PVFormer: Proper Velocity Transformer for Stable and Scalable Hyperbolic Representation Learning

Xianglong Shi[1] **Nicu Sebe**[2] **Bernhard Schölkopf**[3] **Ziheng Chen**[3,2*]
[1]University of Science and Technology of China
[2]University of Trento
[3]Max Planck Institute for Intelligent Systems, Tübingen
xlshi@mail.ustc.edu.cn, ziheng_ch@163.com

Abstract

Hyperbolic representations have shown remarkable success on hierarchical and tree-like data, and Transformers have become a central architecture for representation learning. However, a principled construction of hyperbolic Transformers remains challenging because the self-attention layer decomposes into three coupled primitives, transformation, similarity, and aggregation, that should all respect the underlying geometry. Existing hyperbolic Transformer designs are typically built in the Poincaré or Lorentz models, where bounded domains, time-like constraints, tangent-space detours, or projection steps make it difficult to keep the full block intrinsic and scalable. To address this gap, we propose PVFORMER, to the best of our knowledge, the first Transformer architecture that operates natively in proper velocity (PV) space. The proper velocity model provides an unconstrained representation of hyperbolic geometry, and PVFORMER exploits this structure to redesign the Transformer block in PV space. On the transformation side, we use PV homomorphism transformations to update query and value representations. On the similarity side, we generalize Euclidean inner-product attention with a PV Busemann score inspired by Busemann-based hyperbolic learning and a PV point-to-hyperplane key transformation. On the aggregation side, we replace Euclidean averaging with PV midpoint aggregation and further derive a scalable PV linear attention variant for large graphs. Extensive experiments support the effectiveness of PVFORMER across graph, text, and vision benchmarks.

1 Introduction

Hyperbolic geometry provides a natural representation for hierarchical and tree-like data because negative curvature offers exponential representation capacity [25, 26, 16, 23, 7]. This property has driven successful applications in graphs [21, 42, 41], visual categories [2, 30], and biological sequences [30, 10], and recent work has moved from hyperbolic embeddings toward hyperbolic neural networks that operate within hyperbolic space. However, as emphasized by PVNN [10], the choice of hyperbolic model is central to neural architecture design [30, 10]. Poincaré and Lorentz models provide powerful formulations and many closed-form operations, but bounded domains, explicit manifold constraints, tangent-space detours, and projection steps can introduce numerical and implementation difficulties. In particular, Poincaré embeddings that approach the boundary can make computations unstable and cause gradients to vanish [10].

These issues become especially important for Transformers, whose self-attention layer is a coupled pipeline rather than a single similarity function. Each block must transform hidden states into query, key, and value representations, compare queries and keys, and aggregate values [35]. Existing

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

hyperbolic attention and Transformer methods make important progress toward this goal, but they often align only part of the pipeline with hyperbolic geometry [18, 41]. Euclidean or tangent-space transformations followed by restoration steps preserve feasibility but weaken the link between learned updates and intrinsic geometry [7, 41]; distance-based similarities encode proximity but can discard directionality, while tangent-space inner products depend on a base point [18, 41]; and log/exp-map or projection-based aggregation can add computational and numerical burden [27]. The challenge is therefore not merely to replace the Euclidean dot product, but to realize transformation, similarity, and aggregation in one coherent hyperbolic computational model.

Proper velocity (PV) space suggests such a route. It represents the same hyperbolic geometry as the Poincaré and Lorentz models using unconstrained vectors, while retaining Riemannian operations for hyperbolic learning through the PV metric and gyrovector structure [33, 10]. This model avoids bounded-ball management and explicit hyperboloid restoration, making it attractive for repeatedly composed neural blocks. The PVNN demonstrates that PV space can support stable hyperbolic neural layers, but it does not instantiate the attention pipeline required by Transformers. A native proper velocity Transformer must still define geometry-aware transformations, directional similarity, intrinsic aggregation, and a scalable attention path within the same PV representation.

Motivated by this gap, we propose PVFORMER, to the best of our knowledge, the first Transformer architecture operating natively in proper velocity space. PVFORMER keeps the attention pipeline in PV space by using PV homomorphism transformations for query and value updates, a PV point-to-hyperplane transformation that produces direction-like keys, PV Busemann attention for curvature-aware similarity [17], and PV midpoint aggregation in place of Euclidean weighted sums. For large graphs, we further use a PV linear attention path that avoids materializing a quadratic Busemann score matrix while preserving the midpoint-based aggregation principle. We evaluate PVFORMER across graph, text, and vision benchmarks under a unified protocol, showing that proper velocity space can serve not only as a stable hyperbolic representation model but also as the computational foundation for Transformer attention.

Our contributions are summarized as follows:

- We formulate the three primitives of attention, transformation, similarity, and aggregation, directly in PV space through PV homomorphism transformations, PV point-to-hyperplane keys for Busemann attention, and PV midpoint aggregation.
- We derive PV linear attention, a scalable linear-attention variant that avoids materializing a quadratic Busemann score matrix while preserving PV midpoint aggregation for large graph settings.
- We develop PVFORMER, to the best of our knowledge, the first Transformer architecture operating natively in proper velocity space.
- We show that PVFORMER achieves stronger predictive performance and faster attention computation across graph, text, and vision benchmarks under a unified experimental protocol.

2 Related Work

2.1 Hyperbolic neural networks and geometric models

Hyperbolic representation learning exploits negative curvature for hierarchical or tree-like structure, beginning with Poincaré and Lorentz embeddings and extending to neural and graph architectures through gyrovector and Riemannian operations [6, 25, 26, 16, 23, 7, 5, 4]. Model choice directly affects implementation: Poincaré models are bounded, Lorentz models introduce an explicit time-like dimension, and both can exhibit numerical sensitivity in composed neural layers [24, 44]. Fully Lorentz and intrinsic Lorentz networks keep operations on the hyperboloid [8, 34, 30], whereas PVNN [10] develops closed-form layers in an unconstrained PV model derived from gyrovector structures [33, 10]. PVFORMER builds on this model-level opportunity but targets the Transformer block itself, rather than using PV only as a layer family or representation conversion.

2.2 Hyperbolic attention, Busemann functions, and Transformers

Transformers organize representation learning around transformations, similarity, and aggregation [35]; graph Transformers and efficient graph attention variants further emphasize the scalability

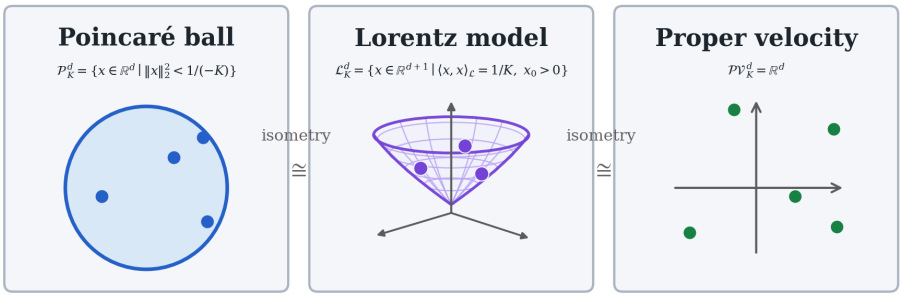


Figure 1: Comparison of three two-dimensional hyperbolic models. PV represents the same hyperbolic geometry as the Poincaré and Lorentz models, but removes equality or inequality constraints.

constraints of this decomposition [43, 28, 32, 38, 39]. Hyperbolic Attention Networks and Hypformer move attention or complete Transformer modules into hyperbolic geometry, with Hypformer providing the main Lorentz-model linear-attention reference for our experiments [18, 41]. In parallel, Busemann-based methods use ideal-boundary potentials for hyperbolic similarity and classification, and hyperbolic vision Transformers illustrate broader non-Euclidean Transformer designs [17, 1]. PVFORMER differs by placing the transformation–similarity–aggregation pipeline in PV space, combining homomorphic updates, PV Busemann attention, PV midpoint aggregation, and PV linear attention in one native PV Transformer.

3 Preliminaries

3.1 Riemannian geometry

We use standard Riemannian notation: $(\mathcal{M}, g)$ denotes a smooth manifold with tangent inner product g_x and norm $\|v\|_x = \sqrt{g_x(v,v)}$; d, Exp_x, Log_x, and $\mathrm{PT}_{x\to y}$ denote geodesic distance, exponential map, logarithmic map, and parallel transport [22]. An isometry is a smooth bijection whose differential preserves the tangent inner products. Hyperbolic space is the simply connected manifold of constant negative curvature [6]. As shown in Figure 1, the Poincaré, Lorentz, and proper velocity models describe this same geometry with different constraints: a bounded ball, an ambient hyperboloid, or an unconstrained PV representation connected to gyrovector constructions [33, 10].

