MINKOWSKI ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper introduces *Minkowski Attention*, a novel formulation for attention and self-attention layers through a generalization of Minkowski spacetime. Attention layers must reason over two distinct representations per token, for content and position. The canonical fix, where content and position embeddings are summed before computing attention, entangles these two signals into cross terms that don't cleanly separate, hence a growing body of work advocates for decoupled attention. Decoupled attention however requires computing separate attention matrices for each pair of representations, inducing considerable compute and memory cost per layer. In this work, we show that with a Minkowski spacetime perspective, it becomes possible to recover a single attention score from decoupled representations. We propose to view token and positional embeddings as multi-dimensional space and time vectors, where token pairs are scored with a Minkowski inner product instead of a Euclidean dot product. The signed structure of Minkowski spacetime keeps content and positions contributions separable under one scalar. We generalize the (3+1)D Minkowski metric to arbitrary (p+q)-dimensional signatures and to the corresponding generalized Lorentz boosts. Our *Minkowski transformer* consistently outperforms coupled and decoupled transformers across various tasks and modalities, demonstrating strong attention diversity over attention heads and robustness to non-discriminative tokens.

1 INTRODUCTION

From text and images to graphs and chemical structures, attention has been the default computational unit of deep learning (Vaswani et al., 2017; Bommasani et al., 2021). Self-attention in particular has mostly displaced recurrence and convolution as the mechanism to model relationships between tokens, as demonstrated in frontier large language models (Touvron et al., 2023; Achiam et al., 2023), vision transformers (Dosovitskiy et al., 2021; Touvron et al., 2021), and graph attention networks (Yang et al., 2024). While simple in nature, the attention layer has a hidden persistent tension in the representation of each token. The attention contribution of a token not only depends on its content, but also on where it is positioned with respect to other tokens. A lot of advances have been proposed to improve the content (Roy et al., 2021) and position (Shaw et al., 2018; Kitaev et al., 2020; Press et al., 2021) encoding of tokens separately. But ultimately, both representations are canonically summed to be able to compute an attention score.

While efficient, coupling content and position information fuses two vastly different signals into one vector, hence the dot product mixes content-content, content-position, position-content, and position-position terms which cannot be disentangled afterwards. Due to this fundamental limitation, a wide range of works have advocated for decoupled attention, which instead keeps content and positional representations separate (He et al., 2020). Standard decoupling however requires computing attentions for each pair of representation, incurring major computational costs (Dai et al., 2019; Shaw et al., 2018).

In this work, we show that the trade-off between entangled-but-cheap and decoupled-but-costly attention is not fundamental, but a limitation incurred by treating attention as a Euclidean similarity computation. Our idea is to view content and position embeddings not as vectors in the same Euclidean space, but as multi-dimensional space and time coordinates of a generalized Minkowski spacetime. Scoring token pairs with a Minkowski inner product yields a single scalar attention score while keeping content and position contributions separable by construction. We generalize the (3+1)D Minkowski metric and its Lorentz transformations to arbitrary pseudo-Euclidean signatures

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and derive the pseudo-Lorentz transformations, allowing us to work with any-dimensional content and position vectors. We show that a single *Minkowski attention* matrix per layer outperforms both coupled and decoupled baselines. Summarized, our contributions are:

- We introduce *Minkowski attention*, a new attention formulation that yields a single decoupled attention score from content and position embeddings per token by casting them as space and time vectors score under a Minkowski inner product.
- We generalize the (3+1)D Minkowski metric to arbitrary-dimensional pseudo-Euclidean signatures and derive the pseudo-Lorentz transformations for the purpose of computing attention scores.
- We use the attention layer to create *Minkowski transformers* and we demonstrate that our decoupled attention is more powerful than coupled and decoupled Euclidean counterparts, as well as recent hyperbolic transformers, which also investigate non-linear attention.
- We show empirically that our attention outperforms standard, decoupled, and hyperbolic attention methods across text, graph, and vision benchmarks.

2 Related work

Coupled attention. The original Transformer (Vaswani et al., 2017) adds a positional signal to each token's content embedding before any projection, as do the learned absolute embeddings of BERT Devlin et al. (2019) and RoBERTa Liu et al. (2019b) and the patch embeddings of the Vision Transformer Dosovitskiy et al. (2021). The resulting logit $(E_i + T_i)^\top W_Q^\top W_K (E_j + T_j)$ expands into content-content, content-position, position-content and position-position terms, all produced by the same projections and summed before the softmax, so none of them can be weighted or recovered separately; Ke et al. (2021) find that the cross terms contribute little beyond noise. This is the design of our *Standard Attention* and vanilla ViT baselines, and of the graph transformers we compare against (Ying et al., 2021b; Jain et al., 2021; Rampasek et al., 2022; Wu et al., 2022; 2023), which attach structural encodings to node features before attention, alongside message-passing baselines carrying no positional stream at all (Kipf & Welling, 2017; Veličković et al., 2018; Wu et al., 2019).

Decoupled attention. Shaw et al. (2018) learn embeddings for pairwise offsets and inject them directly into the attention score, and Transformer-XL Dai et al. (2019) decomposes that score into content- and position-based terms to support recurrence across segments. DeBERTa He et al. (2020; 2023) makes the separation explicit, describing each token by a content and a relative-position vector and combining disentangled content-to-content, content-to-position and position-to-content attention matrices – the decoupling we target, obtained at the cost of several matrices per layer. RoPE Su et al. (2024) takes a different route, rotating queries and keys by a position-dependent angle so that a single dot product depends only on the relative offset. Our Pseudo-Lorentz Transformation (PLT) is the closest analogue of the latter, replacing the Euclidean rotation with a boost that mixes content and position rather than rotating within content alone.

Non-Euclidean attention. A wide range of works have investigated transformers beyond Euclidean space. A popular direction is transformers that operate fully in hyperbolic space (Yang et al., 2024), in products of constant-curvature spaces (Cho et al., 2024), in vision (Ermolov et al., 2022), and intrinsically in the Lorentz model (Lindeman et al., 2026). Bonev et al. (2026) generalize attention to the sphere. All of them constrain representations to a manifold of fixed curvature and pay for it in Riemannian machinery, whereas we work in the ambient $\mathbb{R}^{p,q}$ and never project onto a hyperboloid, so every operation stays a matrix product and the intrinsic/extrinsic distinction of Lindeman et al. (2026) does not bear on our construction. Their aim differs too: they use curvature to model hierarchy while still describing each token by one vector, whereas we use the signature of the metric to carry two, keeping content and position separable within a single score. We compare against non-Euclidean transformers in our experiments. Several works have explored spacetime-based representation learning for graph-structured data (Law & Lucas, 2022; Clough & Evans, 2017), but its application to general-purpose attention remains largely unexplored.

3 DECOUPLED ATTENTION IN MINKOWSKI SPACETIME

Our proposed *Minkowski attention* borrows two ingredients from hyperbolic geometry and hyperbolic deep learning: (i) the indefinite, signed inner product that underlies Minkowski spacetime and the hyperboloid model of hyperbolic space, and (ii) Lorentz boosts, the linear maps that mix "space-like" and "time-like" coordinates while preserving this inner product. We use these tools in a deliberately loose, representation-learning sense: rather than embedding tokens on a fixed-curvature hyperboloid and reasoning about geodesics, we treat every token's content and positional embeddings as the space-like and time-like halves of a vector in a generalized Minkowski space, and let a network learn, per token, how much to "boost" between the two. This yields a single attention score that keeps content and position algebraically separate by construction (Section 3.3), at the cost of the strict invariance guarantees that hold in physics or in fixed-curvature hyperbolic embeddings, a trade-off we discuss explicitly below and that we motivate empirically in Section 4.

3.1 BACKGROUND ON MINKOWSKI SPACETIME

Minkowski spacetime $\mathbb{R}^{3,1}$ replaces the positive-definite Euclidean inner product with an indefinite one: one coordinate (time) enters with the opposite sign of the remaining three (space). Instead of a fixed 3-space/1-time split, we allow arbitrary dimensions where "space" stands for a p-dimensional content embedding and "time" for a q-dimensional positional embedding.

Formally, let $\mathbb{R}^{p,q} = \mathbb{R}^p \times \mathbb{R}^q$ and write a vector $X \in \mathbb{R}^{p,q}$ as a pair $X = (X_S, X_T)$ with $X_S \in \mathbb{R}^p$ its *space-like* (content) part and $X_T \in \mathbb{R}^q$ its *time-like* (positional) part. The Lorentzian inner product in $\mathbb{R}^{p,q}$ between $X = (X_S, X_T)$ and $Y = (Y_S, Y_T)$ is

$$\langle X, Y \rangle_L = X_S^\top Y_S - X_T^\top Y_T, \tag{1}$$

with associated squared interval $\|X\|_L^2 = \langle X, X \rangle_L = \|X_S\|_2^2 - \|X_T\|_2^2$. Unlike a Euclidean norm, $\|X\|_L^2$ is not sign-definite: X is *space-like* if $\|X\|_L^2 > 0$, *time-like* if $\|X\|_L^2 < 0$, and *light-like* (or null) if $\|X\|_L^2 = 0$. Null vectors form the *light cone*, and time-like vectors split into two disconnected *sheets* depending on the sign of a distinguished time coordinate.

This same construction, restricted to $q = 1$ and to the unit time-like sheet $\|X\|_L^2 = -1$, is exactly the *hyperboloid (Lorentz) model* of hyperbolic space $\mathbb{H}^p$ used throughout the hyperbolic deep learning literature (Nickel & Kiela, 2018): distances on the hyperboloid are recovered from the Lorentzian inner product via $d_{\mathbb{H}}(x, y) = \operatorname{arccosh}(-\langle x, y \rangle_L)$, with nearby points having Lorentzian inner product close to -1 and far-apart points an increasingly negative one. Section 3.3 reuses this monotonic relationship between $\langle \cdot, \cdot \rangle_L$ and "closeness" as the intuition behind our attention score, even though our vectors are not constrained to lie on a unit hyperboloid. The linear maps $\Lambda : \mathbb{R}^{p,q} \to \mathbb{R}^{p,q}$ that preserve $\langle \cdot, \cdot \rangle_L$ – i.e. $\langle \Lambda X, \Lambda Y \rangle_L = \langle X, Y \rangle_L$ for all X, Y – form the indefinite orthogonal group $O(p, q)$, generalizing both the Lorentz group $O(3, 1)$ of special relativity and the isometry group $O(p, 1)$ of $\mathbb{H}^p$. $O(p, q)$ contains rotations acting separately within the space-like and time-like blocks, as well as *boosts*, which mix the two blocks; we use only the latter, described next.

