)$$

$$\tilde{\Delta}_i = s_i u_i, \quad \tilde{x}_0 = 0, \quad \tilde{x}_k = \sum_{i=1}^k \tilde{\Delta}_i. \quad (51)$$

We generate $S = 1000$ null draws, and for each $s = 1, \dots, S$ compute

$$\tilde{\theta}_i^{(s)} = \arccos\left(\frac{\tilde{\Delta}_i^{(s)} \cdot \tilde{\Delta}_{i+1}^{(s)}}{\|\tilde{\Delta}_i^{(s)}\| \|\tilde{\Delta}_{i+1}^{(s)}\|}\right),$$

and the corresponding null statistics

$$\tilde{C}^{(s)} = \sum_{i=1}^{L-1} \left[\mathbb{1}\{\tilde{\theta}_i^{(s)} < 80^\circ\} + \mathbb{1}\{\tilde{\theta}_i^{(s)} > 100^\circ\} \right], \quad \tilde{R}^{(s)} = \frac{\sum_{i=1}^L \|\tilde{\Delta}_i^{(s)}\|}{\|\tilde{x}_L^{(s)} - \tilde{x}_0^{(s)}\|}.$$

Fixing $\{s_i\}$ controls for layerwise step magnitudes, while randomizing directions removes learned orientation. Any increase of C or R beyond their null distributions indicates systematic curvature induced by the Transformer's learned geometry.

E.3 Pooled tests

For trajectories $t = 1, \dots, T$, define pooled observed statistics and pooled null draws by summing counts and averaging R :

$$C_{\text{pool}}^{\text{obs}} = \sum_{t=1}^T C^{(t)}, \quad \tilde{C}_{\text{pool}}^{(s)} = \sum_{t=1}^T \tilde{C}^{(t,s)}, \quad (52)$$

$$\bar{R}^{\text{obs}} = \frac{1}{T} \sum_{t=1}^T R^{(t)}, \quad \tilde{\bar{R}}^{(s)} = \frac{1}{T} \sum_{t=1}^T \tilde{R}^{(t,s)}. \quad (53)$$

Define pooled differences

$$\Delta C_{\text{pool}}^{(s)} = C_{\text{pool}}^{\text{obs}} - \tilde{C}_{\text{pool}}^{(s)}, \quad \Delta \bar{R}^{(s)} = \bar{R}^{\text{obs}} - \tilde{\bar{R}}^{(s)}.$$

The pooled right-tailed p -values are

$$p_{\text{MC}}(C_{\text{pool}}) = \frac{1 + \#\{\Delta C_{\text{pool}}^{(s)} \leq 0\}}{S+1}, \quad p_{\text{MC}}(\bar{R}) = \frac{1 + \#\{\Delta \bar{R}^{(s)} \leq 0\}}{S+1}. \quad (54)$$

We test whether the expected differences between observed and null statistics are negative, corresponding to curvature-induced increases in the raw statistics. Specifically, for $p_{\text{MC}}(C_{\text{pool}})$ we test $H_0: \mathbb{E}[\Delta C_{\text{pool}}] \leq 0$ against $H_1: \mathbb{E}[\Delta C_{\text{pool}}] < 0$, and for $p_{\text{MC}}(\bar{R})$ we test $H_0: \mathbb{E}[\Delta \bar{R}] \leq 0$ against $H_1: \mathbb{E}[\Delta \bar{R}] < 0$. These one-sided alternatives correspond to the predicted direction of curvature: larger tail counts C and larger elongation ratios R under learned, non-random orientations relative to their isotropic nulls.

E.4 Paired Mean Tests (Across Trajectories)

Let the per-trajectory null means be

$$\tilde{\mu}_C^{(t)} = \frac{1}{S} \sum_{s=1}^S \tilde{C}^{(t,s)}, \quad \tilde{\mu}_R^{(t)} = \frac{1}{S} \sum_{s=1}^S \tilde{R}^{(t,s)}.$$

Define paired differences

$$D_C^{(t)} = C^{(t)} - \tilde{\mu}_C^{(t)}, \quad D_R^{(t)} = R^{(t)} - \tilde{\mu}_R^{(t)}.$$

With sample means \bar{D}_C, \bar{D}_R and standard deviations s_C, s_R , the paired t -statistics and (right-tailed) p -values are

$$t_C = \frac{\bar{D}_C}{s_C/\sqrt{T}}, \quad p_t(C) = 1 - F_{t_{T-1}}(t_C), \quad (55)$$

$$t_R = \frac{\bar{D}_R}{s_R/\sqrt{T}}, \quad p_t(R) = 1 - F_{t_{T-1}}(t_R), \quad (56)$$

where $F_{t_{T-1}}$ is Student's t CDF with $T-1$ degrees of freedom.