3.2 Proper velocity space

Let $K < 0$ and let $\mathcal{PV}_K^d = \mathbb{R}^d$ be the d-dimensional proper velocity model. Points are stored as unconstrained vectors, but tangent vectors are measured by the PV metric [10]:

$$\beta_u = \frac{1}{\sqrt{1 - K\|u\|^2}}, \qquad g_u(a,b) = \langle a, b\rangle + K\beta_u^2 \langle u, a\rangle\langle u, b\rangle, \tag{1}$$

for $a, b \in T_u\mathcal{PV}_K^d \simeq \mathbb{R}^d$. Because $K < 0$, β_u is finite for every $u \in \mathbb{R}^d$, unlike the bounded Poincaré ball or the Lorentz hyperboloid constraint. PV space also has gyroaddition $\oplus_U$ and scalar gyromultiplication $\otimes_U$ that generalize Euclidean vector operations [33, 10]; PVFORMER uses this structure to keep transformations, similarity, and aggregation PV-compatible. PV is not a different geometry from the Lorentz model but an unconstrained model of the same hyperbolic manifold [10]. The lift $\iota_{\mathrm{PV}\to\mathbb{L}}(x) = (\sqrt{\|x\|^2 - 1/K}, x)$ maps a PV point to the Lorentz hyperboloid, so the PV layers used by PVFORMER can be viewed as Lorentz-model hyperbolic operations represented in PV space, with the time-like constraint absorbed by the metric and closed-form PV operations. This relation lets PVFORMER retain the geometric meaning of Lorentz-based hyperbolic Transformers while avoiding explicit hyperboloid projection or restoration inside the block.

3.3 Busemann functions

For a proper geodesic metric space (X, d) and a unit-speed geodesic ray $\gamma : [0, \infty) \to X$, the Busemann function is

$$B^\gamma(x) = \lim_{t\to\infty} \big(d(x, \gamma(t)) - t\big), \tag{2}$$

where $x \in X$ [3]. In Euclidean space, the ray $\gamma(t) = tv$ with $\|v\| = 1$ yields $B^v(x) = -\langle x, v\rangle$. Thus, Busemann potentials recover directional inner products in the flat case and provide the similarity primitive used below for curvature-aware attention.

4 Method

In this section, we present PVFORMER. Following the three-part view of attention, we redesign the Transformer block directly in proper velocity space. First, we construct *query* (Q) and *value* (V) updates as PV gyro-homomorphic transformations induced by origin-preserving orthogonal maps. These updates are Riemannian isometries of the PV metric and therefore play the role of geometry-preserving analogues of Euclidean linear projections. Second, we reformulate dot-product similarity through the Busemann function and obtain a curvature-aware PV Busemann attention score. Third, because Busemann scoring requires direction-like keys, we introduce a PV point-to-hyperplane key transformation that maps hidden states to the origin tangent space. Finally, for large graphs, we use PV linear attention, which replaces quadratic Busemann scoring with kernelized linear weights while preserving PV midpoint aggregation. The resulting pipeline keeps transformation, similarity, and aggregation in the same PV computational model instead of alternating between Euclidean updates and post-hoc geometric restoration.

4.1 PV homomorphism transformation

In Euclidean attention, the Q and V updates are linear homomorphisms: they preserve vector addition and scalar multiplication. We generalize this role to proper velocity space by requiring the corresponding updates to preserve PV gyroaddition and scalar gyromultiplication. Let $K < 0$ and let $\mathcal{PV}_K^d = \mathbb{R}^d$ be equipped with the PV metric in Equation (1). For $x \in \mathcal{PV}_K^d$, write $\beta_x = \frac{1}{\sqrt{1-K\|x\|^2}}$. The PV gyroaddition and scalar gyromultiplication can be written as [10, 33]

$$x \oplus_U y = x + y + \left(\frac{1-\beta_y}{\beta_y} - K\frac{\beta_x}{1+\beta_x}\langle x, y\rangle\right) x, \tag{3}$$

$$t \otimes_U x = \frac{\sinh\big(t \operatorname{arsinh}(\sqrt{-K}\|x\|)\big)}{\sqrt{-K}\|x\|} x, \qquad t \otimes_U 0 = 0. \tag{4}$$

Theorem 1 (PV isotropy homomorphism) *Let $R \in SO(d)$ and define*

$$\Phi_R^{\mathrm{PV}} : \mathcal{PV}_K^d \to \mathcal{PV}_K^d, \qquad \Phi_R^{\mathrm{PV}}(x) = Rx. \tag{5}$$

Then Φ_R^{PV} is both a gyrovector homomorphism and a Riemannian isometry:

$$\Phi_R^{\mathrm{PV}}(x \oplus_U y) = \Phi_R^{\mathrm{PV}}(x) \oplus_U \Phi_R^{\mathrm{PV}}(y), \tag{6}$$

$$\Phi_R^{\mathrm{PV}}(t \otimes_U x) = t \otimes_U \Phi_R^{\mathrm{PV}}(x), \tag{7}$$

$$g_{Rx}(Ru, Rv) = g_x(u, v). \tag{8}$$

Although $\mathcal{PV}_K^d$ is stored as an unconstrained array, Theorem 1 shows that R is not being used as an arbitrary Euclidean layer: it preserves the PV algebraic operations and the PV Riemannian metric. We provide the proof in Section D.1 and use the theorem here as a design principle for constraining PVFORMER query and value updates to geometry-preserving rotations rather than arbitrary Euclidean projections.

This construction specializes the PV–Lorentz relation above to homomorphism transformations. Conjugating the Lorentz isotropy map $\mathrm{diag}(1, R)$ through the PV–hyperboloid isometry gives exactly $x \mapsto Rx$ in the PV model. Therefore, the origin-preserving Lorentz homomorphism used for hyperbolic Q/V updates becomes an ordinary orthogonal rotation in PV space, while retaining the same geometric meaning.

No bias term is added: a biased map would not preserve the gyro identity 0, and an unconstrained linear map would not generally preserve the PV metric. The key representation is handled separately by the PV point-to-hyperplane key transformation in Section 4.3, because Busemann similarity requires direction-like keys rather than another point-valued homomorphic update. Our construction uses equal-dimensional rotations in $SO(d)$. Its expressiveness and possible dimension-changing extensions are discussed in Section H.

4.2 PV Busemann attention

PVFORMER defines similarity by rewriting Euclidean dot-product attention as a Busemann potential. For nonzero k and $u_k = k/\|k\|$, the identity $\langle q, k\rangle = -\|k\| B^{u_k}(q)$ expresses the dot product as a scaled directional Busemann function. Pulling the Lorentz Busemann function back through the PV–hyperboloid isometry gives the following score entirely in PV space.

Theorem 2 (PV Busemann function) *Let $K < 0$ and $q \in \mathcal{PV}_K^d$. For a unit direction $u \in T_0\mathcal{PV}_K^d \simeq \mathbb{R}^d$ with $\|u\| = 1$, let $\gamma_u(t) = \mathrm{Exp}_0(tu)$ be the unit-speed geodesic ray from the origin. The* PV *Busemann function associated with this ray is*

$$B^u_{\mathrm{PV},K}(q) = \lim_{t\to\infty} \big(d_{\mathrm{PV}}(q, \gamma_u(t)) - t\big) = \frac{1}{\sqrt{-K}} \log\left(\sqrt{1 - K\|q\|^2} - \sqrt{-K}\langle q, u\rangle\right). \tag{9}$$

For any nonzero tangent key $k \in T_0\mathcal{PV}_K^d \setminus \{0\}$, the corresponding PV *Busemann similarity is*

$$s_{\mathrm{PV}}(q, k) = -\|k\| B^{k/\|k\|}_{\mathrm{PV},K}(q) = -\frac{\|k\|}{\sqrt{-K}} \log\left(\sqrt{1 - K\|q\|^2} - \sqrt{-K}\left\langle q, \frac{k}{\|k\|}\right\rangle\right). \tag{10}$$

Moreover, $s_{\mathrm{PV}}(q, k) \to \langle q, k\rangle$ as $K \to 0^-$.

The derivation, positivity of the logarithm, and flat-limit proof are given in Section D.2.