3.2 PSEUDO-LORENTZ TRANSFORMATIONS

A Lorentz boost is the simplest transformation in $O(p, q)$ that mixes space-like and time-like coordinates. We focus on the case relevant to our attention layer, where content and position are projected into a shared dimensionality $p = q = d$, so that a single scalar mixing coefficient v can act coordinate-wise on both blocks. Given $X = (X_S, X_T) \in \mathbb{R}^{d,d}$, a boost with mixing coefficient (velocity) $v \in (0, 1)$ produces

$$X_S' = \gamma(X_S - vX_T), \qquad X_T' = \gamma(X_T - vX_S), \tag{2}$$

with $\gamma = 1/\sqrt{1 - v^2}$ chosen so that $\|X'\|_L^2 = \|X\|_L^2$ whenever v is a single scalar shared by every vector being compared. Boosts with a fixed v form a one-parameter subgroup of $O(d, d)$, exactly analogous to how physical Lorentz boosts form a one-parameter subgroup of $O(3, 1)$.

Our attention layer generalizes this construction one step further: instead of a single global v, we let *every token and every representation* (queries, keys, values) compute its own v from its content and positional norms (Section 3.3). We call the resulting maps *pseudo-Lorentz boosts*. They keep

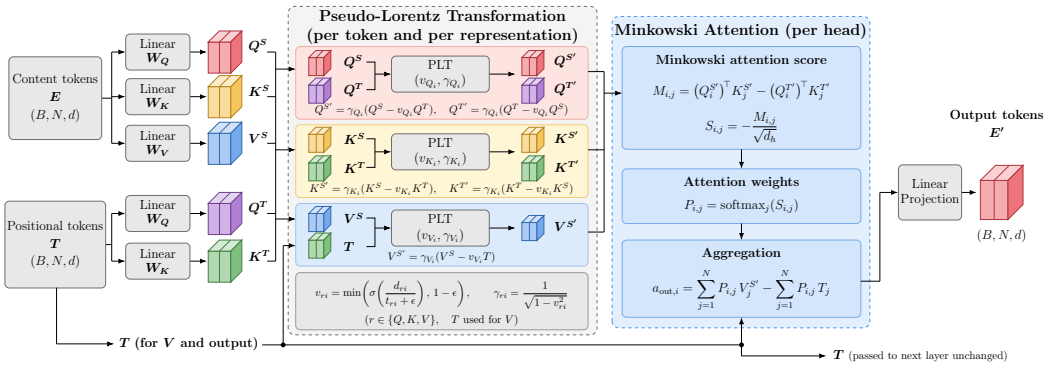

Figure 1: Overview of the Proposed Minkowski Attention.

the functional form of a boost – a smooth, invertible mixing of space-like and time-like coordinates bounded by $|v| < 1$ – but because v now varies across the tokens being compared, the exact interval-preservation property above no longer holds between two differently-boosted vectors: $\langle Q'_i, K'_j \rangle_L \neq \langle Q_i, K_j \rangle_L$ in general when $v_{Q_i} \neq v_{K_j}$. We adopt this relaxation deliberately, akin to hyperbolic learning methods that trade exact isometry for learned, input-dependent geometric operations (e.g. per-point curvature or feature-dependent Möbius maps) in exchange for expressivity. Minkowski attention is therefore best understood as *inspired by* the algebra of Minkowski spacetime and Lorentz boosts, rather than as a construction with the same invariance guarantees. We make no claim that the resulting scores are Lorentz-invariant and we evaluate the design on downstream task performance.

3.3 Transformers in Minkowski Spacetime

We now assemble these tools into an attention layer. Each token $(E_i, T_i) \in \mathbb{R}^{d,d}$ is described by a content embedding E_i and a positional embedding T_i. In a multi-head layer, W_Q, W_K and W_V act on the full hidden size and their outputs are split into H heads of dimension d_h; everything that follows describes a single head, whose index we omit for readability. In particular, the norms below are taken within a head, so each token carries one velocity per head and per representation rather than a single velocity shared across the full hidden size.

Projections. Content and position are projected with the *same* query/key/value weights, so that the resulting space-like and time-like parts share a dimensionality d_h and a boost can mix them:

$$Q_i^S = W_Q E_i + b_Q, \quad K_i^S = W_K E_i + b_K, \quad V_i^S = W_V E_i + b_V, \tag{3}$$

$$Q_i^T = W_Q T_i + b_Q, \quad K_i^T = W_K T_i + b_K. \tag{4}$$

The value stream keeps the raw positional embedding T_i as its time-like part, so positional information is aggregated without ever being projected by content-specific weights.

Norms and mixing coefficients. Each representation is then summarized by the magnitudes of its space-like and time-like parts, with T_i itself playing the time-like role for the value stream:

$$\begin{aligned} d_{Q_i} &= \|Q_i^S\|_2, & t_{Q_i} &= \|Q_i^T\|_2, \\ d_{K_i} &= \|K_i^S\|_2, & t_{K_i} &= \|K_i^T\|_2, \\ d_{V_i} &= \|V_i^S\|_2, & t_{V_i} &= \|T_i\|_2. \end{aligned} \tag{5}$$

For each $r \in \{Q, K, V\}$, the ratio of these two magnitudes is turned into a bounded velocity through a sigmoid, with a small ϵ guarding against division by zero and clipping keeping γ finite:

$$v_{r_i} = \min\left(\sigma\left(\frac{d_{r_i}}{t_{r_i} + \epsilon}\right),\, 1 - \epsilon\right), \qquad \epsilon \ll 1. \tag{6}$$

Because $d_{r_i}, t_{r_i} \geq 0$, this ratio is always non-negative, so v_{r_i} is restricted to $[\sigma(0), 1-\epsilon) = [0.5, 1-\epsilon)$ rather than the full $(-1, 1)$ range available to a general boost – each token can only be boosted

"forward", never "backward". We flag this as a design detail rather than a deliberate restriction: it is the parameterization we evaluate in Section 4, and a follow-up formulation could recover the full range by feeding a signed quantity through $2\sigma(\cdot) - 1$ instead. The scale factor $\gamma_{r_i} = 1/\sqrt{1 - v_{r_i}^2}$ follows for each representation.

Pseudo Lorentz Transformation. Queries, keys, and values are then boosted using their own token- and representation-specific (v_{r_i}, γ_{r_i}):

$$Q_i^{S'} = \gamma_{Q_i}(Q_i^S - v_{Q_i} Q_i^T), \qquad Q_i^{T'} = \gamma_{Q_i}(Q_i^T - v_{Q_i} Q_i^S), \tag{7}$$

$$K_i^{S'} = \gamma_{K_i}(K_i^S - v_{K_i} K_i^T), \qquad K_i^{T'} = \gamma_{K_i}(K_i^T - v_{K_i} K_i^S), \tag{8}$$

$$V_i^{S'} = \gamma_{V_i}(V_i^S - v_{V_i} T_i). \tag{9}$$

The pseudo-Lorentz transformation preserves the Minkowski interval of each individual query and key while allowing token-dependent boosts to alter their pairwise Lorentzian relationships. Thus, the transformation preserves each token's intrinsic geometric scale while providing adaptive content-position mixing for attention.

Minkowski attention score. We score each query-key pair with the Lorentzian inner product between their boosted representations,

$$M_{i,j} = (Q_i^{S'})^\top K_j^{S'} - (Q_i^{T'})^\top K_j^{T'}, \qquad S_{i,j} = \frac{-M_{i,j}}{\sqrt{d_h}}, \tag{10}$$

$$P_{i,j} = \mathrm{softmax}_j(S_{i,j}) = \frac{e^{S_{i,j}}}{\sum_{k=1}^{N} e^{S_{i,k}}}. \tag{11}$$

$M_{i,j}$ is a single scalar computed from two vectors that never had their content and positional coordinates fused. Writing ϕ_{r_i} for the rapidity of each boost, defined by $\tanh \phi_{r_i} = v_{r_i}$ so that $\gamma_{r_i} = \cosh \phi_{r_i}$ and $\gamma_{r_i} v_{r_i} = \sinh \phi_{r_i}$, content-content and position-position agreement contribute to $S_{i,j}$ with the sign structure of Section 3.1, weighted by $\cosh(\phi_{Q_i} - \phi_{K_j})$, while the two content-position cross terms contribute weighted by $\sinh(\phi_{Q_i} - \phi_{K_j})$ and therefore drop out exactly when query and key are boosted at the same velocity. A single attention matrix thus carries all four interactions that decoupled attention mechanisms normally obtain only by computing several attention matrices per layer, with the rapidity gap between the two tokens acting as a data-dependent mixing coefficient over them. Note that $S_{i,j}$ negates $M_{i,j}$: under the hyperboloid reading of Section 3.1, where $-\langle X, Y\rangle_L = \cosh d_{\mathbb{H}}(X, Y)$ for unit time-like vectors, $-M_{i,j}$ is monotone in a Lorentzian distance rather than in a Euclidean similarity. Since Q_i^S and Q_i^T are both linear in (W_Q, b_Q) while v_{Q_i} depends only on their norms, replacing (W_Q, b_Q) by $(-W_Q, -b_Q)$ flips the sign of $M_{i,j}$ exactly, so either convention is reachable within the same parameterization.

Aggregation. The output combines the boosted values and the (unboosted) positional embeddings across all tokens, weighted by $P_{i,j}$:

$$a_{\mathrm{out},i} = \sum_{j=1}^{N} P_{i,j} V_j^{S'} - \sum_{j=1}^{N} P_{i,j} T_j. \tag{12}$$

The per-head outputs are then concatenated and mapped back to the hidden size by the output projection (W_O, b_O):

$$E_i' = W_O \,\mathrm{concat}\big(a_{\mathrm{out},i}^{(1)}, \ldots, a_{\mathrm{out},i}^{(H)}\big) + b_O. \tag{13}$$

E_i' becomes the content embedding fed to the next Minkowski attention layer, while the positional embedding T_i is passed forward unchanged, so every layer repeats the same content/position split.