We test whether each trajectory's observed statistic exceeds its own null mean, indicating curvature beyond random orientation. For the paired differences $D_C^{(t)} = C^{(t)} - \tilde{\mu}_C^{(t)}$ and $D_R^{(t)} = R^{(t)} - \tilde{\mu}_R^{(t)}$, the hypotheses are $H_0: \mathbb{E}[D_C] \leq 0$ vs. $H_1: \mathbb{E}[D_C] > 0$ for the combined tail count and $H_0: \mathbb{E}[D_R] \leq 0$ vs. $H_1: \mathbb{E}[D_R] > 0$ for the elongation ratio. These one-sided alternatives correspond to the expectation that curvature in learned trajectories increases both the frequency of extreme turning angles (C) and the path-to-chord elongation (R) relative to their trajectory-specific null baselines.

E.5 Results

Tables 4 and 5 summarize the empirical results for both experiments. They report the average differences $\Delta C_{\text{pool}}^{(s)}$ and $\Delta \bar{R}^{(s)}$ and their corresponding pooled Monte-Carlo p -values (p_{MC}) for the extreme-angle count and length-to-chord ratio across multiple embedding models. For completeness, they also include the mean paired differences \bar{D}_C, \bar{D}_R and their corresponding paired t -test p -values (p_t). In all cases, both p_{MC} and p_t fall below 0.005, indicating that Transformer trajectories exhibit statistically significant deviations in both extreme-angle frequency and length-to-chord ratio relative to the random isotropic baseline. These results provide direct empirical evidence of curvature in the embedding space.

F Transformer models

We list below the Transformer models employed in our experiments, together with their corresponding Hugging Face identifiers.

- **Small models:**

- DistilBERT (`distilbert-base-uncased`; [25])
- DistilRoBERTa (`distilroberta-base`; [25, 27])
- MiniLM (`sentence-transformers/all-MiniLM-L6-v2`; [24])

- **Base models:**

- BERT (`bert-base-uncased`; [9])
- RoBERTa (`roberta-base`; [27])
- MiniLM (`sentence-transformers/all-MiniLM-L12-v2`; [24])
- DeBERTa (`microsoft/deberta-v3-base`; [13, 14])
- ALBERT (`albert-base-v2`; [16])

- **Large models:**

- BERT (`bert-large-uncased`; [9])
- RoBERTa (`roberta-large`; [27])
- DeBERTa (`microsoft/deberta-large`; [14])
- ALBERT (`albert-large-v2`; [16])

Table 4: Comparison of pooled and paired statistical tests across Transformer models and sizes for first experiment on Section 4.1. Columns under Pooled tests report the aggregate differences in combined counts of sharp and flat angles (ΔC_{pool}) and mean curvature ratio ($\Delta \bar{R}$) between Transformer trajectories and their null counterparts, together with their Monte Carlo p -values $p_{\text{MC}}(\cdot)$. Columns under Paired t-tests show the corresponding mean within-model differences (\bar{D}_C , \bar{D}_R) and their paired t -test p -values $p_t(\cdot)$. All reported p -values are smaller than 0.005 which indicate statistically significant deviations from the null, supporting that Transformer representations follow highly curved, non-random trajectories across layers.

Size	Model	Pooled tests				Paired t-tests			
		ΔC_{pool}	$p_{\text{MC}}(C_{\text{pool}})$	$\Delta \bar{R}$	$p_{\text{MC}}(\Delta \bar{R})$	\bar{D}_C	$p_t(\bar{D}_C)$	\bar{D}_R	$p_t(\bar{D}_R)$
small	DistilBERT	159.00	0.00	0.90	0.00	1.59	0.00	0.90	0.00
	DistilRoBERTa	193.00	0.00	1.45	0.00	1.93	0.00	1.45	0.00
	MiniLM	81.69	0.00	1.07	0.00	0.82	0.00	1.07	0.00
	DeBERTaV3	106.00	0.00	-0.10	1.00	1.06	0.00	-0.10	1.00
base	BERT	45.00	0.00	1.87	0.00	0.45	0.00	1.87	0.00
	RoBERTa	176.00	0.00	2.79	0.00	1.76	0.00	2.79	0.00
	MiniLM	127.31	0.00	2.30	0.00	1.27	0.00	2.30	0.00
	DeBERTaV3	257.00	0.00	0.08	0.00	2.57	0.00	0.08	0.00
	ALBERT	97.00	0.00	1.07	0.00	0.97	0.00	1.06	0.00
large	BERT	633.00	0.00	5.35	0.00	6.33	0.00	5.35	0.00
	RoBERTa	586.00	0.00	0.42	0.00	5.86	0.00	0.42	0.00
	DeBERTa	143.00	0.00	1.65	0.00	1.43	0.00	1.65	0.00
	ALBERT	655.00	0.00	1.71	0.00	6.55	0.00	1.71	0.00