PV Busemann logits. For each head, the query is $q_i = \Phi^{\mathrm{PV}}_{R_Q}(u_i)$ and the point-to-hyperplane key module in Section 4.3 returns k_j. We use $\hat{k}_j = \frac{k_j}{\|k_j\|+\varepsilon}$ and $\rho_i = \sqrt{1 - K\|q_i\|^2}$, and compute the implemented logit as

$$S^{\mathrm{PV}}_{ij} = -\frac{\tau_h}{\sqrt{-K}} \log\left(\rho_i - \sqrt{-K}\langle q_i, \hat{k}_j\rangle + \varepsilon\right), \tag{11}$$

where $\tau_h > 0$ is learnable; replacing it by the key norm recovers the strict dot-product generalization in Theorem 2.

Theorem 3 (PV midpoint aggregation) *Let $v_1, \ldots, v_N \in \mathcal{PV}_K^d$ and let $\alpha_j \geq 0$ with $\sum_{j=1}^N \alpha_j = 1$. Define $b = \sum_{j=1}^N \alpha_j v_j, r = \sum_{j=1}^N \alpha_j \sqrt{1 - K\|v_j\|^2}$ and the corresponding* PV *midpoint aggregation is*

$$\mathrm{Mid}^{\mathrm{PV}}_K\big(\{(v_j, \alpha_j)\}_{j=1}^N\big) = \frac{b}{\sqrt{r^2 + K\|b\|^2}}. \tag{12}$$

Theorem 3 is the PV form of the Lorentzian midpoint: r corresponds to the weighted time-like component and b to the spatial component, so aggregation stays in PV geometry rather than reverting to a Euclidean weighted sum.

Attention weights and aggregation. The full attention variant applies a row-wise softmax to the Busemann logits and midpoint-aggregates values:

$$\alpha_{ij} = \frac{\exp(S^{\mathrm{PV}}_{ij}/\sigma + b_{ij})}{\sum_\ell \exp(S^{\mathrm{PV}}_{i\ell}/\sigma + b_{i\ell})}, \qquad z_i = \mathrm{Mid}^{\mathrm{PV}}_K\big(\{(v_j, \alpha_{ij})\}_{j=1}^N\big), \tag{13}$$

where $\sigma > 0$ is the attention scale and b_{ij} denotes an optional mask, edge bias, or positional bias. The medium-scale experiments use this full PV Busemann attention path.

4.3 Point-to-hyperplane transformation for keys

The Busemann similarity in Section 4.2 requires queries and values to remain point-valued PV representations, while keys must be represented as directions in the origin tangent space $T_0\mathcal{PV}_K^m$. Mapping every key by a logarithmic map at the origin would reintroduce a tangent-chart detour. We therefore construct a direct PV point-to-hyperplane transformation: each output component is a signed distance-like margin to a learned PV hyperplane, and stacking these margins yields the tangent key.

The Euclidean fully connected layer is defined as $y = Wx + b$ with $W \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Shimizu et al. [31, Sec. 3.2] and Chen et al. [11, Sec. 3.1] interpret Euclidean FC layers through

signed distances: we write each output component as $v_\ell(x) = \langle w_\ell, x\rangle - b_\ell$ and view it as the signed distance from y to the axis-aligned hyperplane

$$H_{e^{(\ell)},0} = \{y \in \mathbb{R}^m : \langle e^{(\ell)}, y - 0\rangle = 0\}, \qquad \operatorname{sign}\big(\langle e^{(\ell)}, y - 0\rangle\big) d\big(y, H_{e^{(\ell)},0}\big) = v_\ell(x), \tag{14}$$

where $e^{(\ell)}$ is the ℓth standard basis vector. The left-hand side of Equation (14) is exactly the ℓth output component.

Keeping the left-hand side and replacing the Euclidean affine score by a PV signed hyperplane margin extends this formulation to a PV-to-tangent key layer, analogous to Lorentz point-to-hyperplane layers based on hyperbolic MLR logits [2, Eq. 12]. The PV counterpart uses the signed-margin construction of PVNN [10], reviewed in Section C, and gives the following key map.

Theorem 4 (PV point-to-hyperplane key map) *Let $K < 0$ and let $x \in \mathcal{PV}_K^n$. For each output component $\ell = 1, \dots, m$, take unconstrained parameters $z_\ell \in \mathbb{R}^n \setminus \{0\}$ and $r_\ell \in \mathbb{R}$, and set*

$$p_\ell = \operatorname{Exp}_0\left(r_\ell \frac{z_\ell}{\|z_\ell\|}\right), \qquad a_\ell = \operatorname{PT}_{0\to p_\ell}(z_\ell). \tag{15}$$

Define $\Psi_\theta^{\mathrm{PV}} : \mathcal{PV}_K^n \to T_0\mathcal{PV}_K^m \simeq \mathbb{R}^m$ by $\Psi_\theta^{\mathrm{PV}}(x) = \big[\psi_1(x), \dots, \psi_m(x)\big]^\top$, where

$$\psi_\ell(x) = \frac{\|z_\ell\|}{\sqrt{-K}} \operatorname{arsinh}\left(\cosh(\sqrt{-K} r_\ell) \frac{\sqrt{-K}}{\|z_\ell\|} \langle x, z_\ell\rangle - \sinh(\sqrt{-K} r_\ell) \sqrt{1 - K\|x\|^2}\right). \tag{16}$$

Then $\Psi_\theta^{\mathrm{PV}}(x)$ is the vector of signed PV *point-to-hyperplane margins associated with the learned hyperplanes $\mathcal{H}^{\mathrm{PV}}_{a_\ell, p_\ell}$, represented as an origin tangent-space key. Moreover, as $K \to 0^-$,*

$$\Psi_\theta^{\mathrm{PV}}(x) \to Zx + b, \qquad b_\ell = -r_\ell \|z_\ell\|, \tag{17}$$

where the rows of $Z \in \mathbb{R}^{m\times n}$ are $z_\ell^\top$.

The proof is given in Section D.3. The output is not a point in $\mathcal{PV}_K^m$: it is a tangent vector in $T_0\mathcal{PV}_K^m$ whose components are signed hyperplane margins. This distinction is necessary because the Busemann score in Theorem 2 uses keys as directions. For a token or node representation u_i, the key used by PVFORMER is therefore $k_i = \Psi_\theta^{\mathrm{PV}}(u_i) \in T_0\mathcal{PV}_K^m$, $S_{ij}^{\mathrm{PV}} = s_{\mathrm{PV}}(q_i, k_j)$.

4.4 PV linear attention

Full Busemann attention forms all pairwise query–key scores and therefore has quadratic cost in the number of tokens or nodes. This is not the appropriate path for large graphs such as ogbn-proteins, Amazon2M, and ogbn-arxiv. Following Hypformer's linear-attention logic, we avoid explicitly materializing the $N \times N$ attention matrix by changing the computation order from $(QK^\top)V$ to $Q(K^\top V)$. The difference is that PVFORMER keeps the value aggregation as a PV midpoint rather than a Euclidean weighted sum.

In the linear branch, keys are no longer required to be Busemann directions in $T_0\mathcal{PV}_K^d$; instead, queries, keys, and values are point-valued PV representations obtained by the homomorphism transformations in Section 4.1. For one head, let

$$q_i = \Phi^{\mathrm{PV}}_{R_Q}(u_i), \qquad k_i = \Phi^{\mathrm{PV}}_{R_K}(u_i), \qquad v_i = \Phi^{\mathrm{PV}}_{R_V}(u_i), \qquad R_Q, R_K, R_V \in SO(d_h). \tag{18}$$

Focused nonnegative feature map. As in Hypformer, we use a positive-focused feature map to sharpen linear attention. For an input row $e \in \mathbb{R}^{d_h}$, define

$$\widetilde{e} = \frac{\operatorname{ReLU}(e) + \varepsilon}{t}, \qquad \varphi_p(e) = \frac{\|\widetilde{e}\|_2}{\|\widetilde{e}^{\odot p}\|_2 + \varepsilon} \widetilde{e}^{\odot p}, \qquad p > 1, \tag{19}$$

where $t > 0$ is a learnable scale. This gives nonnegative kernel weights

$$\lambda_{ij} = \varphi_p(q_i)^\top \varphi_p(k_j), \qquad \alpha_{ij} = \frac{\lambda_{ij}}{\sum_{\ell=1}^N \lambda_{i\ell}}. \tag{20}$$

A direct midpoint implementation would insert these α_{ij} into Theorem 3, but doing so would still require all N^2 weights. The key observation is that the midpoint can be algebraically linearized.

Table 1: Testing results on large-scale node property prediction benchmarks and small and medium-sized graph benchmarks. F1-score is used for DISEASE, ROC-AUC for ogbn-proteins, and Accuracy for all other datasets. OOM denotes out of memory during training or testing. The PVFORMER row is highlighted in gray, with best results shown in red bold.