Overall, Minkowski attention replaces the single Euclidean similarity $Q_i K_j^\top$ of standard attention with a per-token pseudo-Lorentz boost followed by a Lorentzian inner product. We see this construction as taking inspiration from, rather than faithfully implementing, hyperbolic geometry and hyperbolic deep learning: the boosts are not exact isometries once v is learned per token, and the resulting scores have no guaranteed geometric interpretation as distances on a curved manifold. We nonetheless find, in Section 4, that this Lorentzian construction is a strong empirical recipe for decoupled attention, and we leave a tighter theoretical treatment – e.g. constraining v to preserve some notion of invariance, or grounding the score in an explicit hyperbolic distance – to future work.

Table 1: Performance comparison between standard and our *Minkowski Attention* under the Masked Language Modeling (MLM) pre-training objective. All models are pre-trained and fine-tuned under identical training setups, and results are reported as mean ± standard deviation over five runs. The comparison is conducted across four model configurations with varying hidden sizes, numbers of attention heads, and numbers of layers. Best results for each configuration are in **bold**.

Method	SQuAD v1.1		MNLI		SST-2	Params
	EM	F1	MNLI-m	MNLI-mm	Acc	
Hidden=256, Heads=4, Layers=6						
Standard Attention	52.53 ± 0.15	64.25 ± 0.11	66.96 ± 0.24	68.48 ± 0.17	79.54 ± 0.89	20.5M
Minkowski Attention	$\mathbf{60.62 \pm 0.27}$	$\mathbf{72.02 \pm 0.31}$	$\mathbf{71.01 \pm 0.28}$	$\mathbf{72.53 \pm 0.17}$	$\mathbf{84.17 \pm 0.45}$	20.5M
Hidden=256, Heads=4, Layers=12						
Standard Attention	61.33 ± 1.68	72.26 ± 1.61	70.00 ± 0.37	71.58 ± 0.10	82.08 ± 0.72	25.2M
Minkowski Attention	$\mathbf{67.27 \pm 0.31}$	$\mathbf{77.73 \pm 0.31}$	$\mathbf{74.50 \pm 0.14}$	$\mathbf{75.56 \pm 0.23}$	$\mathbf{85.64 \pm 0.41}$	25.2M
Hidden=384, Heads=6, Layers=12						
Standard Attention	67.75 ± 0.34	77.78 ± 0.24	74.19 ± 0.27	75.58 ± 0.25	85.78 ± 0.66	45M
Minkowski Attention	$\mathbf{71.74 \pm 0.24}$	$\mathbf{81.12 \pm 0.15}$	$\mathbf{76.64 \pm 0.06}$	$\mathbf{77.36 \pm 0.34}$	$\mathbf{85.78 \pm 0.21}$	45M
Hidden=768, Heads=12, Layers=12						
Standard Attention	75.27 ± 0.18	83.76 ± 0.16	76.95 ± 0.04	77.93 ± 0.28	85.82 ± 0.83	110M
Minkowski Attention	$\mathbf{76.87 \pm 0.14}$	$\mathbf{85.47 \pm 0.16}$	$\mathbf{78.96 \pm 0.13}$	$\mathbf{79.89 \pm 0.15}$	$\mathbf{88.30 \pm 0.59}$	110M

4 EXPERIMENTS

4.1 SETUP

Pre-training setup. Unless otherwise stated, all models are pre-trained on the English Wikipedia corpus (2,500M words) (Wikipedia contributors, 2024) using the `albert-base-v2` tokenizer for 125,000 optimization steps and an effective batch size of 128 across (2 or 4 A40) GPUs with the Masked Language Modeling (MLM) objective. We employ a warm-up phase of 10,000–12,500 steps, followed by cosine learning-rate decay, with a peak learning rate of 3×10^{-4}. These settings follow widely adopted practices inspired by BERT-style pre-training (Devlin et al., 2019). For comparison, Devlin et al. (2019) was trained with an effective batch size of 256 for 1,000,000 steps.

To ensure a fair comparison, both the standard attention baseline and the proposed Minkowski Attention are trained under identical optimization settings, data pipeline, and model capacity. Following RoBERTa (Liu et al., 2019b), we omit the Next Sentence Prediction (NSP) objective during pre-training, as removing it has been shown to match or improve downstream performance.

Downstream evaluation setup. For the text experiments, we evaluate the downstream performance of the pre-trained models on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018), including MNLI and SST-2, as well as SQuAD v1.1 (Rajpurkar et al., 2016). We follow the standard fine-tuning protocol and hyperparameter ranges of BERT Devlin et al. (2019). All models are fine-tuned for 3 epochs using a batch size of 32, with the learning rate selected from $\{5 \times 10^{-5}, 4 \times 10^{-5}, 3 \times 10^{-5}, 2 \times 10^{-5}\}$. For a fair comparison, identical optimization protocol are used for all approaches. Each downstream experiment is repeated over five random seeds, with results reported as mean ± standard deviation across runs. For experiments on graphs and images, we report the standard classification accuracy.

4.2 COMPARATIVE ANALYSIS

Language transformers. In the first experiment, we compare against standard attention across three well-known benchmarks, namely SQuAd v1.1, MNLI, and SST-2. The comparison is shown in Table 1, which shows that across all benchmarks and model sizes, *Minkowski attention* performs better by a clear margin. In Table 2, we draw a comparison to the decoupled attention of DeBERTa (He et al., 2020), as well as RoPE (Su et al., 2024) and ALiBi (Press et al., 2021), which propose more expressive positional encodings. Our approach outperforms all. We note that the latter approaches are not competitors. Any advance regarding positional embeddings also helps our approach and we consider integrating such works in our formulation as an exciting future research.

Table 2: Comparative analysis to decoupled attention mechanisms (DeBERTa) and approaches with more powerful position embeddings (RoPE, ALiBi). All models use the same architecture (hidden size 256, 4 attention heads, and 6 layers). GFLOPs are measured for a single forward pass with batch size 1 and sequence length 512; $\%\Delta$ denotes the relative computational overhead over standard self-attention. Our approach outperforms all baselines.

Method	SQuAD v1.1		MNLI		SST-2	Params	GFLOPs	
	EM	F1	MNLI-m	MNLI-mm			Value	% Δ
MA (Parabolic)	48.45 ± 1.10	59.63 ± 1.16	69.11 ± 0.20	70.57 ± 0.18	82.93 ± 0.65	*20.5M*	*16.86*	*+16.7%*
DeBERTa (APE)	50.64 ± 0.21	62.20 ± 0.29	66.39 ± 0.02	67.34 ± 0.49	81.90 ± 0.25	*20.5M*	*16.86*	*+16.7%*
RoPE	55.65 ± 3.78	66.35 ± 4.14	69.76 ± 0.03	70.78 ± 0.18	83.94 ± 0.39	**20.4M**	**14.44**	**0.0%**
ALiBi	50.79 ± 0.26	62.29 ± 0.29	67.42 ± 0.14	68.93 ± 0.16	82.89 ± 0.22	**20.4M**	**14.44**	**0.0%**
DeBERTa-V1	60.48 ± 0.18	71.61 ± 0.33	70.41 ± 0.19	72.16 ± 0.16	83.22 ± 0.46	21.5M	17.66	*+22.3%*
Minkowski Attention	$\mathbf{60.62 \pm 0.27}$	$\mathbf{72.02 \pm 0.31}$	$\mathbf{71.01 \pm 0.28}$	$\mathbf{72.53 \pm 0.17}$	$\mathbf{84.17 \pm 0.45}$	*20.5M*	*16.86*	*+16.7%*

Table 3: Node classification (accuracy %) on graph benchmarks. Baseline numbers are taken from Table 2 of Yang et al. (2024). Best result in bold, second-best underlined.

Method	Airport	Cora	CiteSeer	PubMed
HGNN (Liu et al., 2019a)	84.7 ± 1.0	77.1 ± 0.8	70.0 ± 1.0	78.3 ± 1.2
HGCN (Chami et al., 2019)	89.3 ± 1.2	76.5 ± 0.6	68.0 ± 0.6	78.0 ± 1.0
GraphFormer (Ying et al., 2021a)	88.1 ± 1.2	60.0 ± 0.5	61.4 ± 0.6	73.3 ± 0.7
GraphTrans (Jain et al., 2021)	94.3 ± 0.6	77.6 ± 0.8	65.1 ± 1.4	77.5 ± 0.7
GraphGPS (Rampasek et al., 2022)	94.5 ± 0.9	73.0 ± 1.4	62.0 ± 1.5	72.8 ± 1.4
FPS-T (Cho et al., 2023)	$\underline{96.0 \pm 0.6}$	82.3 ± 0.7	70.0 ± 0.7	78.5 ± 0.6
HAN (Gulcehre et al., 2019)	92.9 ± 0.6	83.1 ± 0.5	72.4 ± 0.5	79.0 ± 0.6
F-HNN (Chen et al., 2022)	93.0 ± 0.7	81.0 ± 0.7	71.2 ± 0.4	77.5 ± 0.8
NodeFormer (Wu et al., 2022)	80.2 ± 0.6	82.2 ± 0.9	72.5 ± 1.1	79.9 ± 1.0
SGFormer (Wu et al., 2023)	92.9 ± 0.5	83.2 ± 0.9	72.2 ± 0.3	80.0 ± 0.8
Hypformer (Yang et al., 2024)	95.0 ± 0.5	$\underline{85.0 \pm 0.3}$	$\underline{73.3 \pm 0.4}$	$\underline{81.3 \pm 0.3}$
Minkowski Attention	$\mathbf{96.53 \pm 0.48}$	$\mathbf{85.94 \pm 0.36}$	$\mathbf{74.30 \pm 0.44}$	$\mathbf{81.46 \pm 0.36}$

Graph transformers. To demonstrate the general nature of our approach, we adapt it to transductive node classification within the Hypformer (Yang et al., 2024) framework, replacing only the hyperbolic attention encoder while keeping the graph-convolution branch, decoder, and all hyperparameters unchanged. Following Yang et al. (2024), positional information is obtained from a second projection of the raw node features, while final prediction combine the attention and GNN branches using the same per dataset weighting coefficient λ as in Hypformer. Table 3 reports accuracy on Airport, Cora, CiteSeer, and PubMed against GNN, hyperbolic, and graph Transformer baselines reported in Hypformer (Yang et al., 2024). Minkowski attention outperforms all baselines.