Table 5: Comparison of pooled and paired statistical tests across Transformer models and sizes for second experiment on Section 4.1. Columns under Pooled tests report the aggregate differences in combined counts of sharp and flat angles (ΔC_{pool}) and mean curvature ratio ($\Delta \bar{R}$) between Transformer trajectories and their null counterparts, together with their Monte Carlo p -values $p_{\text{MC}}(\cdot)$. Columns under Paired t-tests show the corresponding mean within-model differences (\bar{D}_C , \bar{D}_R) and their paired t -test p -values $p_t(\cdot)$. All reported p -values are smaller than 0.005 which indicate statistically significant deviations from the null, supporting that Transformer representations follow highly curved, non-random trajectories across layers.

Size	Model	Pooled tests				Paired t-tests			
		ΔC_{pool}	$p_{\text{MC}}(C_{\text{pool}})$	$\Delta \bar{R}$	$p_{\text{MC}}(\Delta \bar{R})$	\bar{D}_C	$p_t(\bar{D}_C)$	\bar{D}_R	$p_t(\bar{D}_R)$
small	DistilBERT	196.00	0.00	1.28	0.00	1.59	0.00	1.28	0.00
	DistilRoBERTa	205.00	0.00	1.73	0.00	1.67	0.00	1.73	0.00
	MiniLM	128.63	0.00	2.29	0.00	1.05	0.00	2.29	0.00
	DeBERTaV3	123.00	0.00	0.10	0.00	1.00	0.00	0.10	0.00
base	BERT	161.00	0.00	2.21	0.00	1.31	0.00	2.21	0.00
	RoBERTa	250.00	0.00	3.49	0.00	2.03	0.00	3.49	0.00
	MiniLM	258.17	0.00	3.95	0.00	2.10	0.00	3.95	0.00
	DeBERTaV3	358.00	0.00	0.24	0.00	2.91	0.00	0.24	0.00
	ALBERT	208.00	0.00	1.25	0.00	1.69	0.00	1.25	0.00
large	BERT	959.00	0.00	5.65	0.00	7.80	0.00	5.65	0.00
	RoBERTa	649.00	0.00	0.57	0.00	5.28	0.00	0.57	0.00
	DeBERTa	402.00	0.00	2.03	0.00	3.27	0.00	2.03	0.00
	ALBERT	1419.00	0.00	3.76	0.00	11.54	0.00	3.76	0.00

Table 6: Curvature metrics of embedding trajectories from all words in the first paragraph of *One hundred years of solitude* across Transformer layers. For each model, we report counts of flat ($\theta < 80^\circ$) and sharp ($\theta > 100^\circ$) angles, and the average length-to-chord ratio R , compared to a randomized null model sampled 1000 times.

Size	Model	Angles	Transformers			Null model		
			flat	sharp	Average R	flat	sharp	Average R
small	DistilBERT	615	80	116	3.68	0.00	0.00	2.40
	DistilRoBERTa	615	195	10	4.05	0.00	0.00	2.32
	MiniLM	615	13	116	4.68	0.19	0.18	2.39
base	BERT	1353	82	79	5.63	0.00	0.00	3.42
	RoBERTa	1353	155	95	6.78	0.00	0.00	3.29
	MiniLM	1353	144	115	7.37	0.42	0.40	3.42
	DeBERTa	1353	22	336	3.02	0.00	0.00	2.79
	ALBERT	1353	195	13	4.53	0.00	0.00	3.28
large	BERT	2829	821	138	10.33	0.00	0.00	4.68
	RoBERTa	2829	598	51	5.19	0.00	0.00	4.63
	DeBERTa	2829	394	8	6.79	0.00	0.00	4.76
	ALBERT	2829	258	1161	8.56	0.00	0.00	4.80