	Small/Medium					Large		
Method	DISEASE	AIRPORT	CORA	CITESEER	PUBMED	ogbn-proteins	Amazon2M	ogbn-arxiv
#Nodes	1,044	2,665	2,708	3,327	19,717	132,534	2,449,029	169,343
#Edges	1,043	2,664	5,429	4,732	88,651	39,561,252	61,859,140	1,166,243
MLP	–	–	–	–	–	72.0 ± 0.5	63.5 ± 0.1	55.5 ± 0.2
GCN [21]	69.7 ± 0.4	81.4 ± 0.6	81.3 ± 0.3	71.6 ± 0.4	78.1 ± 0.2	72.5 ± 0.4	83.9 ± 0.1	71.7 ± 0.3
GAT [36]	70.4 ± 0.4	81.5 ± 0.3	83.0 ± 0.7	72.5 ± 1.1	79.0 ± 0.3	–	–	–
SGC [37]	69.1 ± 0.6	82.1 ± 0.5	80.1 ± 0.2	71.9 ± 0.1	78.7 ± 0.1	70.3 ± 0.2	81.2 ± 0.1	67.8 ± 0.3
GCN-NSampler	–	–	–	–	–	73.5 ± 1.3	83.8 ± 0.4	68.5 ± 0.2
GAT-NSampler	–	–	–	–	–	74.6 ± 1.2	85.2 ± 0.3	67.6 ± 0.2
SIGN [15]	–	–	–	–	–	71.2 ± 0.5	81.0 ± 0.3	70.3 ± 0.3
HGNN [23]	81.3 ± 3.5	84.7 ± 1.0	77.1 ± 0.8	70.0 ± 1.0	78.3 ± 1.2	–	–	–
HGCN [7]	88.2 ± 0.7	89.3 ± 1.2	76.5 ± 0.6	68.0 ± 0.6	78.0 ± 1.0	–	–	–
HGAT [7]	90.3 ± 0.6	89.6 ± 1.0	77.4 ± 0.7	68.6 ± 0.3	78.3 ± 1.4	–	–	–
GraphFormer [43]	75.2 ± 0.0	88.1 ± 1.2	60.0 ± 0.5	61.4 ± 0.6	73.3 ± 0.7	OOM	OOM	OOM
GraphTrans [40]	89.3 ± 3.2	94.3 ± 0.6	77.6 ± 0.8	65.1 ± 1.4	77.5 ± 0.7	OOM	OOM	OOM
GraphGPS [28]	92.8 ± 2.7	94.5 ± 0.9	73.0 ± 1.4	62.0 ± 1.5	72.8 ± 1.4	OOM	OOM	OOM
FPS-T [13]	88.6 ± 0.9	96.0 ± 0.6	82.3 ± 0.7	70.0 ± 0.7	78.5 ± 0.6	–	–	–
HAN [18]	85.1 ± 0.8	92.9 ± 0.6	83.1 ± 0.5	72.4 ± 0.5	79.0 ± 0.6	OOM	OOM	OOM
HNN++ [31]	89.5 ± 0.2	92.3 ± 0.3	82.8 ± 0.6	71.5 ± 1.3	79.9 ± 0.4	OOM	OOM	OOM
F-HNN [8]	92.3 ± 1.1	93.0 ± 0.7	81.0 ± 0.7	71.2 ± 0.4	77.5 ± 0.8	OOM	OOM	OOM
NodeFormer [38]	75.9 ± 0.9	80.2 ± 0.6	82.2 ± 0.9	72.5 ± 1.1	79.9 ± 1.0	77.5 ± 1.2	87.9 ± 0.2	59.9 ± 0.4
SGFormer [39]	89.0 ± 3.9	92.9 ± 0.5	83.2 ± 0.9	72.2 ± 0.3	80.0 ± 0.8	79.5 ± 0.3	89.1 ± 0.1	72.4 ± 0.3
Hypformer	93.0 ± 0.7	95.0 ± 0.5	85.0 ± 0.3	73.3 ± 0.4	81.3 ± 0.3	80.4 ± 0.5	89.4 ± 0.3	73.2 ± 0.2
PVFORMER	$\mathbf{94.07 \pm 0.44}$	$\mathbf{96.37 \pm 0.27}$	$\mathbf{86.20 \pm 0.18}$	$\mathbf{73.70 \pm 0.20}$	$\mathbf{83.00 \pm 0.24}$	$\mathbf{81.51 \pm 0.30}$	$\mathbf{90.03 \pm 0.19}$	$\mathbf{73.86 \pm 0.21}$

Theorem 5 (PV linear attention) *Let $K < 0$, let $\lambda_{ij} = \varphi_p(q_i)^\top \varphi_p(k_j) \geq 0$, and define $\rho_j = \sqrt{1 - K\|v_j\|^2}$, $B_i = \sum_{j=1}^N \lambda_{ij} v_j$, $A_i = \sum_{j=1}^N \lambda_{ij} \rho_j$. Then applying the* PV *midpoint in Theorem 3 with normalized weights $\alpha_{ij} = \lambda_{ij} / \sum_\ell \lambda_{i\ell}$ is equivalent to*

$$o_i^{\mathrm{PVLin}} = \frac{B_i}{\sqrt{A_i^2 + K\|B_i\|^2}}. \tag{21}$$

Moreover, as $K \to 0^-$, the expression recovers Euclidean linear attention,

$$o_i^{\mathrm{PVLin}} \longrightarrow \frac{\sum_{j=1}^N \lambda_{ij} v_j}{\sum_{j=1}^N \lambda_{ij}}. \tag{22}$$

The proof is given in Section D.5.

5 Experiments

5.1 Experimental setup

In this section, we evaluate PVFORMER on graph benchmarks and semi-supervised image and text classification settings. Graphs provide a natural testbed because their connectivity often contains hierarchical and long-range structure, so we begin with small and medium graph benchmarks and then move to large-scale node property prediction. We further evaluate Mini-ImageNet and 20News-Groups, where input graphs are constructed by a k-nearest-neighbor procedure or removed entirely, to test whether the gains transfer beyond datasets with a native graph.

To ensure a controlled comparison, we follow the Hypformer protocol and table layout for preprocessing, data splits, metrics, and baselines [20, 12, 38, 41]. The reported PVFORMER rows are our results under the same protocol. Disease uses F1, ogbn-proteins uses ROC-AUC, and all other listed datasets use accuracy. The main architectural difference is the attention core: PVFORMER replaces Euclidean or Lorentz attention with PV homomorphism transformations, PV Busemann scoring, PV midpoint aggregation, and the PV linear attention variant used for large graphs.

Table 2: Experimental results on semi-supervised classification on Mini-ImageNet and 20News-Groups, where we use k-NN graphs with different values of k. The PVFORMER rows are highlighted in gray, with our results shown in red bold.

Method	Mini-ImageNet				20News-Groups			
	$k=5$	$k=10$	$k=15$	$k=20$	$k=5$	$k=10$	$k=15$	$k=20$
GCN [21]	84.86 ± 0.42	85.61 ± 0.40	85.93 ± 0.59	85.96 ± 0.66	65.98 ± 0.68	64.13 ± 0.88	62.95 ± 0.70	62.59 ± 0.62
GAT [36]	84.70 ± 0.48	85.24 ± 0.42	85.41 ± 0.43	85.37 ± 0.51	64.06 ± 0.44	62.51 ± 0.71	61.38 ± 0.88	60.80 ± 0.59
DropEdge [29]	83.91 ± 0.24	85.35 ± 0.44	85.25 ± 0.63	85.81 ± 0.65	64.46 ± 0.43	64.01 ± 0.42	62.46 ± 0.51	62.68 ± 0.71
IDGL [9]	83.63 ± 0.32	84.41 ± 0.35	85.50 ± 0.24	85.66 ± 0.42	65.09 ± 1.23	63.41 ± 1.26	61.57 ± 0.52	62.21 ± 0.79
LDS [14]	OOM	OOM	OOM	OOM	66.15 ± 0.36	64.70 ± 1.07	63.51 ± 0.64	63.51 ± 1.75
NodeFormer [38]	86.77 ± 0.45	86.74 ± 0.23	86.87 ± 0.41	86.64 ± 0.42	66.01 ± 1.18	65.21 ± 1.14	64.69 ± 1.31	64.55 ± 0.97
SGFormer [39]	86.21 ± 0.66	86.46 ± 0.61	86.73 ± 0.84	86.76 ± 0.72	68.55 ± 0.54	67.96 ± 0.68	66.44 ± 0.87	65.46 ± 0.59
NodeFormer w/o graph [39]	87.46 ± 0.36				64.71 ± 1.33			
SGFormer w/o graph [39]	87.25 ± 0.38				67.53 ± 0.43			
Hypformer [41]	87.36 ± 0.73	87.30 ± 0.65	87.41 ± 0.59	87.48 ± 0.61	68.21 ± 0.78	68.01 ± 0.34	66.87 ± 0.30	66.74 ± 0.19
Hypformer w/o graph [41]	87.73 ± 0.63				67.73 ± 0.23			
PVFORMER	$\mathbf{88.31 \pm 0.21}$	$\mathbf{88.38 \pm 0.32}$	$\mathbf{88.51 \pm 0.29}$	$\mathbf{88.76 \pm 0.30}$	$\mathbf{71.99 \pm 0.51}$	$\mathbf{70.62 \pm 0.30}$	$\mathbf{70.33 \pm 0.24}$	$\mathbf{70.16 \pm 0.20}$
PVFORMER w/o graph	$\mathbf{88.47 \pm 0.30}$				$\mathbf{70.29 \pm 0.15}$			