Vision transformers. Finally, we compare standard attention and Minkowski Attention using ViT architectures across five image classification benchmarks, covering both low- and high-resolution datasets. The baseline uses a vanila Pre-LN ViT, while Minkowski Attention follows the Post-LN architecture described in Section 3.3. Table 4 presents this comparison. We additionally analyze the effect of the final layer normalization on the performance of the vanilla ViT across these datasets. Our results show that Minkowski Attention consistently outperforms standard attention across all benchmark datasets under identical training settings, most notably on smaller-scale datasets with a larger number of classes. For detailed implementation details and hierarchical error analysis, we refer the reader to Section A.2 and A.6 respectively.

4.3 Ablation Studies

We investigate the contribution of the principal design choices underlying Minkowski Attention. In particular, we analyze the role of the *Pseudo-Lorentz Transformation (PLT)* 3.3 and the *Spatio-Temporal Aggregation (STA)* 3.3 module. The results show that both components are individually important and jointly complementary: removing either element leads to a measurable drop in downstream performance. The corresponding comparisons are reported in Table 5. In the appendices, we

Table 4: Performance comparison between standard Attention and the proposed Minkowski Attention in ViT architectures. All models are evaluated under identical architectural and input configurations; full experimental details are provided in Appendix A.2.

Method	FLN	CIFAR-10 ↑	CIFAR-100 ↑	TinyImageNet ↑	ImageNette ↑	ImageWoof ↑
ViT (Vanilla)	✓	90.53 ± 0.37	66.25 ± 0.58	51.89 ± 1.01	82.41 ± 0.31	58.51 ± 0.35
ViT (Vanilla)	✗	91.33 ± 0.38	67.53 ± 0.16	53.54 ± 0.20	80.98 ± 0.81	58.18 ± 0.70
MViT (Post-LN)	NA	$\mathbf{93.78 \pm 0.31}$	$\mathbf{75.50 \pm 0.31}$	$\mathbf{58.80 \pm 0.75}$	$\mathbf{84.64 \pm 0.22}$	$\mathbf{62.11 \pm 1.21}$

Table 5: Ablation study on the key components of the proposed attention under the Masked Language Modeling (MLM) pre-training objective. **STA** denotes spatio-temporal aggregation and **PLT** denotes pseudo-Lorentz transformation. All models share the same architecture (hidden size 256, 4 attention heads, 6 layers) to ensure a fair comparison. The table highlights the individual and combined impact of these components on downstream tasks including SQuAD v1.1 (Exact Match and F1), MNLI (matched and mismatched accuracy), and SST-2 (classification accuracy).

STA	PLT	Method	SQuAD v1.1		MNLI		SST-2
			EM	F1	MNLI-m	MNLI-mm	Acc
✗	✗	Standard Attention	52.53 ± 0.15	64.25 ± 0.11	66.96 ± 0.24	68.48 ± 0.17	79.54 ± 0.89
✗	✗	Minkowski Attention	53.14 ± 0.55	64.27 ± 0.52	68.69 ± 0.13	69.96 ± 0.39	83.62 ± 1.05
✓	✗	Minkowski Attention	56.11 ± 2.98	67.24 ± 3.06	70.06 ± 0.21	71.61 ± 0.12	83.64 ± 0.94
✗	✓	Minkowski Attention	55.12 ± 0.59	66.44 ± 0.52	69.91 ± 0.53	71.13 ± 0.12	84.17 ± 0.55
✓	✓	**Minkowski Attention**	$\mathbf{60.62 \pm 0.27}$	$\mathbf{72.02 \pm 0.31}$	$\mathbf{71.01 \pm 0.28}$	$\mathbf{72.53 \pm 0.17}$	$\mathbf{84.17 \pm 0.45}$

provide additional ablation studies, including gradient norms in A.4, evolution of attention entropy across layers and heads in A.5, and stability of the training loss for both pre- and post-LN in A.7.

4.4 Head Diversity Through Relative Attention Patterns

We analyze how attention weights are distributed as a function of relative token distance, providing insight into the inductive biases learned by each architecture. Examining attention as a function of relative displacement allows us to study locality, long-range dependency modeling, positional preferences, and specialization across attention heads. For this analysis, we use the final layer of the pre-trained Transformer models and evaluate both standard attention and Minkowski Attention on a batch of sequences sampled from the English Wikipedia corpus. Each sequence is truncated to length $L = 512$, and for every attention head we compute the mean attention weight associated with each relative token position $(j - i)$, where i denotes the query position and j the key position. The reported curves summarize how strongly each head attends to nearby tokens, distant context, and directional positions to the left or right of the current token, see Figure 2. Both models use a hidden size of 384, 6 attention heads, and 12 layers, resulting in approximately 45M parameters.

The results provide clear evidence that the proposed Minkowski Attention exhibits substantially richer head diversity than the standard baseline. In standard attention, most heads display strongly local behavior with attention concentrated near $(j-i) = 0$, together with largely symmetric patterns across left and right contexts. This suggests that several heads learn similar locality-driven behaviors with limited specialization. In contrast, Minkowski Attention produces more heterogeneous and functionally differentiated head patterns. Such diversity suggests that different heads allocate themselves to complementary roles rather than collapsing toward a common behavior.

These results indicate that Minkowski Attention simultaneously preserves bidirectional contextual modeling, enhances access to long-range dependencies, and promotes stronger head specialization. In Figure 3 we also visualize qualitative examples of the attention maps in the vision transformer. These figures show that standard attention often focuses on non-discriminative tokens such as the background, while our attention maps concentrate on the key visuals of the objects to be recognized, suggesting stronger localization power and potential for better explainability.

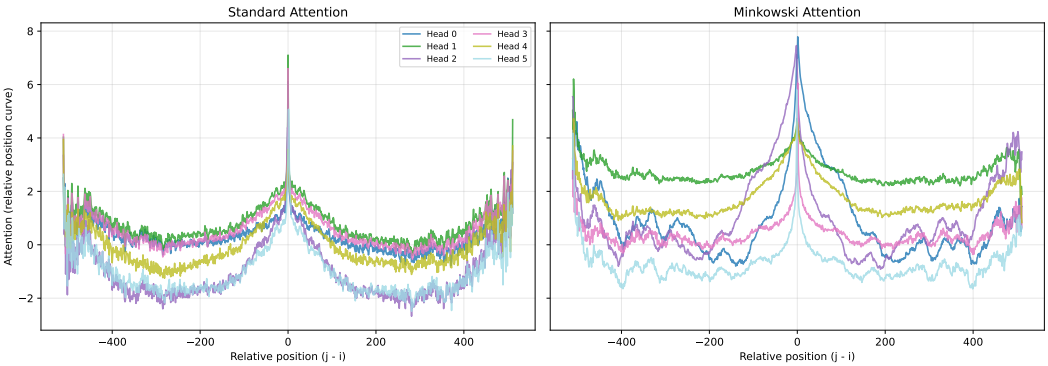

Figure 2: Per-head relative attention as a function of relative token position $(j - i)$, computed from the final Transformer layer using a single batch of Wikipedia data with sequence length $L = 512$. We compare standard attention and Minkowski Attention to analyze how each method distributes attention across relative positions. The figure provides insight into head-wise locality, long-range dependency modeling, and positional bias, highlighting differences in how the two architectures allocate attention mass over nearby and distant tokens.

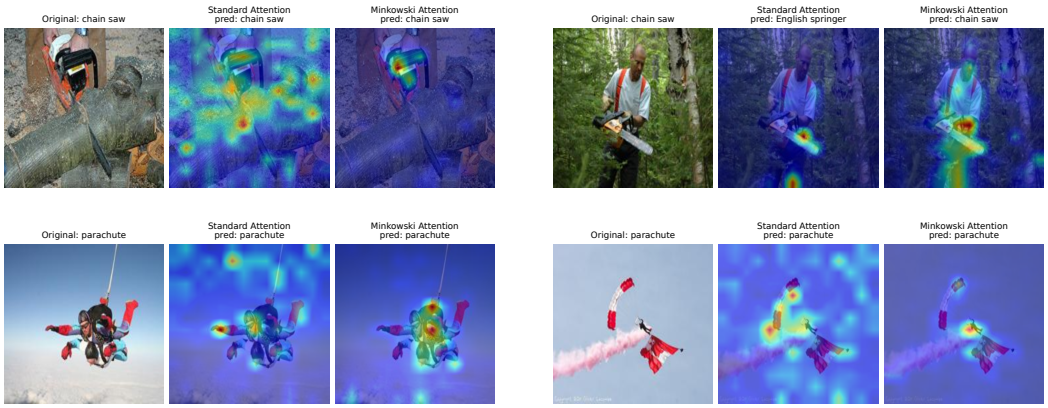

Figure 3: Qualitative results on the ImageNette validation set comparing standard attention and Minkowski Attention. The original images and the corresponding predicted class names are shown, with the predicted class names highlighted for clarity. We find that standard attention often assigns high attention to non-discriminative tokens such as the background, while our approach can uncover the object location, even in scenes with confounding objects, such as a tree for the object chain saw.

5 Conclusions

We introduce *Minkowski Attention*, a geometry-aware attention formulation that incorporates a Lorentzian spacetime structure to jointly model token content and position. Across vision and language benchmarks, Minkowski Attention consistently achieves strong performance compared with standard attention and widely adopted alternatives such as RoPE, ALiBi, and DeBERTa. Our experiments further demonstrate its applicability beyond sequence modeling through node classification, and ablation and qualitative studies provide evidence that the underlying spacetime geometry contributes to the observed improvements and encourages more semantically focused attention patterns. Overall, the results highlight the potential of Lorentzian geometry as a general framework for designing attention mechanisms across modalities and domains.

