

Riemannian Geometry for Pre-trained Language Model Embeddings

Szczepan Konior¹ Alexandre Quemy² Przemysław Kłoczek¹

Bartłomiej Sobieski^{3,4} Grégoire Cattan¹

¹IBM Automation and AI, Krakow, Poland ²Hother, Krakow, Poland

³University of Warsaw ⁴Centre for Credible AI, Warsaw University of Technology

{name.surname}@ibm.com

alexandre@hother.io

sobieski.bartlomiej.jan@gmail.com

Abstract

Understanding the geometric structure of pre-trained language model embeddings matters for interpretability and safety. We ask whether sentence-level classification signal lives in the Riemannian geometry of contextual token embeddings, and probe it by extracting per-token pullback metrics from a learned encoder’s analytical Jacobian and aggregating them with the Fréchet mean on the symmetric positive definite (SPD) manifold, we call this procedure Riemannian Mean Pooling (RMP). Across three datasets with non-trivial linguistic structure (CoLA, CREAK, RTE), RMP outperforms Euclidean mean pooling, while on FEVER-Symmetric, a benchmark constructed to remove annotation-driven lexical artifacts, the method correctly stays at chance. Ablations show that a randomly initialised encoder combined with Fréchet aggregation already beats Euclidean pooling before any training, localising the source of the gain to the geometric aggregation rather than to learned manifold structure. The trained encoder adds a further 1.7 to 2.2 AUC points, consistently across the signal-bearing datasets rather than on any one of them.

1 Introduction

Pre-trained language models achieve remarkable success in natural language understanding, yet the geometric structure of their internal representations remains an open challenge. Most analyses treat token embeddings as points in flat Euclidean space, but three lines of evidence motivate departing from that assumption. First, hierarchical structure in language admits lower-distortion embeddings in negatively curved spaces (Nickel and Kiela, 2017, 2018; Sala et al., 2018; Park et al., 2025) and is reflected in non-standard inner products on the embedding space (Park et al., 2024; Marks and Tegmark, 2024). Second, contextualized representations in BERT, ELMo, and

GPT-2 are strongly anisotropic, occupying narrow cones not invariant under Euclidean transformation (Ethayarajh, 2019; Kataiwa et al., 2025). Third, token embedding spaces may also fail to form smooth manifolds globally, even if local-manifold-like structure still supports geometric analysis at the neighbourhood level (Robinson et al., 2025).

This work investigates whether sentence-level classification signal lives in the Riemannian geometry of token embeddings. We address three questions: (1) Do embeddings exhibit local geometric structure accessible via pullback metrics? (2) Does this geometric structure carry signal beyond what flat Euclidean aggregation extracts? (3) Which component of a geometric aggregation pipeline is responsible for any observed gain: the encoder, the metric, or the aggregation?

We evaluate our approach, *Riemannian Mean Pooling* (RMP), on four diverse NLP tasks: fact verification (FEVER-Symmetric), textual entailment (RTE), grammatical acceptability (CoLA), and commonsense reasoning (CREAK). FEVER-Symmetric is included specifically as a negative control, since it was constructed to remove lexical and annotation artifacts (Schuster et al., 2019), and we expect any method relying solely on claim-surface geometry to perform at chance.

Any differentiable encoder yields a pullback metric through its Jacobian, but the metric’s geometric interpretability depends on the training objective. We use Intrinsic Green’s Learning (IGL) (Quemy, 2026)¹, where a closed-form kernel readout replaces the learned decoder, constraining the encoder to coordinates aligned with the smooth structure of the data manifold rather than with arbitrary nonlinear features. Our ablations (Section 5) test this choice by comparing

¹Reference implementation: <https://github.com/hotherio/intrinsic-green-learning>.

IGL against random and untrained encoders.

Contributions

- We instantiate a pipeline that extracts per-token pullback metrics from a learned encoder’s Jacobian, aggregates them on the SPD manifold via the Fréchet mean, and classifies in the tangent space at the population Riemannian mean.
- Across three signal-bearing datasets (RTE, CoLA, CREAK), Riemannian aggregation consistently outperforms Euclidean mean pooling under a controlled comparison on identical embeddings. On FEVER-Symmetric, included as a negative control, the method correctly stays at chance.
- A randomly initialised encoder combined with Fréchet aggregation already beats Euclidean pooling, localising the source of the geometric signal to the aggregation rather than to learned manifold structure. The trained encoder contributes a further 1.7 to 2.2 AUC points on every signal-bearing dataset.

2 Related work

Riemannian Methods in Neuroscience Riemannian geometry is a ubiquitous machine learning technique in brain-computer interfaces (BCIs) and signal processing (Congedo et al., 2017). Barachant et al. (2012) introduced the use of symmetric positive definite (SPD) matrices as a descriptor for electroencephalography time series, demonstrating that geometric methods can capture the intrinsic structure of signals better than Euclidean approaches. The riemannian minimum-distance-to-means (MDM) and tangent space (TS) classifiers have become ubiquitous tools for neurosciences application, still unbeaten even by the most recent deep learning algorithms (Chevallier et al., 2024), while remaining simple, deterministic, robust to noise, computationally efficient, and prone to transfer learning (Andreev et al., 2025). Beyond the BCI setting, SPD-manifold deep learning has matured: differentiable Fréchet-mean layers (Lou et al., 2020), Riemannian batch normalisation for SPD networks (Brooks et al., 2019), and Riemannian extensions of logistic regression (Chen et al., 2024) now provide standard components for end-to-end SPD pipelines, with recent surveys covering the area (Ju et al., 2024).

Geometric Approaches in NLP In modern NLP architectures, tokens are represented as vectors in a high-dimensional latent space. As tokens propagate through the network layers, their representations evolve inside the Residual Stream, the model’s dynamic internal state. This evolution is not purely linear or syntactic, but can be described as a structured trajectory whose geometry changes depending on the semantic context (Manson, 2025). Concrete instances of this structure have been demonstrated empirically: linear directions in LLM representations encode space, time, and other conceptual attributes (Gurnee and Tegmark, 2024), and architectural variants explicitly equip attention with non-Euclidean geometry, either via mixed-curvature transformers (Cho et al., 2023) or via parallel-transport-based attention on manifolds (Ji, 2025). Recent work has also proposed that the latent space itself can be modelled as a curved spacetime whose curvature reflects linguistic structure (Di Sipio et al., 2025).

Current approaches estimate local tangent spaces from k-nearest-neighbor (kNN) patches and derive curvature information from variations between neighboring tangent directions (e.g., Gao and Liang, 2011; Li, 2017). However, these methods often suffer from numerical instability and sensitivity to sampling density, with metric estimates degrading as the local sample size shrinks (Brand, 2003; Singer, 2006). A different style of approach first projects the data to a lower-dimensional space (for example via UMAP (McInnes et al., 2018)) and computes metrics in the projection, but UMAP preserves local topology rather than metric structure: distances after UMAP are unsuitable as approximations of distances on the original manifold, so subsequent geometric computations operate on a non-isometric image of the data. Recent work has also questioned whether token embeddings form a single connected manifold at all (Robinson et al., 2025). The question we ask here is whether aggregated token geometry nevertheless carries class signal at the sentence level, regardless of whether global manifold structure holds.

Intrinsic Dimension and Manifold Learning Sobieski et al. (2026) showed that local intrinsic dimension (LID) reveals hallucinations in diffusion models by identifying instabilities on model-induced manifolds. Their work treats hallucinations as geometric instabilities, proposing Intrinsic

Quenching to correct these anomalies. Similarly, [Damirchi et al. \(2026\)](#) introduced "Truth as a Trajectory," analysing LLM reasoning through layer-wise geometric metrics rather than static activations. By examining how representations evolve across layers, they uncover geometric invariants distinguishing valid reasoning from spurious behaviour. Beyond these layer-wise trajectory analyses, a body of recent work uses intrinsic dimension as a probe of how LLMs process information across layers: representations expand in early layers and compress in later layers ([Valeriani et al., 2023](#)), with characteristic per-layer profiles that include a high-dimensional abstraction phase in the middle layers ([Cheng et al., 2025](#)). ID has also been used to compare learning paradigms such as supervised fine-tuning and in-context learning ([Janapati and Ji, 2025](#)). Our work complements these perspectives.

Sentence Representation Learning Standard sentence-representation approaches such as mean pooling of token embeddings and the BERT [CLS] token implicitly assume flat Euclidean geometry. We replace arithmetic averaging with an SPD-manifold Fréchet aggregation. Recent work has also highlighted the entanglement of pooled sentence representations ([Tehenan et al., 2025](#)), motivating geometric alternatives.

3 Material & Method

3.1 Datasets

We evaluate on four binary-classification benchmarks. **FEVER-Symmetric** ([Schuster et al., 2019](#)) contains 316 fact-verification claims, debiased by Schuster et al. against the annotation artifacts of the original FEVER ([Thorne et al., 2018](#)) by pairing each claim with its negation against shared evidence. We include it as a *negative control*, since methods relying solely on claim-surface features cannot exceed chance by construction. **RTE** ([Wang et al., 2018](#)): 2,490 premise–hypothesis pairs for textual entailment. **CoLA** ([Warstadt et al., 2018](#)): 3,000 sentences labelled for grammatical acceptability. **CREAK** ([Onoe et al., 2021](#)): 3,000 commonsense statements requiring world knowledge to verify.

3.2 Baseline Method: Linear Probe

We use **Linear Probe** ([Alain and Bengio, 2017](#)) as the headline baseline, referring to this baseline as the Linear Probe throughout and reserving the

term *Euclidean mean pooling* for the underlying aggregation operation when contrasting it with the Riemannian alternative. The baseline operates entirely on internal states of the pre-trained model, isolating the contribution of geometric aggregation from any additional trainable parameters or external information sources.