5.2 Main results

Graph results. Table 1 combines the small/medium graph benchmarks and large-scale node property prediction benchmarks in one grouped table. On small and medium graphs, PVFORMER improves over the Hypformer reference means on all five datasets, with the largest gains on Airport, Cora, and PubMed, and smaller but positive margins on Disease and Citeseer. On the large benchmarks, PVFORMER improves over Hypformer on ogbn-proteins, Amazon2M, and ogbn-arxiv while retaining the scalable PV linear attention path. These results indicate that the PV attention pipeline improves accuracy across graph scales while preserving the efficiency required by large graphs.

Text and vision results. Table 2 evaluates semi-supervised classification on Mini-ImageNet and 20News-Groups. Following graph Transformer protocols, we construct k-nearest-neighbor graphs for $k \in \{5, 10, 15, 20\}$ and also report graph-free variants. This setup separates gains from the learned PV attention block from gains that depend on a particular neighborhood construction. On 20News-Groups, PVFORMER substantially improves over the Hypformer reference across every k and in the graph-free setting. The best gain appears at $k = 5$, where accuracy rises from 68.21 to 71.99, and all four graph settings exceed the baseline by 1.5 points. The graph-free 20News-Groups result of 70.29 shows that the improvement is not tied to a single constructed graph. On Mini-ImageNet, PVFORMER is consistently above the Hypformer paper baseline, with the best graph result of 88.76 at $k = 20$ and 88.47 without an explicit graph.

5.3 Ablations

Curvature sensitivity. Table 3 evaluates whether the PVFORMER results depend on a finely tuned curvature value. We keep the input PV scale fixed and sweep the hidden/output curvature over different K on Airport, Cora, Citeseer, and PubMed. We also include the Euclidean counterpart ($K = 0$) under the same architecture. All tested negative-curvature settings outperform this baseline on all four datasets. Among the negative-curvature settings, the Highest Test span is at most 0.44 points, and it is below 0.25 points on Cora, Citeseer, and PubMed. This indicates that PVFORMER is not sensitive to the exact negative curvature within this range; choosing a reasonable PV curvature is sufficient once the surrounding optimization hyperparameters are fixed.

Table 3: Curvature sensitivity of PVFORMER on small and medium graph benchmarks, including the Euclidean baseline ($K = 0$). Span denotes the difference between the best and worst mean Highest Test scores for $K < 0$.

Dataset	$K=0$ (Euclidean)	$K=-0.5$	$K=-1.0$	$K=-2.0$	$K=-3.0$	$K=-4.0$	Span
AIRPORT	95.02 ± 0.11	96.06 ± 0.11	96.25 ± 0.11	96.50 ± 0.22	96.06 ± 0.29	96.31 ± 0.29	0.44
CORA	85.27 ± 0.15	86.10 ± 0.10	86.07 ± 0.15	86.07 ± 0.15	86.10 ± 0.10	86.13 ± 0.06	0.06
CITESEER	73.13 ± 0.06	73.93 ± 0.15	73.70 ± 0.20	73.70 ± 0.26	73.77 ± 0.15	73.77 ± 0.06	0.23
PUBMED	82.13 ± 0.32	83.27 ± 0.38	83.13 ± 0.32	83.13 ± 0.23	83.13 ± 0.23	83.17 ± 0.21	0.14

Query/value map ablation. To isolate the role of the PV homomorphic query and value maps, we replace only the Q and V transformations with the PVNN fully connected layer [10] while other

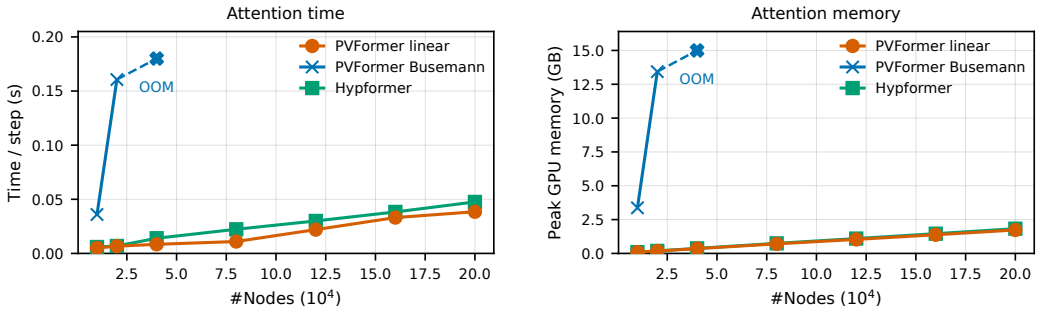


Figure 2: Scalability of PV linear attention and full PV Busemann attention on an A100, with Hypformer linear attention included as a reference. We report forward/backward time per attention step and peak GPU memory as the number of input nodes increases. Full Busemann attention is quadratic in N and becomes infeasible at larger node counts, while both linear-attention branches remain feasible up to 200K nodes.

settings are kept fixed. Table 4 reports the Highest Test under this controlled replacement. According to the table, PVNN-FC is lower than the homomorphic map on all four datasets, with drops of 0.87, 0.33, 0.30, and 0.50 points on Airport, Cora, Citeseer, and PubMed, respectively. Additional gradient diagnostics for these transformations and similarity functions are provided in Section G.2.

Table 4: Ablation replacing PVFORMER homomorphic query/value maps with the PVNN FC.

Q/V map	AIRPORT	CORA	CITESEER	PUBMED
Homomorphic	96.37 ± 0.27	86.20 ± 0.18	73.70 ± 0.20	83.00 ± 0.24
PVNN-FC	95.50 ± 0.43	85.87 ± 0.25	73.40 ± 0.46	82.50 ± 0.30

Similarity and key-map ablation. We next isolate the two geometry components inside full PV attention: the Busemann similarity and the point-to-hyperplane key transformation. The distance-similarity variant in row 2 replaces the Busemann score with the negative PV distance between the query point and a homomorphic key point. The logmap-key variant in row 3 keeps the Busemann score but replaces the point-to-hyperplane key map with the origin log map $\log_0(x)$. Table 5 shows that the PVFORMER design is best on all four datasets. Replacing Busemann attention with distance-based similarity drops Airport, Cora, Citeseer, and PubMed by 6.10, 2.03, 0.93, and 1.93 points, respectively. Replacing the point-to-hyperplane key transformation with an origin log map yields a similar degradation: 6.48, 2.03, 0.77, and 1.97 points on the same datasets. The ablation rows also have much larger standard deviations on Cora, Citeseer, and PubMed, indicating less stable training.

Table 5: Similarity and key-map ablations on small and medium graph benchmarks.

Attention variant	AIRPORT	CORA	CITESEER	PUBMED
PVFORMER	96.37 ± 0.27	86.20 ± 0.18	73.70 ± 0.20	83.00 ± 0.24
Distance similarity	90.27 ± 0.19	84.17 ± 1.33	72.77 ± 1.26	81.07 ± 2.08
Logmap key	89.89 ± 0.20	84.17 ± 1.33	72.93 ± 1.35	81.03 ± 2.04

5.4 Attention Efficiency and Scalability

We compare the two PVFORMER attention mechanisms while varying the number of input nodes, and include Hypformer linear attention as a reference line. We fix the hidden dimension to 128, use one attention head, and measure forward/backward time per attention step and peak GPU memory. The controlled PVFORMER comparison changes only the attention mechanism: as shown in Figure 2, PV linear attention avoids materializing pairwise scores and has complexity $O(Nd_h^2)$, whereas full PV Busemann attention forms an $N \times N$ score matrix and has complexity $O(N^2 d_h)$. Furthermore, compared with Hypformer linear attention, PV linear attention consistently requires less time per

step, indicating higher efficiency under the same input setting. On real large-scale graphs, the attention-branch time per step decreases by 11.7% to 33.9% in the tested settings (Section G.1).