Limitations and future work. We have focused our experiments on extending the canonical attention formulation, since this remains the ubiquitous approach in modern networks. As future work, we are interested in incorporating advances from positional embedding representations into our framework, and generalize our approach to state space models such as Mamba.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Boris Bonev, Max Rietmann, Andrea Paris, Alberto Carpentieri, and Thorsten Kurth. Attention on the sphere. volume 38, pp. 61662–61688, 2026.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. volume 32, 2019.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5672–5686, 2022.

Sungjun Cho, Seunghyuk Cho, Sungwoo Park, Hankook Lee, Honglak Lee, and Moontae Lee. Curve your attention: Mixed-curvature transformers for graph representation learning. *arXiv preprint arXiv:2309.04082*, 2023.

Sungjun Cho, Seunghyuk Cho, Sungwoo Park, Hankook Lee, Honglak Lee, and Moontae Lee. Curve your attention: Mixed-curvature transformers for graph representation learning. 2024. URL https://openreview.net/forum?id=AN5uo4ByWH.

James R Clough and Tim S Evans. Embedding graphs in lorentzian spacetime. *PloS one*, 12(11): e0187301, 2017.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2978–2988, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7399–7409. IEEE, 2022.

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. Hyperbolic attention networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJxHsjRqFQ.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Paras Jain, Zhanghao Wu, Matthew A. Wright, Azalia Mirhoseini, Joseph E. Gonzalez, and Ion Stoica. Representing long-range context for graph neural networks with global attention. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=nYz2_BbZnYk.

Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=09-528y2Fgf.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Marc Law. Ultrahyperbolic neural networks. *Advances in neural information processing systems*, 34:22058–22069, 2021.

Marc T Law and James Lucas. Spacetime representation learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Mees Lindeman, Gowreesh Mago, Pritam Mishra, and Pascal Mettes. Lovit: Intrinsic lorentz vision transformer. In *Beyond Euclidean Workshop*, 2026. URL https://openreview.net/forum?id=oyBOEcoSjD.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. volume 32, 2019a.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, pp. 3779–3788. PMLR, 2018.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Ladislav Rampasek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=lMMaNf6oxKM.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, 2018.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 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*, pp. 353–355, 2018.

Wikipedia contributors. English wikipedia dump. https://dumps.wikimedia.org/enwiki/, 2024. [Online; accessed April 2026].

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6861–6871. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/wu19e.html.

Qitian Wu, Wentao Zhao, Zenan Li, David Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Simplifying and empowering transformers for large-graph representations. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=R4xpvDTWkV.

Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3770–3781, 2024.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021a.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888. Curran Associates, Inc., 2021b. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f1c1592588411002af340cbaedd6fc33-Paper.pdf.

A Appendix

A.1 Architectural Sensitivity and Optimization Stability Setup.

To study robustness under accelerated optimization settings, we conduct an additional series of controlled pre-training experiments on the English Wikipedia corpus using learning rates of 3×10^{-4}, 6×10^{-4}, and 9×10^{-4}. Each model is trained for 10,000 optimization steps with a 1,000-step warm-up phase, followed by cosine learning-rate decay.

This protocol is used exclusively for the analyses in Sections A.7 and A.9, where our objective is to examine architectural sensitivity, convergence behavior, and training stability as the learning rate is progressively increased to test whether models can sustain faster learning without instability. These experiments provide a controlled setting for comparing the robustness of standard attention and Minkowski Attention under increasingly larger step sizes.

A.2 Training Setup for ViT on Vision Classification Benchmarks

Since transformers are known to require large-scale pre-training to generalize well, training them from scratch on small datasets is particularly susceptible to overfitting. We therefore adopt a 7-layer, 8-head encoder architecture with CutMix and MixUp ($\alpha = 0.2$) augmentation across all datasets, for both the vanilla ViT and the proposed MViT (Minkowski Vision Transformer). For a fair comparison, all models are trained for 200 epochs on each of the five benchmark datasets. For lower-resolution datasets, CIFAR-10 and CIFAR-100 use a patch size of 4×4 on 32×32 images, while Tiny-ImageNet uses a patch size of 8×8 on 64×64 images; both settings yield a token sequence length of $64+1$ (including the CLS token). For the higher-resolution datasets, ImageNette and ImageWoof (originally 160×160), we resize images to 224×224 and use a patch size of 16×16 to match the input resolution and patch configuration used in Dosovitskiy et al. (2021). All models reported in Table 4 are trained with a batch size of 128 using the AdamW optimizer, a weight decay of 0.05, and cosine annealing. We use a learning rate of 3×10^{-4} for the smaller datasets (CIFAR-10, CIFAR-100, and Tiny-ImageNet) across all models, 2×10^{-4} for ImageNette, and 1.5×10^{-4} for ImageWoof.

A.3 Token-wise Minkowski Interval Invariance

We show that the proposed Pseudo-Lorentz Transformation (PLT) preserves the Minkowski interval of each token's content–position pair, even though the boost velocity is itself a learned function of that pair's own representation.

Setup. Fix an attention head of dimension d_h. For a given token, let $x \in \mathbb{R}^{d_h}$ denote its content representation and $t \in \mathbb{R}^{d_h}$ its paired positional ("time") representation within that head. the velocity is defined as,

$$v(x,t) \;=\; \sigma\!\left(\frac{\|x\|}{\|t\|}\right), \qquad v(x,t) \in [0,1), \tag{14}$$

where $\sigma(\cdot)$ denotes the sigmoid function, so that $v(x,t) < 1$ strictly for all finite x, t with $\|t\| > 0$. The associated Lorentz factor is defined as,

$$\gamma(x,t) \;=\; \frac{1}{\sqrt{1 - v(x,t)^2}}. \tag{15}$$

The Pseudo-Lorentz Transformation is then given by

$$x' \;=\; \gamma\,(x - v\,t), \qquad t' \;=\; \gamma\,(t - v\,x). \tag{16}$$

We define the *Minkowski interval* of a content–position pair (x,t) as

$$I(x,t) \;\triangleq\; \|x\|^2 - \|t\|^2. \tag{17}$$

Theorem 1 (Token-wise Interval Invariance)**.** *Let $x, t \in \mathbb{R}^{d_h}$ with $t \neq 0$, and let $v = v(x,t)$ be* any *scalar-valued function of x, t satisfying $v \in [0,1)$, with γ as defined in Equation 15. Let (x', t') be given by the transformation in Equation 16. Then*

$$I(x',t') = I(x,t). \tag{18}$$

Proof. Expanding $\|x'\|^2$ and $\|t'\|^2$ using Equation 16:

$$\|x'\|^2 = \gamma^2 \left(\|x\|^2 - 2v\langle x, t\rangle + v^2\|t\|^2\right), \tag{19}$$

$$\|t'\|^2 = \gamma^2 \left(\|t\|^2 - 2v\langle x, t\rangle + v^2\|x\|^2\right). \tag{20}$$

Subtracting Equation 20 from Equation 19, the cross terms $-2v\langle x, t\rangle$ cancel identically:

$$\|x'\|^2 - \|t'\|^2 = \gamma^2\Big[\left(\|x\|^2 - v^2\|x\|^2\right) - \left(\|t\|^2 - v^2\|t\|^2\right)\Big] \tag{21}$$

$$= \gamma^2(1 - v^2)\left(\|x\|^2 - \|t\|^2\right). \tag{22}$$

By definition, $\gamma^2 = (1 - v^2)^{-1}$, hence $\gamma^2(1 - v^2) = 1$, giving

$$\|x'\|^2 - \|t'\|^2 \;=\; \|x\|^2 - \|t\|^2, \tag{23}$$

i.e., $I(x', t') = I(x, t)$. Note that the proof makes no assumption on how v was derived from x, t — the identity $\gamma^2(1 - v^2) = 1$ holds by construction of γ for *any* $v \in [0, 1)$, so the result holds in particular for the content-dependent velocity of Equation 14. □

Corollary 1 (Application to disentangled Q/K/V)**.** *In the proposed self-attention mechanism, each of the query, key, and value streams is boosted independently as its own content–position pair:* (q, t_q)*,* (k, t_k)*, and* $(\hat{v}, t_v)$*,*[1] *each with its own self-derived velocity* $v_q, v_k, v_{\hat{v}}$ *computed via Equation 14. By Theorem 1, applied independently to each pair,*

$$I(q', t'_q) = I(q, t_q), \qquad I(k', t'_k) = I(k, t_k), \qquad I(\hat{v}', t'_v) = I(\hat{v}, t_v), \tag{24}$$

for every token and every attention head. That is, the boost preserves the Minkowski interval of every token's own query, key, and value representation.

Remark 1 (Scope of the invariance)**.** *Theorem 1 and Corollary 1 establish invariance of each token's own content–position interval under its self-derived boost. They do* not *assert that the cross-token quantity computed during attention,* $\langle q'_i, k'_j\rangle - \langle t'_{q,i}, t'_{k,j}\rangle$ *for* $i \neq j$*, is invariant under the boosts, since tokens* i *and* j *generally have distinct velocities* $v_i \neq v_j$*. Writing* $x_1 = q_i$*,* $t_1 = t_{q,i}$*,* $x_2 = k_j$*,* $t_2 = t_{k,j}$*, direct expansion gives*

$$x'_1x'_2 - t'_1t'_2 \;=\; \gamma_1\gamma_2\Big[(1 - v_1v_2)(x_1x_2 - t_1t_2) \;+\; (v_1 - v_2)(x_1t_2 - t_1x_2)\Big], \tag{25}$$

which reduces to the classical invariant bilinear form only in the special case $v_1 = v_2$*, recovering the standard two-frame Lorentz invariance of special relativity. When* $v_1 \neq v_2$*, as is generically the case here since* v *is content-dependent per token, the term* $(v_1 - v_2)(x_1t_2 - t_1x_2)$ *in Equation 25 does not vanish. We view this as a deliberate design choice rather than a limitation to be corrected: allowing each token to define its own local frame, rather than enforcing a single shared frame across all tokens, is precisely what makes the strength of content–position coupling content-adaptive rather than fixed by position alone.*

A.4 Optimization Dynamics: Loss Convergence and Gradient Stability

In this section, we compare the optimization behavior of the proposed **Minkowski Attention** against standard attention by analyzing training loss trajectories and gradient norms during pre-training. These diagnostics provide insight into convergence speed, update efficiency, and numerical stability throughout optimization.