Token embeddings and pooling. We extract token-level embeddings from the ninth hidden layer of BERT-base-uncased ([Devlin et al., 2019](#)), yielding 768-dimensional representations per token. The layer choice is motivated by prior work showing that upper layers encode more abstract semantic features while lower layers capture syntactic information ([Rogers et al., 2020](#)). For single-sentence datasets (FEVER, CoLA, CREAK) we encode the input directly. For the textual-entailment task (RTE) we concatenate premise and hypothesis with separator tokens ([CLS] [premise] [SEP] [hypothesis] [SEP]). Token embeddings are aggregated via arithmetic mean pooling:

$$\mathbf{v}_{\text{sentence}} = \frac{1}{n} \sum_{i=1}^n \mathbf{v}_i \in \mathbb{R}^{768}, \quad (1)$$

with $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ the token embeddings for a sentence with n tokens.

Classification. Pooled sentence representations are normalised and classified with a logistic regression using scikit-learn’s ([Pedregosa et al., 2011](#)) default hyperparameters (no regularisation, LBFGS solver, max-iter 1000). The same classification head is used for the Riemannian pipeline downstream of the Fréchet aggregation (Section 3.5), so performance differences attribute cleanly to the aggregation step.

3.3 Intrinsic Green’s Learning

Inverse-PDE objective. IGL ([Quemy, 2026](#)) is a framework for learning low-dimensional manifold representations through an inverse-PDE formulation. A small MLP encoder $\Psi_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^{d_{\max}}$ maps each input $\mathbf{v} \in \mathbb{R}^D$ to a latent code $\xi = \Psi_\theta(\mathbf{v})$. In the encoder’s coordinates, the target function (here, the input itself) is expressed as the solution of a linear PDE

$$Lu = s, \quad (2)$$

where L is a fixed operator (a Laplacian in our setup whose Green’s function is approximated by

a Gaussian kernel) and s is a learned source term.

Closed-form kernel readout. The PDE is solved in closed form via a Green’s-function kernel. In practice, at any latent point ξ , the reconstruction is a linear combination of kernel evaluations at K learned anchor points $\{\zeta_k\}_{k=1}^K$,

$$\hat{u}(\xi) = \sum_{k=1}^K w_k G(\xi, \zeta_k), \quad (3)$$

where G is the Gaussian kernel. The weights w are not learnable parameters. At every minibatch they are recomputed by ridge regression on the reconstruction targets (variable projection).

Encoder-only optimisation. IGL has no symmetric decoder. Because w is closed-form, gradient descent operates only on the encoder parameters θ , and the encoder must find coordinates in which the input admits a low-rank kernel decomposition. Those coordinates align with the smooth structure of the data manifold rather than with arbitrary nonlinear features a learned decoder could compensate for. As a byproduct, because Ψ_θ is a differentiable MLP, its Jacobian $J(\mathbf{v}) = \partial\Psi_\theta/\partial\mathbf{v}$ is analytically computable, giving direct access to the pullback metric at every input point.

Under the configuration used in this work (MLP encoder, $d_{\max}=64$, $K=128$ anchors, 4 Gaussian scales per factor), training converges to stable reconstruction. Full architectural details are deferred to Appendix A. The full pipeline (training top, inference bottom) is shown schematically in Figure 1. Representative validation-loss curves are shown in Appendix F.

3.4 Pullback metric extraction

After training the IGL encoder $\Psi_\theta : \mathbb{R}^{768} \rightarrow \mathbb{R}^{d_{\max}}$ on token-level BERT embeddings $\mathbf{v} \in \mathbb{R}^{768}$ (Section 3.3), we extract a local Riemannian metric at each token via the *pullback metric*. The intuition is that Ψ_θ maps the ambient embedding space to a lower-dimensional latent space. The pullback metric measures distances in the input space as they appear after this mapping, capturing how the encoder locally stretches or compresses different directions of the embedding space.

From Jacobian to metric. For an input $\mathbf{v} \in \mathbb{R}^{768}$, the Jacobian of the encoder is

$$J(\mathbf{v}) = \partial\Psi_\theta/\partial\mathbf{v} \in \mathbb{R}^{d_{\max} \times 768}, \quad (4)$$

computed in closed form by automatic differentiation through Ψ_θ . Pulling back the ambient Euclidean metric I_{768} through the encoder yields the co-metric on the latent manifold:

$$g^{-1}(\mathbf{v}) = J(\mathbf{v}) J(\mathbf{v})^\top \in \mathbb{R}^{d_{\max} \times d_{\max}}. \quad (5)$$

By construction $g^{-1}(\mathbf{v})$ is symmetric and positive semi-definite. To obtain a metric tensor that is strictly positive-definite (required for the Riemannian operations in Section 3.5), we add a small regulariser before inversion:

$$g(\mathbf{v}) = (g^{-1}(\mathbf{v}) + \epsilon I_{d_{\max}})^{-1}, \quad \epsilon = 10^{-6}, \quad (6)$$

and a further additive regularisation $\lambda I_{d_{\max}}$ with $\lambda = 10^{-2}$ to guarantee numerical stability across the full range of input tokens. The resulting $g(\mathbf{v}) \in \mathcal{S}_{++}^{d_{\max}}$ is the per-token SPD matrix used as input to the sentence-level aggregation.

Remark. *The expression pullback metric is an abuse of language, since $d_{\max} < 768$, Ψ_θ is a submersion, and (5) is not a metric but a co-metric, obtained by pushing the ambient I_{768} forward along Ψ_θ . The strict pullback would be the degenerate $J^\top J$ on $\mathbb{R}^{768}$. Restricting Ψ_θ to an estimated data manifold M of dimension $m < d_{\max}$ makes it an immersion and replaces (5) with $J P_M J^\top$. Our runs show that it does not improve RMP on any dataset, achieving the same or worse performance than the unrestricted metric. We report that in Appendix G.*

Why pullback rather than direct estimation? The two most direct alternatives, that is, neighbour-based metric estimation and projection-then-metric approaches (e.g. UMAP), suffer from sensitivity to local sampling density and non-isometric distortion, respectively (Section 2). The pullback through a trained encoder avoids both issues: the metric is computed analytically from J rather than estimated from neighbours, and its position dependence is a smooth function of the original 768-dimensional input rather than of a non-isometric projection of it. The sensitivity of neighbour-based estimation is measurable on these embeddings and has been reported in Appendix G. Pulling back an ambient metric through an encoder’s Jacobian follows Hauser and Ray (2017) and Arvanitidis et al. (2018). Recent extensions analyse discretisation of continuous inputs (Brandon et al., 2025) and score-based extraction of data-manifold geometry (Diepeveen et al., 2024).

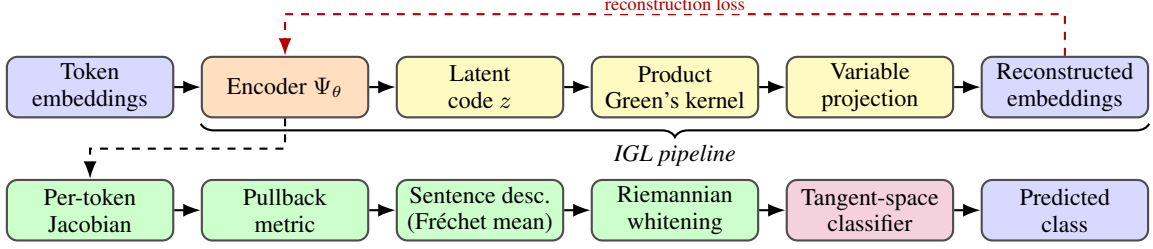


Figure 1: Overview of the pipeline. **Top row (training):** the encoder Ψ_θ is trained to reconstruct token embeddings (early stopping with patience=50, $\delta = 2 \times 10^{-5}$), mapping them to a low-dimensional latent code from which a product Green’s kernel with variable projection recovers the input. The reconstruction loss flows back to the encoder (dashed arrow). **Bottom row (inference):** once trained, the encoder’s Jacobian induces a pullback metric at each token. The per-token metrics are aggregated on the SPD manifold via the Fréchet mean to a sentence-level descriptor, preprocessed with Riemannian whitening, projected to the tangent space, and classified. The classification (inference) path carries no gradients and uses the encoder frozen post-training, and only the reconstruction (training) path is differentiable. The full architecture with formulas is given in Figure 3 of Appendix A.

Independence from the training loss. Because the pullback metric depends only on the encoder’s Jacobian rather than on its training loss, the geometric pipeline downstream of $g(\mathbf{v})$ is well-defined for any differentiable encoder. We exploit this in Section 5 to isolate the contribution of the encoder from the contribution of the geometric aggregation.

3.5 Sentence aggregation and classification

Token-to-Sentence Aggregation Given a sentence with n tokens producing per-token SPD matrices $\{g(\mathbf{v}_1), \dots, g(\mathbf{v}_n)\} \in \mathcal{S}_{++}^{d_{\max}}$ (Section 3.4), we aggregate them into a single sentence-level SPD descriptor using the Fréchet mean (Fletcher et al., 2004) on the SPD manifold:

$$G_{\text{sentence}} = \arg \min_{G \in \mathcal{S}_{++}^{d_{\max}}} \sum_{i=1}^n d_{\text{Riem}}^2(G, g(\mathbf{v}_i)), \quad (7)$$

with d_{Riem} the affine-invariant Riemannian distance (Förstner et al., 2000):

$$d_{\text{Riem}}(G_1, G_2) = \left\| \log(G_1^{-1/2} G_2 G_1^{-1/2}) \right\|_F. \quad (8)$$

G_{sentence} is the geometric counterpart of Euclidean mean pooling, a Fréchet-mean point on the SPD manifold rather than an arithmetic mean.