6 Conclusion

We introduced PVFORMER, to the best of our knowledge, the first Transformer architecture operating natively in proper velocity space. By aligning the transformation, similarity, and aggregation primitives of self-attention with PV geometry, PVFORMER combines PV homomorphism transformations, PV Busemann attention, a PV point-to-hyperplane key map, PV midpoint aggregation, and scalable PV linear attention. Experiments under Hypformer protocols show consistent improvements on small and medium graph benchmarks, stable curvature sensitivity, controlled query/value, similarity, and key-map ablations, strong gains on the large-scale graph benchmarks, and substantial improvements on 20News-Groups, while Mini-ImageNet remains above baseline as well. PVFORMER may benefit hierarchical modeling, but requires careful validation before high-stakes use. As future work, we plan to extend our PVFORMER to large language models, exploiting PV space for large-scale deep learning.

Acknowledgements

This work was supported by the EU Horizon Europe projects ELIAS (No. 101120237) and ELLIOT (No. 101214398), and by the FIS project GUIDANCE (No. FIS2023-03251). The authors gratefully acknowledge CINECA and the EuroHPC Joint Undertaking for providing access to the Leonardo supercomputer at CINECA, Italy. The authors declare no competing interests.

References

[1] Haya Alyoussef, Ahmad Bdeir, Diego Coello de Portugal Mecke, Tom Hanika, Niels Landwehr, and Lars Schmidt-Thieme. Hexformer: Hyperbolic vision transformer with exponential map aggregation. *arXiv preprint arXiv:2601.19849*, 2026.

[2] Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Fully hyperbolic convolutional neural networks for computer vision. *arXiv preprint arXiv:2303.15919*, 2023.

[3] Martin R Bridson and André Haefliger. *Metric spaces of non-positive curvature*, volume 319. Springer Science & Business Media, 2013.

[4] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[5] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[6] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31(59-115):2, 1997.

[7] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

[8] 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)*, pages 5672–5686, 2022.

[9] Yu Chen, Lingfei Wu, and Mohammed Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *Advances in neural information processing systems*, 33:19314–19326, 2020.

[10] Ziheng Chen, Zihan Su, Bernhard Schölkopf, and Nicu Sebe. Proper velocity neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026.

[11] Ziheng Chen, Xiaojun Wu, Bernhard Schölkopf, and Nicu Sebe. Building transformation layers for riemannian neural networks, 2026.

[12] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 257–266, 2019.

[13] 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.

[14] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *International conference on machine learning*, pages 1972–1982. PMLR, 2019.

[15] Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*, 2020.

[16] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

[17] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. *Advances in neural information processing systems*, 34:103–115, 2021.

[18] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.

[19] Kyle Helfrich, Devin Willmott, and Qiang Ye. Orthogonal recurrent neural networks with scaled Cayley transform. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1969–1978. PMLR, 2018.

[20] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[22] John M Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.

[23] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[24] Gal Mishne, Zhengchao Wan, Yusu Wang, and Sheng Yang. The numerical stability of hyperbolic representation learning. In *International Conference on Machine Learning*, pages 24925–24949. PMLR, 2023.

[25] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

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

[27] Jongmin Park, Seunghoon Han, Soohwan Jeong, and Sungsu Lim. Hyperbolic heterogeneous graph attention networks. In *Companion Proceedings of the ACM Web Conference 2024*, pages 561–564, 2024.

[28] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

[29] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

[30] Xianglong Shi, Ziheng Chen, Yunhan Jiang, and Nicu Sebe. Intrinsic lorentz neural network. *arXiv preprint arXiv:2602.23981*, 2026.

[31] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210*, 2020.

[32] Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, pages 31613–31632. PMLR, 2023.

[33] Abraham Albert Ungar. *Analytic hyperbolic geometry and Albert Einstein's special theory of relativity*. World Scientific, 2008.

[34] Robert van der Klis, Ricardo Chávez Torres, Max van Spengler, Yuhui Ding, Thomas Hofmann, and Pascal Mettes. Fast and geometrically grounded lorentz neural networks. *arXiv preprint arXiv:2601.21529*, 2026.

[35] 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.

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[37] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

[38] Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in neural information processing systems*, 35:27387–27401, 2022.

[39] Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Sgformer: Simplifying and empowering transformers for large-graph representations. *Advances in neural information processing systems*, 36:64753–64773, 2023.

[40] Zhanghao Wu, Paras Jain, Matthew Wright, Azalia Mirhoseini, Joseph E Gonzalez, and Ion Stoica. Representing long-range context for graph neural networks with global attention. *Advances in neural information processing systems*, 34:13266–13279, 2021.

[41] 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*, pages 3770–3781, 2024.

[42] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377, 2019.

[43] 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, 2021.

[44] Tao Yu and Christopher M De Sa. Representing hyperbolic space accurately using multi-component floats. *Advances in Neural Information Processing Systems*, 34:15570–15581, 2021.

Appendix Contents

A Notation **13**

B Preliminaries **14**

B.1 Riemannian notation and operations 14

B.2 Hyperbolic models and isometries 14

B.3 Lorentz model operations 15

B.4 Proper velocity geometry 15

B.5 Busemann functions 16

C PV Point-to-Hyperplane Keys **16**

C.1 PV hyperplanes and signed margins 17

C.2 Efficient reparameterization for keys 17

C.3 Flat limit 17

D Proofs and Derivations **18**

D.1 Proof of the PV isotropy homomorphism 18

D.2 Derivation of the PV Busemann score 18

D.3 Proof of the PV point-to-hyperplane key map 18

D.4 Proof of PV midpoint aggregation 19

D.5 Proof of PV linear attention 19

D.6 Euclidean inner product from the Busemann definition 19

E Implementation and Experimental Details **20**

E.1 Transformer block implementation 20

E.2 Graph, text, and vision protocols 20

E.3 Reproducibility notes 20

F Numerical Stability of PV **20**

G Additional Experiments **21**

G.1 Efficiency on real large-scale graphs 21

G.2 Gradient stability compared with PVNN 22

H Discussion of Our Homomorphic Transformations **22**

I Limitations **23**

A Notation

Table 6 summarizes the notation used throughout the main paper and appendix. We use lowercase letters for vectors in the proper velocity and Poincaré models, and uppercase letters for ambient Lorentz vectors when the context is clear.

Table 6: Summary of notation used throughout the main paper and appendix.

Notation	Description
$K < 0$	Constant sectional curvature; all formulas keep K negative.
$\mathbb{R}^d$	d-dimensional Euclidean space.
$\langle \cdot, \cdot \rangle$ and $\| \cdot \|$	Euclidean inner product and norm.
$(\mathcal{M}, g)$	Riemannian manifold with metric g.
$T_x\mathcal{M}$, $g_x(\cdot, \cdot)$, and $\| \cdot \|_x$	Tangent space at x, Riemannian metric, and induced tangent norm.
$d_{\mathcal{M}}$, Exp_x, Log_x, and $\mathrm{PT}_{x \to y}$	Geodesic distance, exponential map, logarithmic map, and parallel transport.
$\mathcal{PV}_K^d = \mathbb{R}^d$	d-dimensional proper velocity model equipped with the PV metric.
β_x and ρ_x	Relativistic factors $\beta_x = (1 - K\|x\|^2)^{-1/2}$ and $\rho_x = \beta_x^{-1}$.
$\oplus_U$, $\ominus_U x$, and $t \otimes_U x$	PV gyroaddition, inverse, and scalar gyromultiplication [33, 10].
$\mathbb{L}_K^d$ and $\mathbb{P}_K^d$	Lorentz model and Poincaré ball model with curvature $K < 0$.
$\langle X, Y \rangle_{\mathcal{L}}$ and $\|V\|_{\mathcal{L}}$	Lorentz inner product and the induced norm on tangent vectors.
$X = (X_0, X_s)$ and $\bar{0}$	Time-like/spatial decomposition of a Lorentz vector and the Lorentz origin $((-K)^{-1/2}, 0, \ldots, 0)$.
N, h, and d_h	Number of tokens or nodes, attention-head index, and per-head hidden dimension.
u_i, q_i, k_i, and v_i	PV hidden state, point-valued query, direction-like key, and point-valued value in one attention head.
$\hat{k}_i$, S_{ij}^{PV}, and α_{ij}	Normalized key direction, PV attention logit, and normalized attention weight.
$\mathrm{Mid}_K^{\mathrm{PV}}$	PV midpoint aggregation operator.
γ and B^γ	Geodesic ray and its Busemann function.
$B_K^{\mathrm{PV},u}$	PV Busemann function associated with unit tangent direction $u \in T_0\mathcal{PV}_K^d$.
$\mathcal{H}_{a,p}^{\mathrm{PV}}$	PV hyperplane with base point p and normal $a \in T_p\mathcal{PV}_K^d$.
$\Psi_\theta^{\mathrm{PV}}$	Learned point-to-hyperplane map from a PV point to an origin-tangent key vector.