As shown in Figures 5 and 4, standard attention exhibits noticeably slower loss reduction during the warm-up phase. This behavior is accompanied by relatively small gradient norms, typically in the range of 0.4–0.5, suggesting weaker parameter updates during the early stages of training. In contrast, Minkowski Attention reaches substantially larger gradient norms during warm-up, approaching values close to 1.0, which correlates with faster and more effective optimization.

Beyond warm-up, Minkowski Attention continues to maintain gradient norms tightly concentrated around 1.0 with minimal fluctuation throughout training. Such behavior is desirable, as it indicates

[1] We use $\hat{v}$ for the value stream to avoid overloading v, which denotes the boost velocity throughout.

stable gradient flow and consistent optimization dynamics. By comparison, standard attention remains closer to 0.8 on average and exhibits several noticeable spikes in gradient magnitude during training, indicating noisier updates and a less stable optimization trajectory.

Overall, these results suggest that incorporating spacetime geometry into attention not only improves downstream performance, but also yields a smoother and more robust training process. The proposed Minkowski Attention appears to facilitate stronger early learning signals and more controlled optimization dynamics over the course of pre-training.

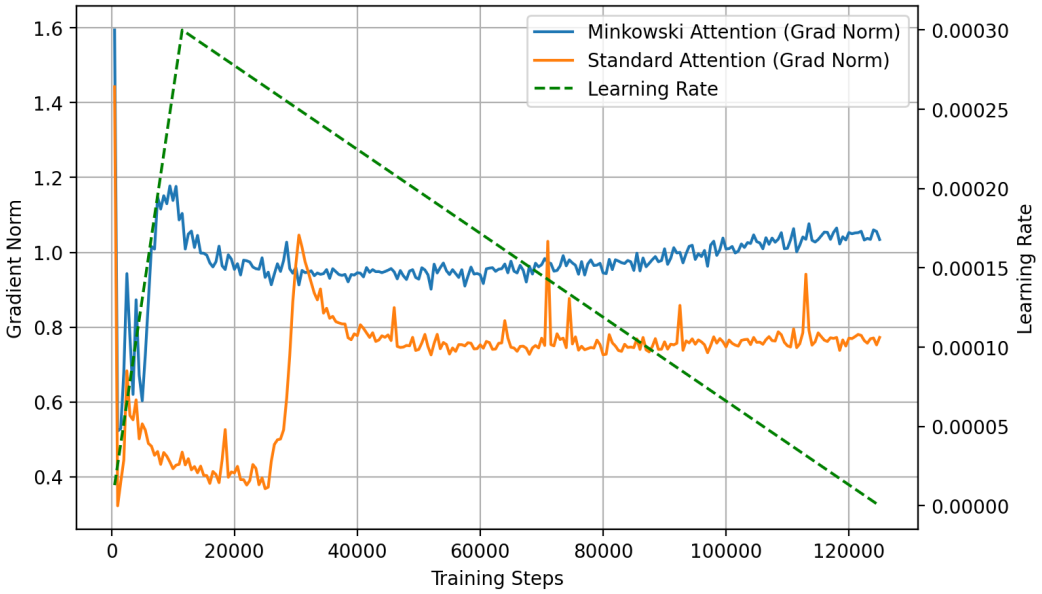

Figure 4: Comparison of gradient norms during Masked Language Modeling (MLM) pre-training for **Standard Attention** and the proposed **Minkowski Attention**. The figure also overlays the learning-rate schedule used during training in order to visualize the warm-up phase and the subsequent cosine annealing decay.

A.5 Evolution of Attention Entropy Across Layers and Heads

In this section, we investigate how Shannon entropy evolves across network depth and attention heads in Standard Attention and Minkowski Attention. Entropy provides a useful measure of how concentrated or diffuse the attention distribution is: lower entropy indicates sharper, more deterministic attention patterns, whereas higher entropy reflects a broader allocation of attention mass across tokens. Studying this quantity across layers offers insight into information flow, specialization, and potential optimization bottlenecks within the Transformer.

For this analysis, we randomly sample a batch of 128 sequences from the Wikipedia corpus and compute the normalized mean attention entropy for each head at every layer. We first summarize the results in Figure 6, which presents comparative heatmaps over all layers and heads for both Standard Attention and Minkowski Attention. To further highlight where the two models differ, Figure 7 shows the heatmap of entropy differences between the two methods.

We then provide a finer-grained view in Figure 8, where each subplot corresponds to one Transformer layer (Layers 0–11) and compares the entropy of all six heads between Standard and Minkowski Attention. The shaded region denotes the standard deviation across the batch. Complementarily, Figure 9 reorganizes the same analysis by head: six subplots are shown, one for each head, where entropy is evolved across layer depth from Layer 0 to Layer 11, again with standard deviation bands.

Several consistent patterns emerge from these figures. Standard Attention exhibits a pronounced reduction in entropy within the middle layers, particularly around Layers 3–7, and this trend ap-

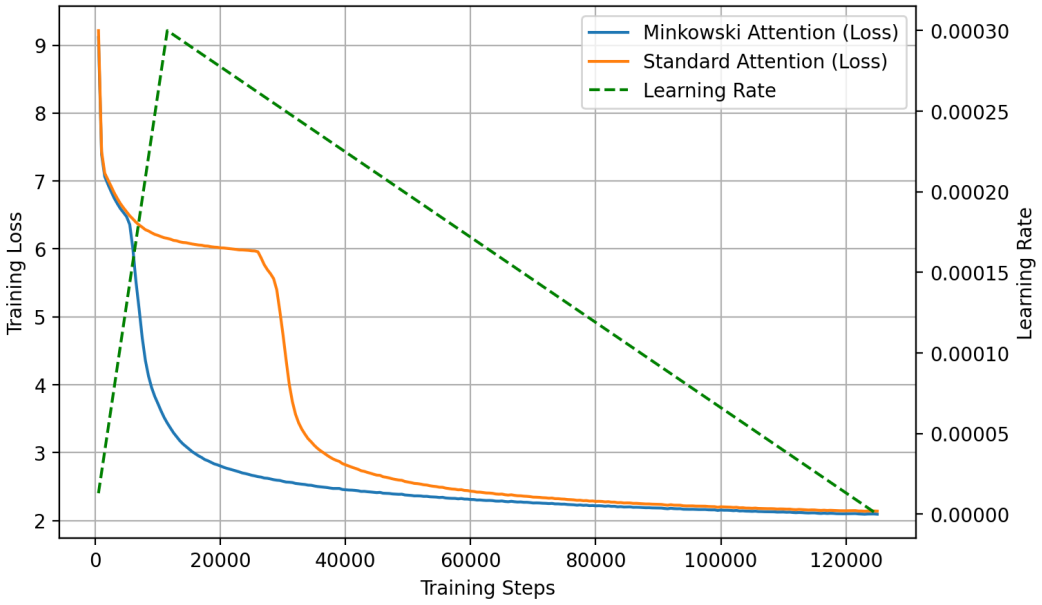

Figure 5: Comparison of training loss convergence during Masked Language Modeling (MLM) pre-training for **Standard Attention** and the proposed **Minkowski Attention**. The learning-rate schedule is also shown to illustrate the optimization phases, including the initial warm-up period followed by cosine annealing decay.

pears across multiple heads. In contrast, Minkowski Attention shows a smoother and more gradual decrease in entropy as depth increases, indicating a more progressive refinement of attention rather than an abrupt collapse.

We hypothesize that the sharp mid-layer entropy drop in Standard Attention may reflect a form of *entropy collapse*, where attention becomes overly deterministic too early in the network. Such behavior can create an information bottleneck, reducing head diversity and limiting the ability of intermediate layers to integrate broader contextual signals. The subsequent rise in entropy near the final layers may then represent a late-stage correction. By comparison, the more stable entropy trajectory of Minkowski Attention suggests desirable information propagation and a more balanced specialization process throughout depth.

A.6 Hierarchical Error Analysis of Minkowski Attention

Table 6 evaluates whether the errors of MViT respect the semantic class hierarchy, following the protocol and label hierarchies of Lindeman et al. (2026). Under identical architecture size and training settings, MViT improves over the Euclidean ViT on every metric on both datasets, reducing the tree-induced error by 28% on CIFAR-100 (0.839 vs. 1.168) and by 16% on Tiny-ImageNet (3.999 vs. 4.774). These gains reflect both fewer errors and, on CIFAR-100, closer ones: restricted to misclassified images, 37% of MViT's mistakes remain within the correct superclass, compared with 31% for the Euclidean ViT, whereas on Tiny-ImageNet the severity of the remaining errors is comparable across the two models. MViT also attains lower TIE and LCA than the values reported for LoViT, a fully hyperbolic model with roughly seven times more parameters, although these were obtained under a different training pipeline. One possible explanation lies in the structure of the pseudo-Euclidean space in which Minkowski attention operates. The space is flat, but its metric is indefinite. As a result, the level sets of the Minkowski quadratic form are not spheres, as in Euclidean space, but pseudo-hyperbolic hypersurfaces (Law, 2021), closely related to the hyperboloid model used by hyperbolic networks. Minkowski attention may therefore share some of the geometric properties that make hyperbolic models suited to hierarchies, without constraining tokens to a curved manifold, and the Lorentz boost applied to queries and keys preserves the Minkowski norm of each

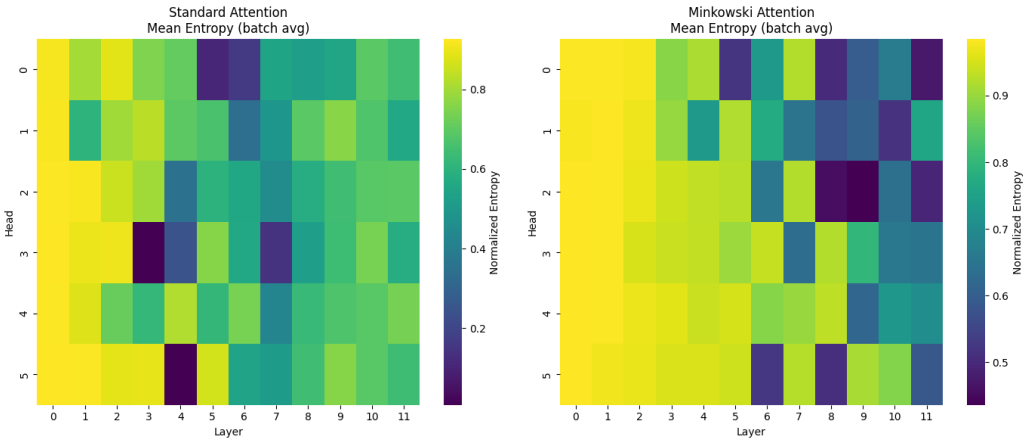

Figure 6: Comparison of normalized mean attention entropy between **Standard Attention** and the proposed **Minkowski Attention**, averaged over a mini-batch of size 128 across all heads and layers. Attention entropy is used as a measure of distributional sharpness versus dispersion, where lower entropy indicates more concentrated attention and higher entropy reflects broader allocation of attention mass. The figure provides a layer-wise and head-wise view of how the two architectures differ in their attention patterns, offering insight into representation diversity, specialization of heads, and the evolution of attention structure throughout the network depth.