Classification The sentence-level SPD matrices are classified in three steps. First, we apply Riemannian whitening, which rescales each matrix to unit determinant while preserving its affine-invariant geometric properties, removing scale differences across samples. Second, we project each whitened matrix to the tangent space at the population Riemannian mean of the training set.

The tangent space at a point on the SPD manifold is a flat Euclidean space, so standard linear classifiers apply. Third, we fit a logistic regression on the vectorised tangent-space representations. We use pyRiemann’s (Barachant et al., 2026) `TSCClassifier` with default hyperparameters, which implements the projection-plus-linear-classifier pipeline.

3.6 Experimental configuration

All experiments employed 5-fold stratified cross-validation to ensure balanced class distributions across folds, using two seeds. We report the mean and standard deviation across folds to capture performance variability.

Because the classification pipeline reuses established BCI components (e.g., Fréchet mean of SPD matrices, Riemannian whitening, and tangent-space classifier) we adopt the evaluation and statistical protocol of the Mother of all BCI Benchmarks (MOABB) (Aristimunya et al., 2023). Statistical significance was computed independently for each dataset using significance thresholds of $\alpha < 0.05$ and $\alpha < 0.01$. Effect sizes are reported using the standardised mean difference (SMD), and the aggregated meta-effect across datasets is also provided.

4 Experimental results

Table 1 reports two baselines, Linear Probe (Euclidean mean pooling) and CLS Token Aggregation, alongside RMP on the three signal-bearing datasets (CoLA, CREAK, RTE) and the FEVER-Symmetric negative control, with means and standard deviations across the multi-seed cross-

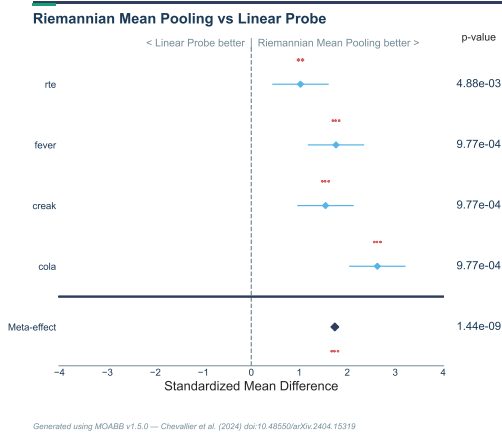


Figure 2: Statistical comparison between Linear Probe and Riemannian Mean Pooling across random seeds. Box plots show the distribution of AUC scores obtained from 10 evaluations (2 seeds $\times$ 5 folds). Corresponding p-values are displayed on the right side of each plot. One (respectively, two and three) red star(s) indicate that the p-value is below the significance threshold of $\alpha < 0.05$ (respectively, $\alpha < 0.01$, $\alpha < 0.001$). FEVER-Symmetric is a negative control (Section 3), so a chance-level result there is the expected behaviour.

validation protocol.

Figure 2 presents forest plot comparing the distribution of AUC scores across all 10 evaluations for each dataset.

Per-dataset AUC, Accuracy, and F1 bar plots together with the CLS-vs-RMP forest plot (Figures 10, 11, 12, 13) are deferred to Appendix C. Figure 2 and Table 1 carry the headline comparison in the body.

The multi-seed meta-analysis shows RMP outperforming both baselines (Linear Probe and CLS Token Aggregation) on CoLA, CREAK, and RTE, the three datasets with classification signal accessible from claim surface. CLS Token Aggregation underperforms Linear Probe on three of four datasets (FEVER, CoLA, CREAK), with RTE the exception (0.611 vs 0.591 AUC). This replicates the observation of Reimers and Gurevych (2019) that vanilla CLS embeddings underperform mean pooling on transfer tasks without task-specific fine-tuning, and provides a second controlled Euclidean baseline against which to measure the Riemannian gain. On FEVER-Symmetric, RMP stays at chance, which is the behaviour expected from a negative control. Its point estimate of 0.507 AUC nominally exceeds 0.5, but it falls well inside the 95 % interval [0.419, 0.579] of a label-permutation null over 10,000 draws (Müller-Putz

et al., 2008; Maris and Oostenveld, 2007), at $p = 0.83$ and a standardised mean difference of 0.16, below the conventional threshold for a small effect. The two Euclidean baselines do not stay at chance but fall significantly below it. This is further discussed in Section 6.

5 Decomposing the geometric gain

5.1 Ablation Components

To localise the source of the geometric gain, we ablate the pipeline along two axes: (i) whether the encoder is trained or randomly initialised, and (ii) whether the encoder is the IGL MLP architecture or a simpler random linear projection. **Encoder-Only** projects tokens through the trained IGL encoder and applies Euclidean mean pooling in latent space (no metric extraction), testing whether the encoder alone matters. **Frozen Random IGL** replaces the trained encoder with a randomly initialised MLP of identical architecture (256-d hidden, SiLU) and runs the full geometric pipeline, testing whether encoder training matters. **Random Projection + SiLU** uses a frozen random linear projection $\mathbb{R}^{768} \rightarrow \mathbb{R}^{64}$ followed by SiLU, then the full geometric pipeline, testing whether the depth and nonlinearity of the IGL MLP matter even without training.

5.2 Experimental Results

The full ablation table (Table 2) and per-comparison forest/bar plots are deferred to Appendix B, and the qualitative findings are summarised below. In addition, Appendix E reports a further control in which the tangent-space and mean-pooled BERT features are evaluated separately and concatenated under one classifier, and finds that the size of the feature set does not account for the geometric gain.

5.3 Findings

Three patterns emerge from Table 2. First, a randomly initialised encoder recovers most of the geometric gain on every signal-bearing dataset (CoLA 0.760 against 0.782, CREAK 0.667 against 0.685, RTE 0.615 against 0.632), indicating that the geometric aggregation, rather than encoder training, carries most of the gain. Second, the trained encoder adds 1.7 to 2.2 AUC points on all three datasets (paired over folds, CoLA gains +0.022 at $p = 0.0002$, CREAK +0.018 at $p = 0.004$, and RTE +0.017 at $p = 0.024$). Third,

Dataset	Method	Accuracy	F1-Score	AUC
FEVER[†]	Linear Probe	0.405 $\pm$ 0.046	0.426 $\pm$ 0.036	0.425 $\pm$ 0.029
	CLS Token Aggregation	0.375 $\pm$ 0.051	0.398 $\pm$ 0.053	0.402 $\pm$ 0.052
	Riemannian Mean Pooling	0.494 $\pm$ 0.043	0.513 $\pm$ 0.044	0.507 $\pm$ 0.039
CoLA	Linear Probe	0.675 $\pm$ 0.018	0.678 $\pm$ 0.017	0.731 $\pm$ 0.021
	CLS Token Aggregation	0.661 $\pm$ 0.014	0.662 $\pm$ 0.016	0.711 $\pm$ 0.012
	Riemannian Mean Pooling	0.711 $\pm$ 0.014	0.723 $\pm$ 0.014	0.782 $\pm$ 0.021
CREAK	Linear Probe	0.607 $\pm$ 0.026	0.604 $\pm$ 0.026	0.646 $\pm$ 0.032
	CLS Token Aggregation	0.593 $\pm$ 0.026	0.593 $\pm$ 0.027	0.631 $\pm$ 0.024
	Riemannian Mean Pooling	0.633 $\pm$ 0.014	0.628 $\pm$ 0.021	0.685 $\pm$ 0.018
RTE	Linear Probe	0.559 $\pm$ 0.014	0.554 $\pm$ 0.020	0.591 $\pm$ 0.025
	CLS Token Aggregation	0.573 $\pm$ 0.028	0.581 $\pm$ 0.028	0.611 $\pm$ 0.029
	Riemannian Mean Pooling	0.594 $\pm$ 0.021	0.597 $\pm$ 0.018	0.632 $\pm$ 0.025

Table 1: Multi-seed meta-analysis results (mean $\pm$ std across 10 evaluations). [†]FEVER is included as a negative control (Section 3). Chance-level performance is the expected and correct behaviour, so no method is bolded for this dataset.

the simpler Random Projection + SiLU underperforms the random IGL encoder on every signal-bearing dataset, indicating that the depth and non-linearity of the MLP matter even without training. The FEVER-Symmetric rows are reported for completeness but, by the negative-control framing of Section 3, are not informative about the source of geometric signal.

6 Discussion

6.1 Hypothesis validation

Geometric advantage and what it tests. Riemannian aggregation of token-level pullback metrics outperforms Euclidean mean pooling on the three signal-bearing datasets and stays at chance on the FEVER-Symmetric negative control (Table 1 and Figure 2). The gain is largest on CoLA (Δ AUC = 0.051) and similar on RTE and CREAK (0.041 and 0.040). The ablations localise most of this gain to the geometric aggregation rather than to learned manifold structure. RMP with a random encoder already significantly beats Euclidean pooling on CoLA and CREAK, while the trained encoder adds 1.7 to 2.2 AUC points across all three datasets (Table 2, with per-comparison forest plots in Appendix B). Rather than concluding that BERT embeddings lie on a globally coherent manifold, the evidence supports *locally manifold-like structure* (Robinson et al., 2025), under which aggregated geometric features yield classification gains even where global manifold structure fails.

Role of the negative control The inclusion of FEVER-Symmetric (Schuster et al., 2019) serves

a methodological purpose distinct from the other three datasets. Where CoLA, CREAK, and RTE probe whether geometric structure encodes specific linguistic properties, FEVER-Symmetric probes whether the method exploits residual signal in the underlying representations.