B Preliminaries

B.1 Riemannian notation and operations

For completeness, we spell out the Riemannian conventions compressed in the main text. A Riemannian manifold $(\mathcal{M}, g)$ is a smooth manifold equipped with a smoothly varying inner product $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ on each tangent space. The induced norm is $\|v\|_x = \sqrt{g_x(v, v)}$. Geodesics generalize straight lines, and the Riemannian distance $d_{\mathcal{M}}(x, y)$ is the length of the shortest geodesic between x and y when it exists. The exponential map Exp_x moves from x along a geodesic with a specified initial tangent vector, while Log_x is its local inverse. Parallel transport $\mathrm{PT}_{x \to y}$ moves tangent vectors along a geodesic while preserving the Riemannian metric. A smooth bijection between two Riemannian manifolds is an isometry if its differential preserves tangent inner products.

B.2 Hyperbolic models and isometries

Hyperbolic space of curvature $K < 0$ can be represented by several equivalent models. The Lorentz model summarized in Table 6 realizes it as the upper sheet of the two-sheeted hyperboloid. Its geodesic distance is

$$d_{\mathbb{L}}(X, Y) = \frac{1}{\sqrt{-K}} \operatorname{arcosh}(K\langle X, Y \rangle_{\mathcal{L}}), \qquad X, Y \in \mathbb{L}_K^d, \tag{23}$$

which is the form used by Lorentz hyperbolic Transformers [41] and Busemann-based hyperbolic learning [17].

The PVNN paper shows that the PV model is isometric to both the Poincaré and Lorentz models [10]. The PV–Poincaré maps are

$$\pi_{\mathcal{PV} \to \mathbb{P}}(x) = \frac{\beta_x}{1 + \beta_x} x, \qquad \pi_{\mathbb{P} \to \mathcal{PV}}(y) = 2\gamma_y^2 y, \qquad \gamma_y = \frac{1}{\sqrt{1 + K\|y\|^2}}. \tag{24}$$

The PV–Lorentz maps are especially simple:

$$\iota_{\mathcal{PV}\to\mathbb{L}}(x) = \left(\sqrt{\|x\|^2 - \frac{1}{K}},\ x\right), \qquad \pi_{\mathbb{L}\to\mathcal{PV}}(X_0, X_s) = X_s. \tag{25}$$

Thus the PV representation stores exactly the spatial part of the Lorentz point, while the time-like component is recovered from the hyperboloid constraint. This observation is used throughout PVFORMER: operations derived in Lorentz space can be pulled back to the unconstrained PV model, avoiding a bounded Poincaré domain and avoiding explicit hyperboloid projection.

B.3 Lorentz model operations

For $X \in \mathbb{L}_K^d$, the tangent space and induced Riemannian metric are

$$T_X\mathbb{L}_K^d = \{V \in \mathbb{R}^{d+1} : \langle V, X\rangle_{\mathcal{L}} = 0\}, \qquad \langle U, V\rangle_X = \langle U, V\rangle_{\mathcal{L}}. \tag{26}$$

For $Z \in T_X\mathbb{L}_K^d$ and $Y \in \mathbb{L}_K^d$, let $\alpha = \sqrt{-K}\|Z\|_X$ and $\beta = K\langle X, Y\rangle_{\mathcal{L}}$. The exponential and logarithmic maps are

$$\mathrm{Exp}_X^{\mathbb{L}}(Z) = \cosh(\alpha)X + \frac{\sinh(\alpha)}{\alpha}Z, \tag{27}$$

$$\mathrm{Log}_X^{\mathbb{L}}(Y) = \frac{\operatorname{arcosh}(\beta)}{\sqrt{\beta^2 - 1}}(Y - \beta X), \tag{28}$$

with the usual continuous extension when $Z = 0$ or $Y = X$. At the origin $\bar{0} = ((-K)^{-1/2}, 0)$, identifying $T_{\bar{0}}\mathbb{L}_K^d$ with spatial vectors $z \in \mathbb{R}^d$ gives

$$\mathrm{Exp}_{\bar{0}}^{\mathbb{L}}(z) = \frac{1}{\sqrt{-K}}\left(\cosh(\sqrt{-K}\|z\|),\ \sinh(\sqrt{-K}\|z\|)\frac{z}{\|z\|}\right), \tag{29}$$

$$\mathrm{Log}_{\bar{0}}^{\mathbb{L}}(Y) = \left(0,\ \frac{\operatorname{arcosh}(\sqrt{-K}Y_0)}{\sqrt{(-K)Y_0^2 - 1}}Y_s\right), \tag{30}$$

again with the zero-vector convention. Parallel transport along the geodesic from X to Y is

$$\mathrm{PT}_{X\to Y}^{\mathbb{L}}(V) = V + \frac{\langle Y, V\rangle_{\mathcal{L}}}{-1/K - \langle X, Y\rangle_{\mathcal{L}}}(X + Y), \tag{31}$$

with the identity map when $X = Y$.

Lorentz transformations are matrices $A \in \mathbb{R}^{(d+1)\times(d+1)}$ satisfying $A^\top JA = J$ for $J = \mathrm{diag}(-1, 1, \ldots, 1)$. The time-orientation-preserving subgroup maps the upper sheet to itself, and the origin-preserving rotations have the form $\mathrm{diag}(1, R)$ with $R \in SO(d)$. This is the Lorentz transformation that becomes $x \mapsto Rx$ after projection to the PV model.

For a unit spatial direction $u \in \mathbb{S}^{d-1}$, the origin-centered geodesic ray is

$$\Gamma_u(t) = \left((-K)^{-1/2}\cosh(\sqrt{-K}t),\ (-K)^{-1/2}\sinh(\sqrt{-K}t)u\right). \tag{32}$$

The corresponding Lorentz Busemann function is

$$B_{\mathbb{L},K}^u(X) = \lim_{t\to\infty}\left(d_{\mathbb{L}}(X, \Gamma_u(t)) - t\right) = \frac{1}{\sqrt{-K}}\log\left(\sqrt{-K}X_0 - \sqrt{-K}\langle X_s, u\rangle\right). \tag{33}$$

B.4 Proper velocity geometry

The PV metric is

$$g_x(u, v) = \langle u, v\rangle + K\beta_x^2\langle x, u\rangle\langle x, v\rangle, \qquad u, v \in T_x\mathcal{PV}_K^d \simeq \mathbb{R}^d. \tag{34}$$

Because $K < 0$, $1 - K\|x\|^2 > 0$ for every $x \in \mathbb{R}^d$, so the model domain is all of $\mathbb{R}^d$. The associated PV gyrovector operations are

$$x \oplus_U y = x + y + \left(\frac{1 - \beta_y}{\beta_y} - K\frac{\beta_x}{1 + \beta_x}\langle x, y\rangle\right)x, \tag{35}$$

$$t \otimes_U x = \frac{\sinh\left(t\,\mathrm{arsinh}(\sqrt{-K}\|x\|)\right)}{\sqrt{-K}\|x\|}x, \qquad t \otimes_U 0 = 0. \tag{36}$$

These operations reduce to ordinary addition and scalar multiplication as $K \to 0^-$ and provide the algebraic structure used by PV homomorphism transformations.