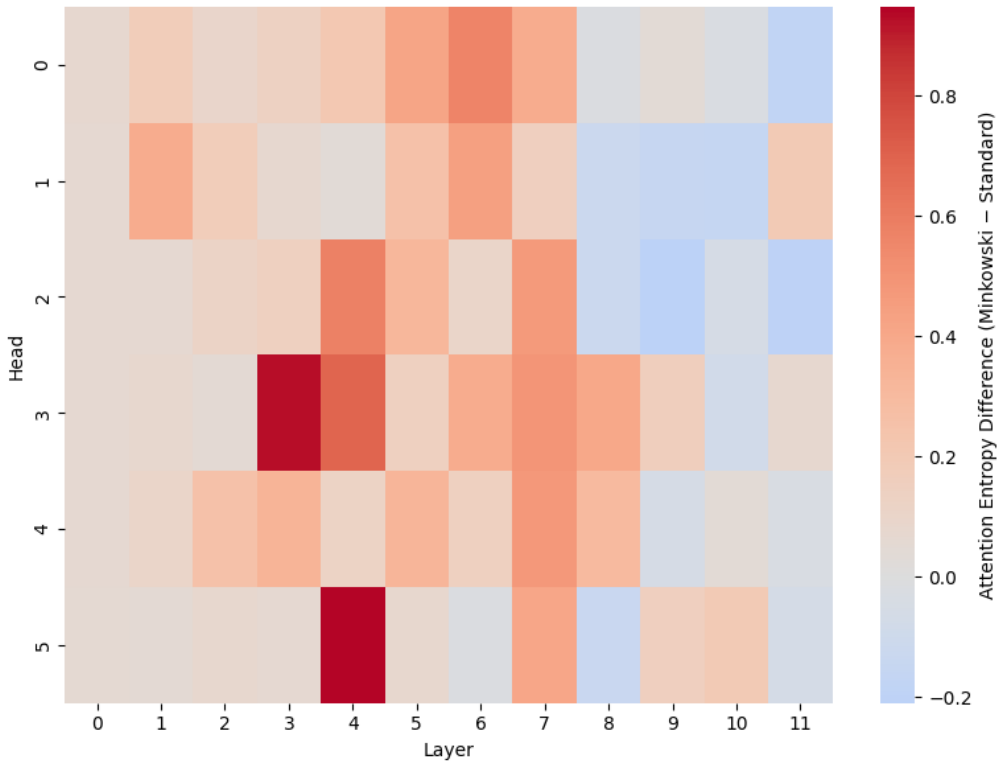

Figure 7: Difference heatmap of normalized mean attention entropy, computed as **Minkowski Attention** minus **Standard Attention**, averaged over a mini-batch of size 128 across all heads and layers. Positive values indicate heads or layers where Minkowski Attention exhibits higher entropy (more distributed attention), while negative values indicate lower entropy (more concentrated attention) relative to Standard Attention. This visualization highlights where the proposed geometry-aware mechanism most strongly alters attention behavior throughout the network depth.

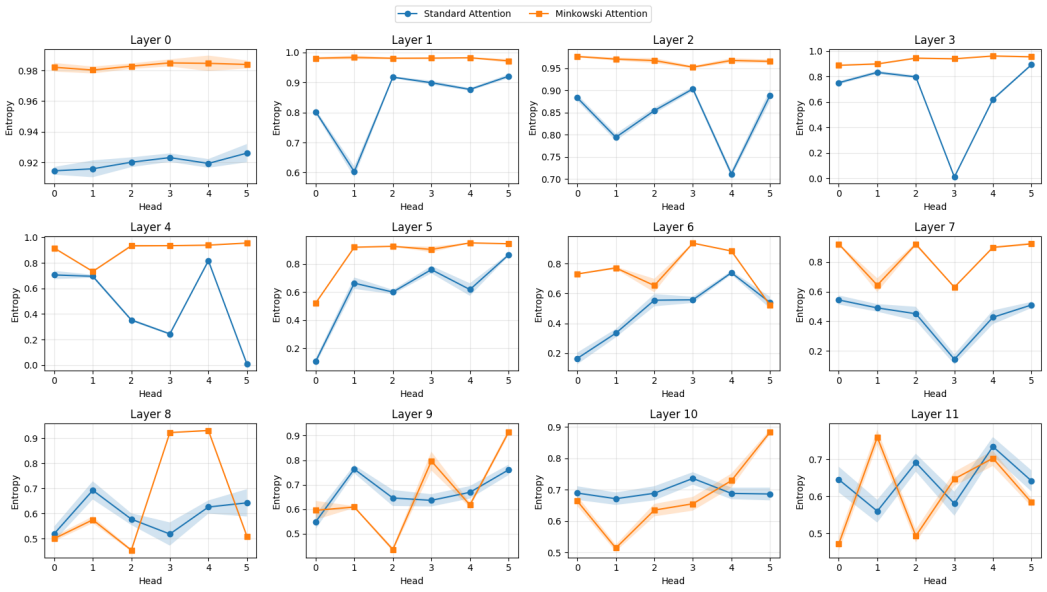

Figure 8: Layer-wise comparison of mean attention entropy across heads for **Standard Attention** and the proposed **Minkowski Attention**. The figure contains 12 subplots corresponding to the 12 Transformer layers. Within each subplot, the entropy of the 6 attention heads (indexed from 0 to 5) is shown, enabling a detailed head-level comparison at each network depth. This visualization reveals how attention sharpness, diversity, and head specialization evolve across layers, and highlights where Minkowski Attention departs from the standard mechanism in allocating attention mass.

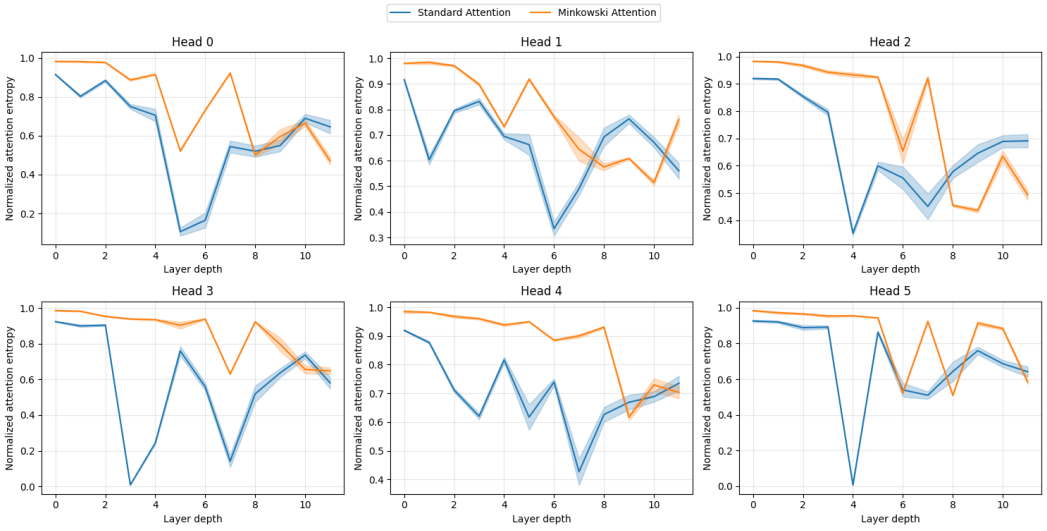

Figure 9: Evolution of normalized attention entropy with network depth for **Standard Attention** and the proposed **Minkowski Attention**. The figure contains 6 subplots, one for each attention head, where entropy is tracked sequentially from layer 0 to layer 11 and averaged over a mini-batch of size 128. Error bars (or shaded regions) denote the corresponding standard deviation across the batch. This view illustrates how attention concentration and dispersion develop through the Transformer hierarchy for each head, highlighting differences in stability, specialization, and depth-wise behavior between the two attention mechanisms.

Table 6: Hierarchical classification results on CIFAR-100 and Tiny-ImageNet. Following Lindeman et al. (2026), all metrics are averaged over every test prediction: tree-induced error (TIE) and lowest-common-ancestor height (LCA) measure how far a prediction lies from the true class in the label hierarchy (lower is better; zero for correct predictions), while the Jaccard similarity (J) and hierarchical precision and recall (P_H, R_H) measure the overlap of their ancestor sets (higher is better). All models are evaluated on the same label hierarchies as LoViT: the official 20-superclass taxonomy for CIFAR-100 and the WordNet hierarchy for Tiny-ImageNet. $^+$ Results reported by Lindeman et al. (2026) for base-size models (∼86M parameters) trained under their pipeline. All other models are trained under our setup with identical architecture size and training settings, and evaluated on the CIFAR-100 test set and the Tiny-ImageNet validation set. Best results in **bold**, second best underlined.