RMP stays at chance on it, reaching 0.507 against a label-permutation null of 0.500 (10,000 permutations, 95 % interval [0.419, 0.579], $p = 0.83$, SMD 0.16). The two Euclidean baselines perform significantly below chance (Linear Probe reaches 0.425 and CLS Token Aggregation 0.402). This phenomenon is explained by our evaluation protocol combined with dataset construction. By dataset construction, we mean that FEVER-Symmetric pairs most of the claims (228 over 316 rows) with a minimal edit carrying the opposite label. In fact, BERT embeddings place a claim and its twin close together in representation space. Therefore, a classifier that operates on claim-surface features will assign both members of a pair the same predicted label. This is an artifact of our evaluation method. Indeed, under five-fold cross-validation a pair is split across training and test four times out of five, so the classifier is trained on one claim and evaluated on its negation 80% of the time. The member of the pair in the testing set then receives the label of its twin in the training set. This is why both baselines invert on the twinned rows, while RMP staying close to chance is the expected behaviour. Appendix D reports the negative control in full.

What does the pullback metric encode?

The random-encoder ablations clarify what the pipeline is actually testing. The pullback met-

ric $g(\mathbf{v}) = (JJ^\top + \epsilon I)^{-1}$ captures the local anisotropy of the encoder map, that is, the extent to which a unit input perturbation at $\mathbf{v}$ is stretched or compressed across latent dimensions. For a random MLP with SiLU nonlinearity, $J(\mathbf{v})$ varies smoothly with $\mathbf{v}$ because of the nonlinearity, even though the encoder weights themselves are random. Consequently $g(\mathbf{v})$ encodes the position of $\mathbf{v}$ in input space through the local Jacobian structure of the random transformation, not through any property the encoder learned. Fréchet aggregation on the SPD manifold then summarises this local-geometry signal into a sentence-level descriptor. The geometric-aggregation gain over Euclidean pooling thus reflects that token positions in BERT embeddings carry classification signal in their local nonlinear-transformation structure, which Euclidean averaging discards.

6.2 Limitations and challenges

Computational cost. On top of one-time IGL training, each sentence requires a per-token Jacobian and an SPD Fréchet-mean iteration, an order of magnitude more arithmetic than Euclidean pooling, though still negligible against LLM inference or gradient-based fine-tuning. Exact wall-clock and FLOP figures are reserved for a scaling study.

Hyperparameter and architectural scope. Our encoder-width and anchor-count sweep was conducted only on FEVER-Symmetric, so sensitivity on the signal-bearing datasets and to the metric regulariser λ is unverified. The encoder objective itself was held fixed at IGL. Alternative objectives that also yield a Jacobian (vanilla autoencoders, VAEs, normalising flows, contrastive encoders) were not swept, although the random-encoder ablation already suggests the specific objective is largely transparent on most tasks.

Scope of the controlled comparison. The pipeline was evaluated against Euclidean and CLS-token baselines on a single encoder (BERT-base) at a single layer (layer 9) and a fixed projection dimensionality. This is a deliberate methodological choice. The comparison isolates the contribution of geometric aggregation on identical embeddings, which is why fine-tuned BERT and Sentence-BERT (Reimers and Gurevych, 2019) are excluded rather than missing. Absolute classification performance is not the question this pa-

per investigates. The question is whether geometric aggregation extracts signal that flat-Euclidean pooling discards.

6.3 Future work

Architecture and layer coverage. Extending the evaluation to larger and structurally different encoders (RoBERTa, DistilBERT, decoder-only LLMs) and across layers would clarify whether the geometric-aggregation gain is general or specific to BERT-base layer 9, and whether the layer-wise intrinsic-dimension trajectory (Valeriani et al., 2023; Cheng et al., 2025) interacts with where the pullback metric becomes informative.

Richer SPD descriptors. The Fréchet mean is the simplest aggregator on the SPD manifold. Weighted Fréchet means, trajectory-based descriptors in the spirit of layer-wise geometric probes (Damirchi et al., 2026; Mir, 2025), and explicit curvature features are natural next steps, with per-token pullback eigenvalues offering a route to relate the geometric signal to interpretable linguistic phenomena.

7 Conclusion

The SPD Fréchet mean of per-token pullback metrics beats Euclidean mean pooling on CoLA, CREAK and RTE under a controlled comparison on identical embeddings. On FEVER-Symmetric, included as a negative control, it correctly stays at chance. Random-encoder ablations localise most of the gain to the geometric aggregation rather than to learned manifold structure, with the trained encoder contributing a further 1.7 to 2.2 AUC points on every signal-bearing dataset. Together these results support the framing of embeddings as locally manifold-like rather than globally coherent (Robinson et al., 2025), under which aggregated geometric features yield classification gains even where global manifold structure may fail.

References

- Guillaume Alain and Yoshua Bengio. 2017. [Understanding intermediate layers using linear classifier probes](#). In *International Conference on Learning Representations (ICLR) Workshop Track*.
- Anton Andreev, Gregoire Cattan, and Marco Congedo. 2025. [The Riemannian Means Field Classifier for EEG-Based BCI Data](#). *Sensors*, 25(7):2305.

- Bruno Aristimunha, Igor Carrara, Pierre Guetschel, Sara Sedlar, Pedro Rodrigues, Jan Sosulski, Divyesh Narayanan, Erik Bjareholt, Barthélemy Quentin, Robin Tibor Schirrmester, Emmanuel Kalunga, Ludovic Darnet, Cattán Gregoire, Ali Abdul Hussain, Ramiro Gatti, Vladislav Goncharenko, Jordy Thielen, Thomas Moreau, Yannick Roy, and 3 others. 2023. [Mother of all BCI benchmarks](#). In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.
- Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. 2018. [Latent space oddity: On the curvature of deep generative models](#). In *International Conference on Learning Representations (ICLR)*.
- Alexandre Barachant, Quentin Barthélemy, Jean-Rémi King, Alexandre Gramfort, Sylvain Chevalier, Pedro L. C. Rodrigues, Emanuele Olivetti, Vladislav Goncharenko, Gabriel Wagner vom Berg, Ghiles Reguig, Arthur Lebeurier, Erik Bjareholt, Maria Sayu Yamamoto, Pierre Clisson, Marie-Constance Corsi, Igor Carrara, Apolline Mellot, Bruna Junqueira Lopes, Brent Gaisford, and 4 others. 2026. [pyriemann](#).
- Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. 2012. [Multiclass brain-computer interface classification by Riemannian geometry](#). *IEEE transactions on bio-medical engineering*, 59(4):920–928.
- Matthew Brand. 2003. Charting a manifold. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Julian Brandon, Angus Chadwick, and Arthur Pellegrino. 2025. [Emergent Riemannian geometry over learning discrete computations on continuous manifolds](#). arXiv preprint arXiv:2512.00196.
- Daniel Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. 2019. [Riemannian batch normalization for SPD neural networks](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Ziheng Chen, Yue Song, Rui Wang, Xiaojun Wu, and Nicu Sebe. 2024. [RMLR: Extending multinomial logistic regression into general geometries](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Jade Yu, Alessandro Laio, and Marco Baroni. 2025. [Emergence of a high-dimensional abstraction phase in language transformers](#). In *International Conference on Learning Representations (ICLR)*.
- Sylvain Chevallier, Igor Carrara, Bruno Aristimunha, Pierre Guetschel, Sara Sedlar, Bruna Lopes, Sebastien Velut, Salim Khazem, and Thomas Moreau. 2024. [The largest EEG-based BCI reproducibility study for open science: the MOABB benchmark](#). arXiv preprint. ArXiv:2404.15319.
- Sungjun Cho, Seunghyuk Cho, Sungwoo Park, Han-kook Lee, Honglak Lee, and Moontae Lee. 2023. [Curve your attention: Mixed-curvature transformers for graph representation learning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. 2017. [Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review](#). *Brain-Computer Interfaces*, 4(3):155–174.
- Hamed Damirchi, Ignacio Meza De la Jara, Ehsan Abbasnejad, Afshar Shamsi, Zhen Zhang, and Javen Shi. 2026. [Truth as a trajectory: What internal representations reveal about large language model reasoning](#). Preprint, arXiv:2603.01326.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.
- Riccardo Di Sipio, Jairo Diaz-Rodriguez, and Luis Serano. 2025. [The curved spacetime of transformer architectures](#). arXiv preprint arXiv:2511.03060.
- Willem Diepeveen, Georgios Batzolis, Zakhar Shumaylov, and Carola-Bibiane Schönlieb. 2024. [Score-based pullback Riemannian geometry: Extracting the data manifold geometry using anisotropic flows](#). arXiv preprint arXiv:2410.01950.
- Kawin Ethayarajh. 2019. [How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- P Fletcher, Conglin Lu, Stephen Pizer, and Sarang Joshi. 2004. [Principal geodesic analysis for the study of nonlinear statistics of shape](#). *Medical Imaging, IEEE Transactions on*, 23:995 – 1005.
- Wolfgang Förstner, Boudewijn Moonen, and Carl Gauss. 2000. [A metric for covariance matrices](#).
- Xiaofang Gao and Jiye Liang. 2011. [The dynamical neighborhood selection based on the sampling density and manifold curvature for isometric data embedding](#). *Pattern Recognition Letters*, 32(2):202–209.
- Wes Gurnee and Max Tegmark. 2024. [Language models represent space and time](#). In *International Conference on Learning Representations (ICLR)*.
- Michael Hauser and Asok Ray. 2017. [Principles of Riemannian geometry in neural networks](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.