Table 7: Average trajectory divergence metrics across 50 sentence triples for experiment in Section 4.3. For each model, we compare the target word’s layerwise path pairwise between the **with**, **without** and **base** variants. Reported metrics are d_{final} (final-layer distance), d_{layer} (mean layerwise distance), Δ_{curv} (curvature divergence), and Δ_{θ} (turning-angle gap). Higher values indicate stronger contextual bending. Standard deviations reported in parenthesis below each average.

size	model	with vs. without				without vs. base				with vs. base			
		d_{final}	d_{layer}	Δ_{curv}	Δ_{θ}	d_{final}	d_{layer}	Δ_{curv}	Δ_{θ}	d_{final}	d_{layer}	Δ_{curv}	Δ_{θ}
base	ALBERT	0.15 (0.07)	10.03 (2.58)	0.16 (0.05)	0.05 (0.01)	0.18 (0.09)	11.19 (3.25)	0.21 (0.09)	0.05 (0.03)	0.05 (0.08)	4.89 (4.04)	0.07 (0.09)	0.03 (0.04)
	BERT	0.20 (0.10)	10.22 (2.39)	0.32 (0.12)	0.05 (0.01)	0.26 (0.13)	11.31 (2.32)	0.39 (0.12)	0.05 (0.01)	0.07 (0.12)	4.47 (2.67)	0.10 (0.11)	0.02 (0.01)
	RoBERTa	0.05 (0.02)	7.53 (1.43)	0.39 (0.10)	0.06 (0.02)	0.05 (0.02)	7.87 (1.29)	0.42 (0.09)	0.06 (0.02)	0.01 (0.00)	2.50 (1.03)	0.07 (0.05)	0.02 (0.01)
	MiniLM	0.11 (0.06)	4.78 (1.27)	0.24 (0.10)	0.04 (0.01)	0.13 (0.07)	5.00 (1.24)	0.27 (0.10)	0.05 (0.01)	0.02 (0.02)	1.43 (0.54)	0.03 (0.02)	0.02 (0.01)
	ALBERT	0.09 (0.04)	9.16 (2.33)	0.27 (0.07)	0.09 (0.04)	0.13 (0.08)	10.34 (2.95)	0.33 (0.10)	0.10 (0.04)	0.06 (0.10)	5.43 (4.06)	0.15 (0.13)	0.06 (0.03)
	BERT	0.21 (0.10)	12.52 (2.79)	0.34 (0.10)	0.04 (0.01)	0.26 (0.15)	13.34 (2.70)	0.39 (0.10)	0.05 (0.02)	0.09 (0.15)	5.12 (2.22)	0.09 (0.07)	0.02 (0.02)
large	DeBERTa	0.28 (0.07)	23.83 (4.36)	0.52 (0.12)	0.05 (0.01)	0.30 (0.07)	24.80 (3.96)	0.56 (0.10)	0.05 (0.01)	0.06 (0.04)	8.40 (3.43)	0.12 (0.07)	0.02 (0.01)
	RoBERTa	0.02 (0.01)	12.31 (2.32)	0.44 (0.10)	0.06 (0.01)	0.02 (0.01)	13.07 (1.95)	0.48 (0.08)	0.07 (0.01)	0.00 (0.00)	4.52 (1.82)	0.09 (0.07)	0.03 (0.01)
	DistilBERT	0.12 (0.05)	6.84 (1.45)	0.24 (0.08)	0.06 (0.03)	0.16 (0.06)	7.52 (1.45)	0.30 (0.08)	0.06 (0.02)	0.03 (0.04)	2.91 (1.44)	0.06 (0.06)	0.02 (0.02)
	DistilRoBERTa	0.04 (0.02)	6.05 (1.26)	0.39 (0.11)	0.09 (0.03)	0.04 (0.02)	6.34 (1.05)	0.42 (0.09)	0.09 (0.04)	0.00 (0.00)	1.70 (0.81)	0.05 (0.04)	0.02 (0.01)
	DeBERTaV3	0.18 (0.09)	12.20 (2.38)	0.44 (0.11)	0.05 (0.02)	0.17 (0.07)	12.71 (1.97)	0.48 (0.09)	0.05 (0.02)	0.03 (0.02)	3.73 (2.09)	0.07 (0.07)	0.02 (0.01)
	MiniLM	0.11 (0.06)	4.44 (1.02)	0.22 (0.09)	0.04 (0.02)	0.13 (0.07)	4.61 (1.00)	0.25 (0.09)	0.04 (0.02)	0.02 (0.02)	1.35 (0.56)	0.03 (0.02)	0.01 (0.01)