Model	CIFAR-100					Tiny-ImageNet					Params
	TIE↓	LCA↓	J↑	P_H↑	R_H↑	TIE↓	LCA↓	J↑	P_H↑	R_H↑	
Euclidean$^+$	1.084	0.542	0.773	0.819	0.819	4.594	4.034	0.722	0.790	0.791	∼86M
LViT$^+$	1.022	0.511	0.785	0.830	0.830	4.670	4.078	0.717	0.787	0.789	∼86M
HexFormer$^+$	1.076	0.538	0.775	0.821	0.821	4.708	4.132	0.715	0.786	0.786	∼86M
LoViT$^+$	0.871	0.436	0.816	0.855	0.855	4.155	3.650	0.746	0.810	0.811	∼86M
Vanilla ViT	1.168	0.584	0.756	0.805	0.805	4.774	4.221	0.710	0.783	0.781	**12.6M**
MViT (ours)	**0.839**	**0.419**	**0.823**	**0.860**	**0.860**	**3.999**	**3.520**	**0.757**	**0.818**	**0.818**	**12.6M**

vector. We regard this as a plausible explanation rather than an established mechanism: isolating the contribution of the indefinite metric would require ablations beyond the scope of this analysis.

A.7 Pre-LN vs Post-LN Architectures: Stability, Convergence, and Performance

In this section, we compare *Pre-LayerNorm (Pre-LN)* and *Post-LayerNorm (Post-LN)* Transformer architectures for both Standard Attention and Minkowski Attention. Our objective is to examine how normalization placement influences optimization stability, loss convergence, and downstream task performance. This analysis is particularly relevant because Pre-LN architectures are often adopted to improve training stability, especially under larger learning rates or deeper networks, whereas Post-LN designs are commonly associated with stronger final task performance when optimization remains stable. Figure 10 presents the comparative loss trajectories for all settings considered in this study.

The results reveal a clear distinction for Standard Attention. Under identical learning-rate schedules, the Pre-LN variant consistently exhibits smoother optimization and faster loss reduction than the corresponding Post-LN model. In contrast, the Post-LN baseline is more sensitive to optimization hyperparameters and displays less stable convergence, particularly as the learning rate increases. However, this additional optimization stability of Pre-LN is accompanied by weaker downstream performance, suggesting a trade-off between trainability and representation quality.

For Minkowski Attention, the behavior is notably different. Both Pre-LN and Post-LN variants achieve similarly stable optimization and closely matched convergence curves, indicating that the proposed geometry-aware attention mechanism is intrinsically easier to optimize. Unlike the standard baseline, the Post-LN Minkowski model does not suffer from the same instability typically associated with Post-LN training.

These findings suggest that Minkowski Attention inherits many of the optimization benefits commonly attributed to Pre-LN architectures, while retaining the stronger representational capacity of the Post-LN design. In practice, this means that Minkowski Attention can achieve stable training, rapid convergence, and improved downstream accuracy simultaneously, whereas Standard Attention often requires an explicit compromise between stability and final performance.

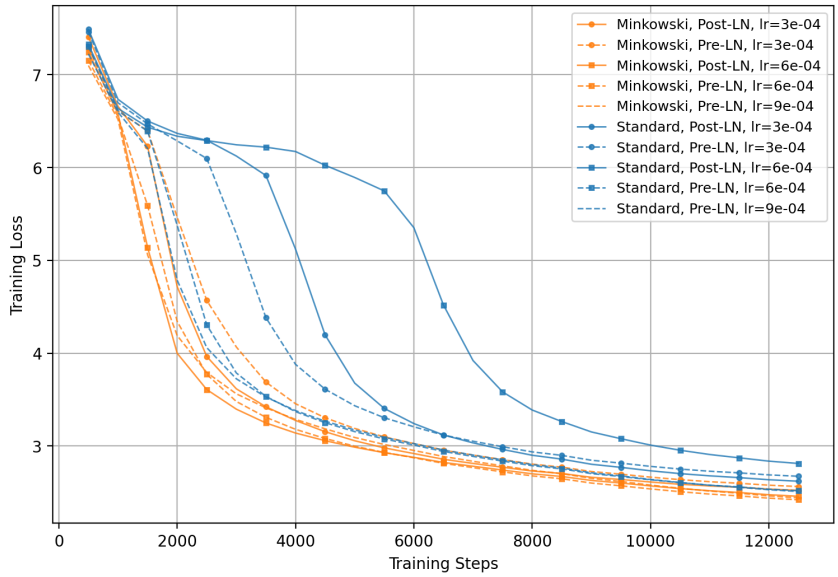

Figure 10: Comparison of training loss convergence for **Standard Attention** and **Minkowski Attention** under both **Pre-LayerNorm (Pre-LN)** and **Post-LayerNorm (Post-LN)** Transformer architectures. Experiments are conducted with learning rates of 3×10^{-4}, 6×10^{-4}, and 9×10^{-4} to examine optimization behavior across multiple stability regimes. The figure provides a comprehensive view of how normalization placement and learning-rate magnitude jointly influence convergence speed, robustness, and final training performance. Curves that continue to decrease smoothly indicate stable optimization, whereas divergence or stagnation highlights sensitivity to architectural choice and aggressive learning-rate.

Table 7: Comparison of Minkowski Attention using learned positional encoding (LPE) and fixed positional encoding (FPE) on SQuAD v1.1, MNLI, and SST-2. Results are reported as mean $\pm$ standard deviation across five random seeds, with the best result for each configuration in **bold**. Parameter counts are given in millions.

Method	SQuAD v1.1		MNLI		SST-2	Params
	EM	F1	MNLI-m	MNLI-mm	Acc	
Hidden=256, Heads=4, Layers=6						
Minkowski Attention LPE	60.62 ± 0.27	72.02 ± 0.31	$\mathbf{71.01 \pm 0.28}$	$\mathbf{72.53 \pm 0.17}$	$\mathbf{84.17 \pm 0.45}$	20.5M
Minkowski Attention FPE	$\mathbf{61.50 \pm 0.27}$	$\mathbf{72.65 \pm 0.17}$	69.21 ± 0.55	71.30 ± 0.68	82.68 ± 0.56	**20.4M**
Hidden=384, Heads=6, Layers=12						
Minkowski Attention LPE	71.74 ± 0.24	81.12 ± 0.15	$\mathbf{76.64 \pm 0.06}$	$\mathbf{77.36 \pm 0.34}$	85.78 ± 0.21	$\sim$45M
Minkowski Attention FPE	$\mathbf{73.21 \pm 0.29}$	$\mathbf{82.23 \pm 0.29}$	76.38 ± 0.04	77.46 ± 0.25	$\mathbf{86.77 \pm 0.56}$	$\sim$45M

A.8 Comparison of Learned and Fixed Positional Encodings

Unless otherwise stated, all results in the paper use learned positional encodings (LPE). To test whether our findings depend on this choice, we also train Minkowski Attention with fixed positional encodings (FPE) under identical model configurations. Table 7 reports results on SQuAD v1.1, MNLI, and SST-2 for two model sizes, averaged over five random seeds.

A.9 Geometry-Aware Attention and Learning-Rate Robustness

In the previous section, we showed that Standard Attention in the Post-LN setting is substantially more sensitive to optimization instability than its Pre-LN counterpart. In this section, we further investigate that phenomenon by progressively increasing the learning rate from 3×10^{-4} to 6×10^{-4} and 9×10^{-4} until training becomes unstable. This provides a direct measure of how the underlying attention mechanism influences sensitivity to optimization hyperparameters, particularly step

size. Figure 11 presents the corresponding loss trajectories for Standard Attention and Minkowski Attention under these learning-rate schedules.

The results show a clear difference in robustness. For Standard Attention, training remains stable at 3×10^{-4}, but increasing the learning rate to 6×10^{-4} leads to unstable optimization, where the loss stops improving after approximately 2,500 steps and fails to recover. This indicates that the optimization landscape of standard Post-LN attention becomes difficult to train once the update magnitude is moderately increased.

In contrast, Minkowski Attention remains stable at 6×10^{-4} and converges to an even lower training loss than the same model trained at 3×10^{-4}. This suggests that the proposed geometry-aware formulation can effectively exploit larger learning rates to accelerate optimization rather than destabilize it. Only at 9×10^{-4} does Minkowski Attention begin to exhibit instability, as illustrated in Figure 11.

Overall, these findings indicate that the geometric structure introduced by Minkowski Attention substantially improves learning-rate robustness in Post-LN Transformers. The proposed method tolerates larger optimization steps, reaches stronger minima under faster training schedules, and preserves the representational advantages of the Post-LN architecture. This combination of stability and efficiency is especially valuable in large-scale pre-training, where optimization speed and hyperparameter robustness are critical.

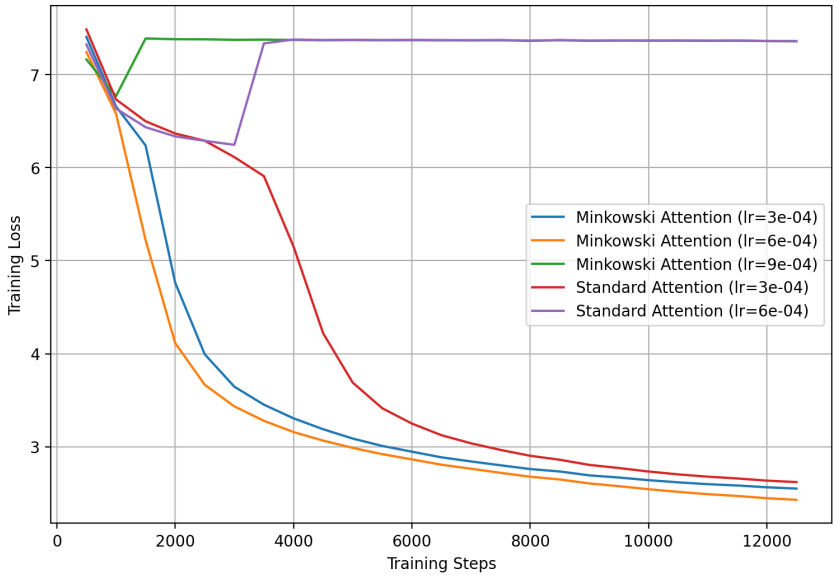

Figure 11: Comparison of training loss convergence for **Standard Attention** and the proposed **Minkowski Attention** under different learning-rate settings, designed to evaluate optimization stability at higher step sizes. The figure illustrates how both models behave during the initial warm-up phase and the subsequent training stage after warm-up. Curves that become unstable or fail to decrease after the warm-up period indicate sensitivity to aggressive learning rates and poor optimization robustness. In contrast, stable loss reduction after warm-up reflects stronger numerical stability and more reliable convergence under challenging training conditions.