- Saahith Janapati and Yangfeng Ji. 2025. [A comparative study of learning paradigms in large language models via intrinsic dimension](#). In *Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025)*, pages 59–86, Albuquerque, NM. Association for Computational Linguistics.
- Zhongping Ji. 2025. [RiemannFormer: A framework for attention in curved spaces](#). arXiv preprint arXiv:2506.07405.
- Ce Ju, Reinmar Kobler, Antoine Collas, Motoaki Kawanabe, Cuntai Guan, and Bertrand Thirion. 2024. [SPD matrix learning for neuroimaging analysis: Perspectives, methods, and challenges](#). arXiv preprint arXiv:2504.18882.
- Takuya Kataiwa, Cho Hakaze, and Tetsushi Ohki. 2025. [Measuring intrinsic dimension of token embeddings](#). arXiv preprint arXiv:2503.02142.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. [Matryoshka representation learning](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yangyang Li. 2017. [Curvature-aware Manifold Learning](#). arXiv preprint. ArXiv:1706.07167 [cs.LG].
- Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. 2020. [Differentiating through the Fréchet mean](#). In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.
- Rob Manson. 2025. [Curved inference: Concern-sensitive geometry in large language model residual streams](#). Preprint, arXiv:2507.21107.
- Eric Maris and Robert Oostenveld. 2007. [Nonparametric statistical testing of EEG- and MEG-data](#). *Journal of Neuroscience Methods*, 164(1):177–190.
- Samuel Marks and Max Tegmark. 2024. [The geometry of truth: Emergent linear structure in large language model representations of true/false datasets](#). In *Conference on Language Modeling (COLM)*.
- Leland McInnes, John Healy, and James Melville. 2018. [UMAP: Uniform manifold approximation and projection for dimension reduction](#). arXiv preprint arXiv:1802.03426.
- Amir Hameed Mir. 2025. [The geometry of truth: Layer-wise semantic dynamics for hallucination detection](#). arXiv preprint arXiv:2510.04933.
- Gernot Müller-Putz, Reinhold Scherer, Clemens Brunner, Robert Leeb, and Gert Pfurtscheller. 2008. Better than Random? A closer look on BCI results. *International Journal of Bioelektromagnetism*, 10:52–55.
- Maximilian Nickel and Douwe Kiela. 2017. [Poincaré embeddings for learning hierarchical representations](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Maximilian Nickel and Douwe Kiela. 2018. [Learning continuous hierarchies in the Lorentz model of hyperbolic geometry](#). In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.
- Yasumasa Onoe, Michael J.Q. Zhang, Eunsol Choi, and Greg Durrett. 2021. [Creak: A dataset for common-sense reasoning over entity knowledge](#). *OpenReview*.
- Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. 2025. [The geometry of categorical and hierarchical concepts in large language models](#). In *International Conference on Learning Representations (ICLR)*.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. [The linear representation hypothesis and the geometry of large language models](#). In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. [Scikit-learn: Machine learning in Python](#). *Journal of Machine Learning Research*, 12:2825–2830.
- Alexandre Quemy. 2026. [Intrinsic Green’s Learning: Supervised Learning on Manifolds via Inverse PDE](#). In *AI & PDE Workshop at ICLR*.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Michael Robinson, Sourya Dey, and Tony Chiang. 2025. [Token embeddings violate the manifold hypothesis](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. [A primer in BERTology: What we know about how BERT works](#). *Transactions of the Association for Computational Linguistics (TACL)*, 8:842–866.
- Frederic Sala, Christopher De Sa, Albert Gu, and Christopher Ré. 2018. [Representation tradeoffs for hyperbolic embeddings](#). In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.
- Tal Schuster, Darsh J Shah, Yun Jie Serene Yeo, Daniel Filizola, Enrico Santus, and Regina Barzilay. 2019.

- Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- A. Singer. 2006. [From graph to manifold laplacian: The convergence rate](#). *Applied and Computational Harmonic Analysis*, 21(1):128–134. Special Issue: Diffusion Maps and Wavelets.
- Bartłomiej Sobieski, Matthew Tivnan, Dawid Pludowski, Michał Jan Włodarczyk, Pengfei Jin, Przemysław Biecek, and Quanzheng Li. 2026. [Local intrinsic dimension unveils hallucinations in diffusion models](#). *Preprint*, arXiv:2605.05026.
- Matthieu Tehenan, Vikram Natarajan, Jonathan Michala, Milton Lin, and Juri Opitz. 2025. [Mechanistic decomposition of sentence representations](#). arXiv preprint arXiv:2506.04373.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. [FEVER: a large-scale dataset for fact extraction and VERification](#). In *NAACL-HLT*.
- Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. 2023. [The geometry of hidden representations of large transformer models](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. [Neural network acceptability judgments](#). *arXiv preprint arXiv:1805.12471*.

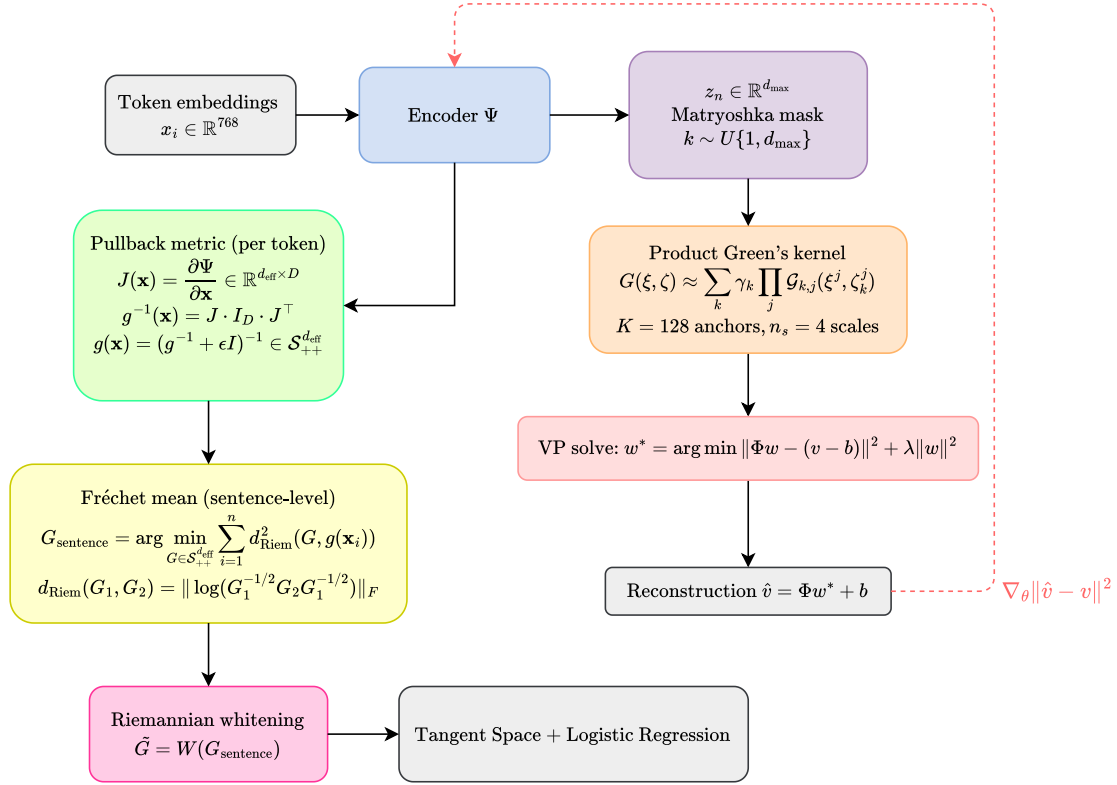


Figure 3: Detailed architecture of token-level IGL for sentence classification via RMP. **Right path (reconstruction):** Token embeddings $\mathbf{v}_i \in \mathbb{R}^{768}$ are encoded to latent space $\mathbf{z} \in \mathbb{R}^{d_{\max}}$ via encoder Ψ_θ , then mapped through Product Green’s kernel $G(\xi, \zeta)$ to design matrix Φ . Variable projection (VP) solves for optimal weights $w^* = \arg \min_w \|\Phi w - (y - b)\|^2 + \lambda \|w\|^2$ to reconstruct input $\hat{\mathbf{v}} = \Phi w^* + b$. **Left path (classification):** The trained encoder’s Jacobian $J = \frac{\partial \Psi_\theta}{\partial \mathbf{v}}$ induces pullback metrics $g(\mathbf{v}_i) \in \mathcal{S}_{++}^{d_{\max}}$ at each token. These are aggregated via Fréchet mean to sentence-level SPD matrix G_{sentence} , preprocessed with Riemannian whitening, projected to tangent space, vectorised, and classified with logistic regression. **Solid arrows:** forward flow. **Dashed red arrow:** reconstruction-loss gradient $\nabla_\theta \|\hat{\mathbf{v}} - \mathbf{v}\|^2$ flowing back to encoder Ψ_θ . The encoder is trained jointly with VP solve until reconstruction converges (early stopping with patience=50, $\delta = 2 \times 10^{-5}$), then frozen for metric extraction and classification.

A IGL architecture details

This appendix collects the architectural details of the IGL configuration used in Section 3.3.

Encoder and Matryoshka Truncation A small MLP encoder $\Psi_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^{d_{\max}}$ maps input to latent codes $\xi_n = \Psi_\theta(v_n)$ with $d_{\max} = 64$. During training, a random truncation $k \sim \text{Uniform}\{1, \dots, d_{\max}\}$ is sampled and trailing coordinates zeroed out, following Matryoshka representation learning (Kusupati et al., 2022). This ensures earlier dimensions capture the most informative features, with effective dimension $d_{\max}$ determined post-training from the dimension-loss curve.

Product Green’s Kernel The reconstruction target is the input v_n itself. We model the reconstruction by a tensor-factorised product Green’s kernel:

$$G(\xi, \zeta) \approx \sum_{k=1}^K \gamma_k \prod_{j=1}^{d_{\max}} G_{k,j}(\xi_j, \zeta_j),$$

$$K = n_{\text{anchors}} = 128 \quad (9)$$

Each factor $G_{k,j}$ is a one-dimensional kernel evaluated at the latent coordinate ξ_j^j and an anchor ζ_k^j , with K anchors placed adaptively. We use Gaussian kernel factors. Each factor is itself a convex mixture of $n_{\text{scales}} = 4$ Gaussian basis functions at learnable bandwidths $\sigma_{s,j}$,

Dataset	Method	Accuracy	AUC
FEVER [†]	Linear Probe	0.405 ± 0.046	0.425 ± 0.029
	CLS Token Aggregation	0.375 ± 0.051	0.402 ± 0.052
	Riemannian Mean Pooling	0.494 ± 0.043	0.507 ± 0.039
	Encoder-Only	0.506 ± 0.043	0.537 ± 0.036
	Frozen Random IGL	0.426 ± 0.040	0.441 ± 0.035
	Random Projection + SiLU	0.498 ± 0.080	0.515 ± 0.081
CoLA	Linear Probe	0.675 ± 0.018	0.731 ± 0.021
	CLS Token Aggregation	0.661 ± 0.014	0.711 ± 0.012
	Riemannian Mean Pooling	0.711 ± 0.014	0.782 ± 0.021
	Encoder-Only	0.686 ± 0.031	0.754 ± 0.032
	Frozen Random IGL	0.695 ± 0.022	0.760 ± 0.021
	Random Projection + SiLU	0.641 ± 0.024	0.692 ± 0.028
CREAK	Linear Probe	0.607 ± 0.026	0.646 ± 0.032
	CLS Token Aggregation	0.593 ± 0.026	0.631 ± 0.024
	Riemannian Mean Pooling	0.633 ± 0.014	0.685 ± 0.018
	Encoder-Only	0.623 ± 0.011	0.664 ± 0.017
	Frozen Random IGL	0.623 ± 0.017	0.667 ± 0.017
	Random Projection + SiLU	0.584 ± 0.023	0.621 ± 0.028
RTE	Linear Probe	0.559 ± 0.014	0.591 ± 0.025
	CLS Token Aggregation	0.573 ± 0.028	0.611 ± 0.029
	Riemannian Mean Pooling	0.594 ± 0.021	0.632 ± 0.025
	Encoder-Only	0.584 ± 0.026	0.630 ± 0.031
	Frozen Random IGL	0.576 ± 0.022	0.615 ± 0.025
	Random Projection + SiLU	0.561 ± 0.024	0.583 ± 0.030

Table 2: Ablation study results (mean ± std across 10 evaluations). Bold indicates best per dataset. F1 is omitted from this table (tracks Accuracy closely on binary tasks). The full Accuracy/F1/AUC table is in Table 1, which is drawn from the same run, so the rows the two tables share are identical by construction. [†]FEVER is included as a negative control (Section 3). Rows are reported for completeness but are not informative about the source of geometric signal, so no method is bolded.

$$G_{k,j}(\xi_j, \zeta_j^k) = \sum_{s=1}^4 \alpha_s \exp \left(-\frac{(\xi_j - \zeta_j^k)^2}{2\sigma_{s,j}^2} \right),$$

$$a = \text{softmax}(\cdot) \quad (10)$$

Variable-Projection Reconstruction Each minibatch solves bias-centered ridge regression for weights, with $\lambda = 10^{-3}$:

$$w^* = \arg \min_w \|\Phi w - (v - b)\|^2 + \lambda \|w\|^2 \quad (11)$$

The design matrix Φ costs $O(NKd)$ to compute, while the closed-form solve is $O(NK^2 + K^3)$ and independent of d . Weights w^* are recomputed per minibatch (not stored), so gradient descent operates only on encoder parameters θ .

B Ablation figures

The full ablation-results table and per-comparison forest plots and bar plots for the three ablation contrasts (Section 5) are collected here for space.

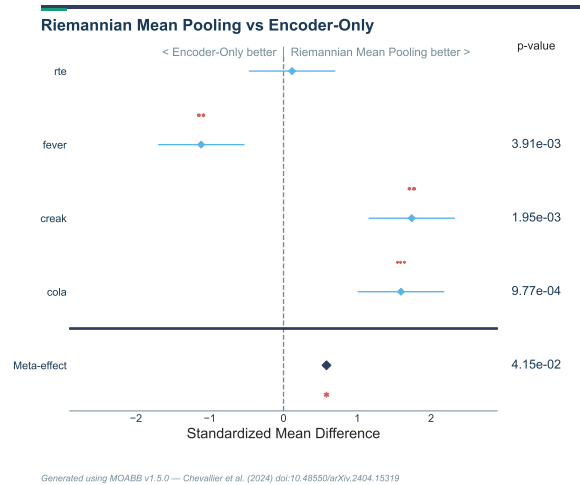


Figure 4: Riemannian Mean Pooling vs Encoder-Only: forest plot. Metric: AUC.

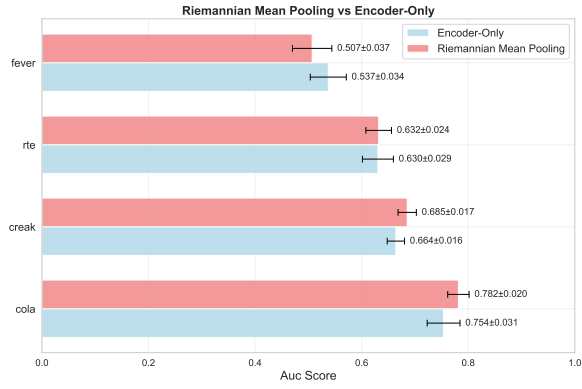


Figure 5: AUC bar comparison: Riemannian Mean Pooling vs Encoder-Only.

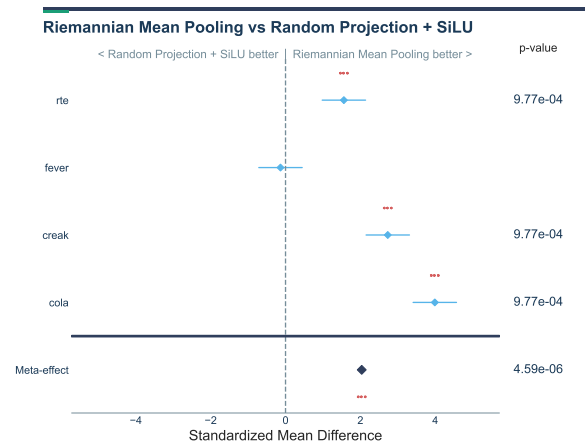


Figure 8: Riemannian Mean Pooling vs Random Projection + SiLU: forest plot. Metric: AUC.

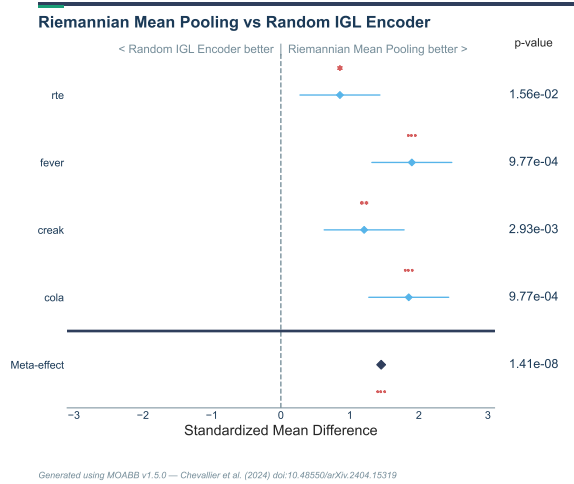


Figure 6: Riemannian Mean Pooling vs Frozen Random IGL: forest plot. Metric: AUC.

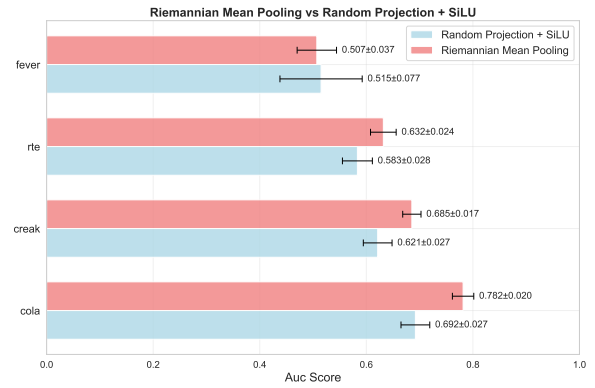


Figure 9: AUC bar comparison: Riemannian Mean Pooling vs Random Projection + SiLU.

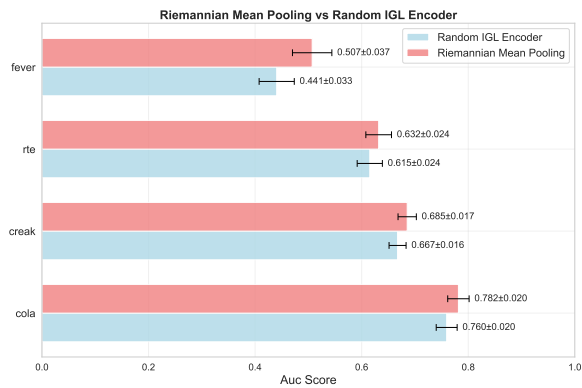


Figure 7: AUC bar comparison: Riemannian Mean Pooling vs Frozen Random IGL.

C Additional results figures

The per-dataset AUC, Accuracy, and F1 bar plots together with the CLS-vs-RMP forest plot are deferred here. The body keeps the headline Linear-Probe-vs-RMP forest plot (Figure 2) and Table 1 as the main visualisations.

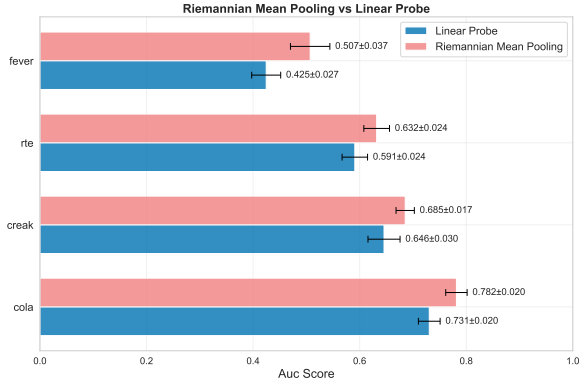


Figure 10: Multi-seed AUC comparison across datasets.

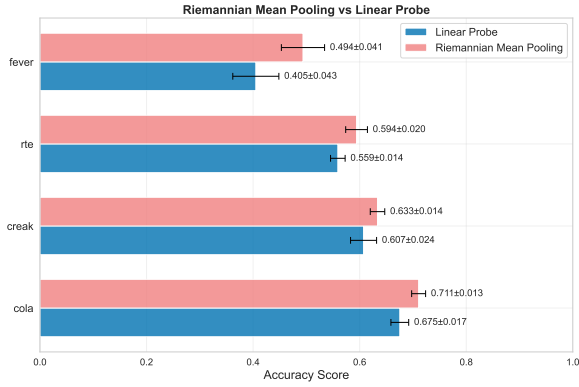


Figure 11: Multi-seed accuracy comparison across datasets. Error bars represent standard deviation across 10 evaluations.

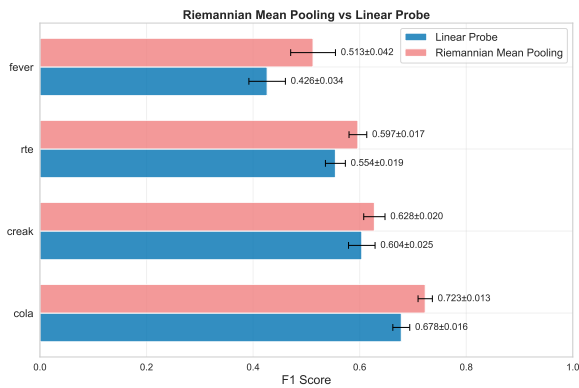


Figure 12: Multi-seed F1-score comparison across datasets.

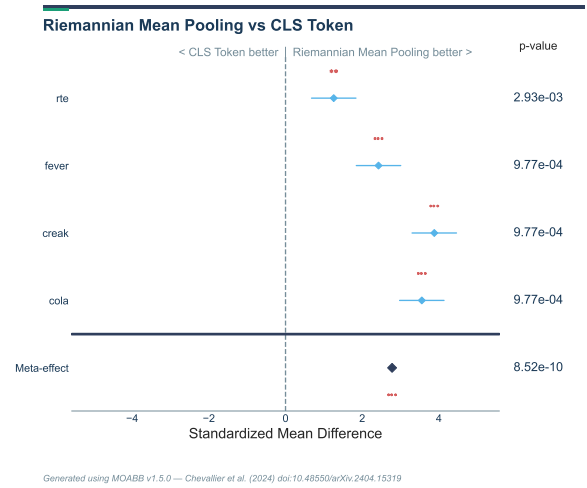


Figure 13: Statistical comparison between CLS Token Aggregation and RMP across random seeds. Box plots show the distribution of AUC scores from 10 evaluations (2 seeds $\times$ 5 folds). Effect sizes are reported using SMD. FEVER-Symmetric is included as a negative control (Section 3). A chance-level result is the expected and correct behaviour for both methods.

D The FEVER-Symmetric twin split

FEVER-Symmetric was built to remove the annotation artifacts of the original FEVER. Each claim is paired with a minimal edit of itself that carries the opposite label. The edit is sometimes a negation and more often a replaced noun or date, as in *founded in 1971* against *founded in 1991*, or *the football championship of Africa* against *the football championship of Asia*. Explicit negation words appear in only 22 of the 316 claims, so most pairs differ by a substitution rather than by a negation. A detector that relies on explicit negation alone has almost nothing to fire on and reaches 0.543 AUC, close to chance, so the pairing leaves no lexical shortcut to the labels.

Schuster et al. (2019) do not release the pair index, so we recover the pairing from claim text and treat it as an approximation. We obtain the pairings by matching each row to its opposite-label nearest neighbour at Jaccard overlap ≥ 0.6 , yielding 114 pairs (228 rows). The remaining 88 rows have no partner and no debiasing guarantee. The threshold is an arbitrary choice: 0.4, 0.5, and 0.7 yield 292, 282, and 168 paired rows respectively.

Method	All (316)	Twinned (228)	Untwinned (88)
Linear Probe	0.425	0.357	0.601
CLS Token	0.402	0.332	0.594
Riemannian Mean Pooling	0.509	0.445	0.662

Table 3: FEVER-Symmetric split by whether a row has an opposite-label twin (AUC, mean across 10 evaluations). All three methods share the fold structure. Five-fold cross-validation separates a pair four times in five, so a method that relies on claim-surface features is trained on one member and evaluated on the other. The RMP rows come from the permutation run reported in this appendix rather than from the run behind Table 1, so the full-set value reads 0.509 here against 0.507 there, a gap well inside the fold-to-fold standard deviation of 0.039. The baseline rows are the same in both.

Inversion on the twinned rows Both Euclidean baselines fall far below chance on the rows that carry a twin, to 0.357 and 0.332, and rise above it on the rows that do not, to 0.601 and 0.594. This difference of 0.25 AUC between two halves of the same dataset under the same model is consistent with the pairing construction, not with a property of the claims themselves.

Riemannian Mean Pooling on the twinned rows RMP also sits below chance on the twinned rows, at 0.445, so it is not immune, yet it is significantly less affected. Paired over the 10 evaluations it exceeds Linear Probe by 0.088 ($t = 4.63$, $p = 0.001$) and CLS Token Aggregation by 0.113 ($t = 7.87$, $p = 2.5 \times 10^{-5}$) on those rows. On the untwinned rows the three methods lie within 0.07 of each other and the differences are not significant. The full-set reading of 0.509 falls between the two halves, weighted towards the 228 twinned rows against the 88 untwinned.

Residual signal in the Encoder-Only ablation

The Encoder-Only ablation is the one method above chance on the full set, at 0.537 ± 0.036 ($p = 0.009$). It pools the trained encoder’s coordinates in latent space rather than reading its Jacobian. The FEVER-Symmetric construction (Schuster et al., 2019) targets surface features arising from the annotation process and does not target all model-accessible content, and BERT encodes substantial world knowledge through pre-training, so a Euclidean classifier over learned coordinates can detect a small residual signal. That signal is not specific to the geometric pipeline. In fact, on the untwinned rows the Linear Probe

reaches 0.601, so the content the debiasing leaves behind is accessible to mean pooling as well.

E Feature concatenation

A natural question is whether the geometric gain is a consequence of feature count. The tangent-space descriptor has 2080 coordinates against the 768 of a mean-pooled BERT vector, so a larger budget alone might account for it. Table 4 evaluates the two feature sets separately and concatenated, under one classifier. As in Table 2, the FEVER-Symmetric rows are reported for completeness and carry no information here, since no method on that dataset is expected to exceed chance. The comparisons below are over the three signal-bearing datasets.

Dataset	Features (d)	AUC
FEVER	BERT only (768)	0.425 ± 0.029
	Tangent space (2080)	0.401 ± 0.059
	Both (2848)	0.413 ± 0.035
CoLA	BERT only (768)	0.731 ± 0.021
	Tangent space (2080)	0.740 ± 0.013
	Both (2848)	0.757 ± 0.020
CREAK	BERT only (768)	0.646 ± 0.032
	Tangent space (2080)	0.615 ± 0.031
	Both (2848)	0.626 ± 0.035
RTE	BERT only (768)	0.591 ± 0.025
	Tangent space (2080)	0.592 ± 0.018
	Both (2848)	0.599 ± 0.026

Table 4: Feature concatenation on the whitened SPD descriptors (AUC, mean $\pm$ std across 10 evaluations). All three feature sets use the same standardised logistic regression, so the rows are comparable to each other. They are *not* comparable to the Riemannian Mean Pooling rows of Table 2, which use pyRiemann’s `TSClassifier` (tangent space followed by logistic regression, with no per-coordinate scaler). The BERT-only row is the Linear Probe of Table 1, recomputed here as one half of the comparison, and reproduces it exactly.

The largest feature set leads on CoLA and RTE while the smallest leads on CREAK, so the ordering is not monotone in the size of the feature set. With the classifier held fixed, more coordinates buy nothing consistent. Nearly three times as many gain +0.009 on CoLA ($p = 0.16$) and +0.001 on RTE ($p = 0.93$) while losing 0.031 on CREAK ($p = 0.04$). In other words, AUC does not follow feature count.

Against the BERT-only features, concatenation gains +0.027 AUC on CoLA ($p = 0.0002$), loses 0.019 on CREAK ($p = 0.03$), and does not move

RTE (+0.008, $p = 0.13$). It is inconsistent in the same way, therefore we do not treat concatenation as a component of the method.

F Reconstruction convergence

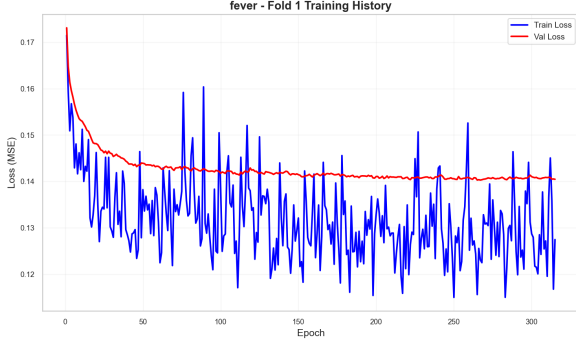


Figure 14: Training (blue) and validation (red) reconstruction MSE for the IGL encoder on fold 1 of FEVER-Symmetric, over ~ 310 epochs. The validation curve decreases monotonically from ~ 0.17 to ~ 0.14 . This is representative of all folds and datasets.

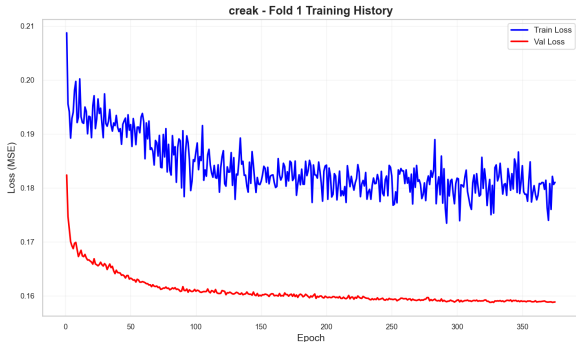


Figure 15: Training (blue) and validation (red) reconstruction MSE for the IGL encoder on fold 1 of CREAK, over ~ 375 epochs. The validation curve decreases from ~ 0.182 to ~ 0.159 .

G Immersion

Equation (5) treats Ψ_θ as a submersion. An alternative is to restrict it to the data manifold $M \subset \mathbb{R}^{768}$ of dimension $m < d_{\max}$, which makes it an immersion and to replace (5) with

$$g^{-1}(\mathbf{v}) = J(\mathbf{v}) P_M J(\mathbf{v})^\top, \quad (12)$$

where P_M projects onto $T_{\mathbf{v}}M$. We estimate P_M from the reconstruction readout as $J_R J_R^\top$. This appendix reports what (12) does to the descriptor. All runs are paired on fold, seed and encoder draw. They share one trained encoder per fold and differ only in the projector P_M .

Dimension estimates No estimator we ran returned a stable dimension for layer-9 token embeddings, so m is a choice rather than a measurement and the immersion variant is not identified by the data. TwoNN falls with sample size (14.2, 13.2, 10.3 at $n = 1000, 2000, 5000$). Maximum-likelihood estimates run 10.6 to 16.1 as k goes from 5 to 50, while local PCA runs 17, 29, 48, 69 as k goes from 25 to 200 at $n = 5000$ and does not agree across sample sizes. As a control we run the same estimators on a 2-torus embedded in $\mathbb{R}^{768}$, a surface of known dimension 2, where local PCA is flat at 2.0 for every k . The Matryoshka reconstruction curve, the estimator inherent to IGL, has no elbow. Figure 16 plots TwoNN, local PCA and the Matryoshka curve.

Choice of m and variants measured We set $m = 16$, which is a choice informed by the estimators rather than one derived from them. It is the top of the range where the only two estimators that agree with each other land, TwoNN at 10.3 to 14.2 across sample sizes and the maximum-likelihood estimate at 10.6 to 16.1 across k . The Matryoshka elbow detector returns 3 on a smooth monotone curve, which the gate rejects, and local PCA spans 17 to 69. We characterised the geometry at $m \in \{8, 16, 32, 48, 64\}$ and measured AUC at $m = 16$ alone, on CoLA (layer 9, 2 seeds $\times$ 5 folds, trained IGL encoder). Table 5 reports AUC, the numerical rank of g^{-1} before regularisation, and the fraction of J 's Frobenius mass inside each run subspace (overlap). A sixth variant, RANK_ROT, draws an independent rank- m projector at each token. We omit it from the table because it collapses the Fréchet mean for reasons unrelated to the direction of the projection, which is the comparison RANK_FIXED isolates by holding one projector fixed across tokens.

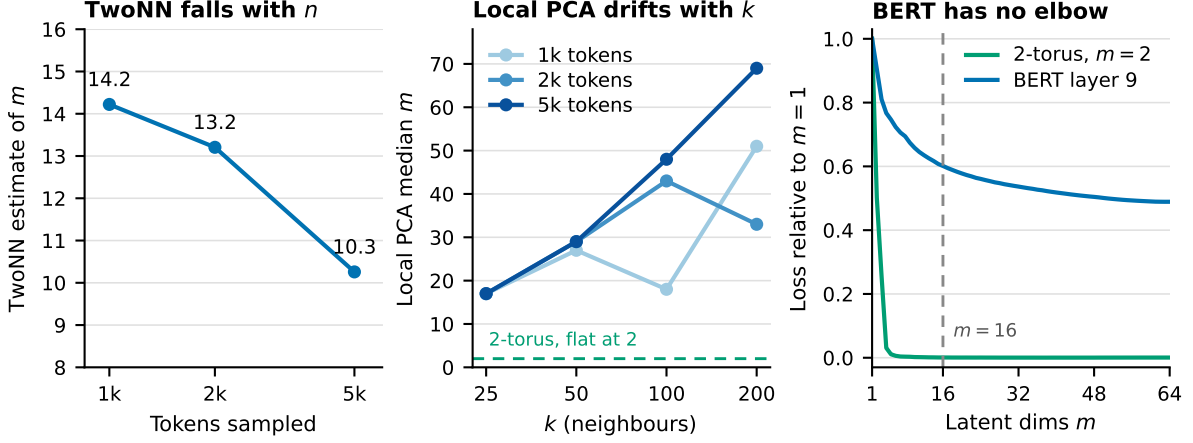


Figure 16: Stage 1 dimension estimates on CoLA layer-9 token embeddings (seed 67). **Left:** TwoNN falls as the sample grows, so it has not converged by $n = 5000$. **Centre:** the local PCA median rises with the neighbourhood size k and does not agree across sample sizes, while the same estimator returns 2.0 at every k on the torus control (dashed). **Right:** Matryoshka reconstruction loss against latent width, each curve divided by its own $m = 1$ value because the two absolute losses differ by two orders of magnitude. The torus curve reaches its floor by $m = 4$, while the BERT curve decays smoothly with no elbow to locate. We set $m = 16$ (dashed) for the variants in Table 5.

Arm	AUC	Rank	Overlap
BASELINE	0.783	64	1.000
IMMERSION	0.774	40	0.408
IMMERSION_TRUNC	0.776	16	0.089
RANK_FIXED	0.784	16	0.023
IMMERSION_LOCALPCA	0.710	16	0.040

Table 5: Immersion variants on CoLA ($m = 16$, $n = 10$ folds). The projector P_M is the identity for BASELINE, $J_R J_R^+$ for IMMERSION, the top- m singular directions of J_R for IMMERSION_TRUNC, a random shared rank- m subspace for RANK_FIXED, and an encoder-free local PCA estimate for IMMERSION_LOCALPCA. Rank is the numerical rank of (12) before regularisation. Overlap is the fraction of J ’s Frobenius mass inside the variant’s subspace.

Comparison of the rank-16 variants The decisive comparison is IMMERSION_TRUNC against RANK_FIXED. Both restrict to rank 16, but one uses the decoder’s top- m singular directions and the other an arbitrary fixed subspace. RANK_FIXED ties the baseline ($|\Delta \text{AUC}| < 0.001$, $p = 0.95$). IMMERSION_TRUNC sits below it ($\Delta \text{AUC} = -0.007$, $t = -2.38$, $p = 0.041$, uncorrected over the variant contrasts), so an arbitrary subspace of the same rank does as well as the decoder’s. The encoder-free tangent estimate (IMMERSION_LOCALPCA) is 0.074 below baseline, and its overlap with J_R ’s column space is 0.040 against 0.023 for the random rank-16 control, so the two estimates of $T_{\vee} M$ are near-

orthogonal.

Rank of the immersion variant The rank column of Table 5 should be read with the same caution as m . J_R has no spectral gap. Its singular values decay smoothly from 26.4 to 2.7 (a condition number of 9.6), and the largest ratio between consecutive values anywhere in the spectrum is 1.2. Counting the directions that carry 90% of the energy gives 38, counting the values above numerical zero gives 64, and a second run of the same variant on the same dataset reports 64 where the run in Table 5 reports 40. On the torus control the same estimator leaves no such ambiguity, with 3 directions carrying 90% of the energy and a factor of 1.1×10^4 between the fifth and sixth singular values. The decoder’s estimate of the tangent space is therefore no better identified than m is, which is consistent with the projector making no difference to AUC.

Numerical conditioning In `float32` the metrics lose positive-definiteness on rank-deficient matrices of this kind. Eigenvalues reach -2.2×10^{-7} against machine epsilon 1.2×10^{-7} . In such a scenario eigenvalues are numerically zero within arithmetic precision. While the regularised matrix is practically invertible, the regulariser dominates, and all variants tie for numerical reasons. We therefore perform the inversion in `float64`, where the regularisation fire rate is 0.0% across ev-

ery run and every m . The lack of improvement reflects geometry rather than numerical breakdown.

Restricting the metric to an estimated manifold therefore does not help, and since neither m nor $T_{\vee}M$ is identified on these embeddings, we report the submersion metric of (5) throughout.