PV Riemannian operations can be obtained by transferring the corresponding Poincaré operations through the isometry in Equation (24). Let $\pi = \pi_{\mathcal{PV}\to\mathbb{P}}$ and let $d_x\pi$ be its differential. For $x, y \in \mathcal{PV}_K^d$ and $v \in T_x\mathcal{PV}_K^d$, define

$$z = (\ominus_U x) \oplus_U y, \qquad \bar{z} = \pi(z). \tag{37}$$

Then the distance is

$$d_{\mathrm{PV}}(x, y) = \frac{2}{\sqrt{-K}} \tanh^{-1}\Big(\sqrt{-K}\, \|\pi((\ominus_U x) \oplus_U y)\|\Big). \tag{38}$$

The logarithmic map can be written as

$$\mathrm{Log}_x(y) = \sigma(x, y)z + \tau(x, y)\langle x, z\rangle x, \tag{39}$$

where

$$\sigma(x, y) = \frac{2}{\sqrt{-K}} \frac{\tanh^{-1}(\sqrt{-K}\|\bar{z}\|)}{\|z\|}, \qquad \tau(x, y) = \frac{2\beta_x}{1 + \beta_x} \frac{\sqrt{-K}\tanh^{-1}(\sqrt{-K}\|\bar{z}\|)}{\|z\|}. \tag{40}$$

At the origin, the operators simplify to

$$\mathrm{Exp}_0(v) = \frac{1}{\sqrt{-K}} \sinh(\sqrt{-K}\|v\|)\frac{v}{\|v\|}, \tag{41}$$

$$\mathrm{Log}_0(y) = \frac{1}{\sqrt{-K}} \operatorname{arsinh}(\sqrt{-K}\|y\|)\frac{y}{\|y\|}, \tag{42}$$

$$d_{\mathrm{PV}}(0, y) = \frac{1}{\sqrt{-K}} \operatorname{arsinh}(\sqrt{-K}\|y\|), \tag{43}$$

with the usual continuous extension at the zero vector. These identity-centered expressions are the forms used to interpret tangent directions and key vectors in PVFORMER.

B.5 Busemann functions

Let (X, d) be a proper geodesic metric space and let $\gamma : [0, \infty) \to X$ be a unit-speed geodesic ray. The Busemann function associated with γ is

$$B^\gamma(x) = \lim_{t\to\infty} \left(d(x, \gamma(t)) - t\right). \tag{44}$$

In Euclidean space, if $\gamma(t) = tv$ for a unit direction v, then

$$B^v(x) = -\langle x, v\rangle. \tag{45}$$

Thus the Busemann function is a curvature-aware analogue of a directional inner product, which motivates Busemann-based hyperbolic similarity [3, 17].

The Lorentz closed form in Equation (33) is the curvature-K directional Busemann potential. Pulling it back through the PV–Lorentz isometry in Equation (25) gives the PV form. For $q \in \mathcal{PV}_K^d$ and $\|u\| = 1$,

$$B^u_{\mathrm{PV},K}(q) = \frac{1}{\sqrt{-K}} \log\Big(\sqrt{1 - K\|q\|^2} - \sqrt{-K}\langle q, u\rangle\Big). \tag{46}$$

For a nonzero key vector $k \in T_0\mathcal{PV}_K^d$, PVFORMER uses the scaled similarity

$$s_{\mathrm{PV}}(q, k) = -\|k\| B^{k/\|k\|}_{\mathrm{PV},K}(q). \tag{47}$$

As $K \to 0^-$, Equation (47) converges to $\langle q, k\rangle$, so the score recovers the Euclidean attention logit in the flat limit.

C PV Point-to-Hyperplane Keys

This section expands the point-to-hyperplane key transformation used in Section 4.3. The construction follows the PV hyperplane and PV MLR derivations in PVNN [10], but the role of the output is different: PVFORMER uses the resulting components as tangent-space key directions for Busemann attention rather than as class logits.

C.1 PV hyperplanes and signed margins

For $p \in \mathcal{PV}_K^d$ and $a \in T_p\mathcal{PV}_K^d$, define the PV hyperplane

$$\mathcal{H}_{a,p}^{\mathrm{PV}} = \left\{ x \in \mathcal{PV}_K^d : g_p\left(\mathrm{Log}_p(x), a\right) = 0 \right\}. \tag{48}$$

Let $\pi = \pi_{\mathcal{PV}\to\mathbb{P}}$ be the PV–Poincaré isometry in Equation (24). Since isometries preserve distances and map totally geodesic hyperplanes to totally geodesic hyperplanes, the PV point-to-hyperplane distance can be computed by moving the hyperplane to the Poincaré ball. Equivalently, the signed margin has the closed form

$$\mathrm{margin}_{a,p}(x) = \frac{1}{\sqrt{-K}} \operatorname{arsinh}\left(\frac{\sqrt{-K}\,\langle (\ominus_U p) \oplus_U x,\, d_p\pi(a) \rangle}{\|d_p\pi(a)\|} \right), \tag{49}$$

where the sign indicates on which side of $\mathcal{H}_{a,p}^{\mathrm{PV}}$ the point lies. This form is the PV analogue of the signed-distance interpretation of Euclidean affine components.

C.2 Efficient reparameterization for keys

A direct parameterization by (p, a) is redundant and expensive for a multi-output key map. Following PVNN, we use a unidirectional parameterization. For the ℓth output component, take $z_\ell \in \mathbb{R}^d \setminus \{0\}$ and $r_\ell \in \mathbb{R}$, and set

$$p_\ell = \mathrm{Exp}_0\left(r_\ell \frac{z_\ell}{\|z_\ell\|} \right), \qquad a_\ell = \mathrm{PT}_{0\to p_\ell}(z_\ell). \tag{50}$$

The resulting signed component can be written without explicitly constructing p_ℓ or a_ℓ:

$$\psi_\ell(x) = \frac{\|z_\ell\|}{\sqrt{-K}} \operatorname{arsinh}\left(\cosh(\sqrt{-K}r_\ell) \frac{\sqrt{-K}}{\|z_\ell\|} \langle x, z_\ell \rangle - \sinh(\sqrt{-K}r_\ell) \sqrt{1 - K\|x\|^2} \right). \tag{51}$$

Stacking m components defines the key map

$$\Psi_\theta^{\mathrm{PV}}(x) = [\psi_1(x), \ldots, \psi_m(x)]^\top \in T_0\mathcal{PV}_K^m \simeq \mathbb{R}^m. \tag{52}$$

For PVFORMER, $\Psi_\theta^{\mathrm{PV}}(u_i)$ is not interpreted as a point in $\mathcal{PV}_K^m$. It is the tangent key vector that determines the Busemann direction in Equation (11). This separation keeps query and value representations point-valued while giving the similarity stage the directional object required by Busemann scoring.

C.3 Flat limit

The key map recovers an affine Euclidean projection as curvature vanishes. Using the Taylor expansions

$$\cosh(\sqrt{-K}r_\ell) = 1 + O(K), \qquad \sinh(\sqrt{-K}r_\ell) = \sqrt{-K}r_\ell + O((-K)^{3/2}), \tag{53}$$

$$\sqrt{1 - K\|x\|^2} = 1 + O(K). \tag{54}$$

the argument of arsinh in Equation (51) is

$$\sqrt{-K}\left(\frac{\langle x, z_\ell \rangle}{\|z_\ell\|} - r_\ell \right) + O((-K)^{3/2}). \tag{55}$$

Since $\operatorname{arsinh}(t) = t + O(t^3)$ near zero,

$$\lim_{K\to 0^-} \psi_\ell(x) = \langle x, z_\ell \rangle - r_\ell \|z_\ell\|. \tag{56}$$

Thus $\Psi_\theta^{\mathrm{PV}}$ is a curvature-aware replacement for an affine key projection, while preserving the usual Euclidean key layer in the flat limit.

D Proofs and Derivations

D.1 Proof of the PV isotropy homomorphism

We prove Theorem 1. Let $R \in SO(d)$ and $\Phi_R^{\mathrm{PV}}(x) = Rx$. Since $R^\top R = I$, $\|Rx\| = \|x\|$ and $\beta_{Rx} = \beta_x$. For tangent vectors $u, v \in T_x\mathcal{PV}_K^d$,

$$g_{Rx}(Ru, Rv) = \langle Ru, Rv\rangle + K\beta_{Rx}^2\langle Rx, Ru\rangle\langle Rx, Rv\rangle \tag{57}$$

$$= \langle u, v\rangle + K\beta_x^2\langle x, u\rangle\langle x, v\rangle = g_x(u, v), \tag{58}$$

so Φ_R^{PV} is a Riemannian isometry.

For gyroaddition, write

$$A(x, y) = \frac{1 - \beta_y}{\beta_y} - K\frac{\beta_x}{1 + \beta_x}\langle x, y\rangle, \qquad x \oplus_U y = x + y + A(x, y)x. \tag{59}$$

Orthogonality gives $\beta_{Rx} = \beta_x$, $\beta_{Ry} = \beta_y$, and $\langle Rx, Ry\rangle = \langle x, y\rangle$, hence $A(Rx, Ry) = A(x, y)$. Therefore

$$R(x \oplus_U y) = Rx + Ry + A(x, y)Rx = Rx \oplus_U Ry. \tag{60}$$

The scalar law follows from the radial form of $t \otimes_U x$ in Equation (36) and the identity $\|Rx\| = \|x\|$:

$$R(t \otimes_U x) = t \otimes_U (Rx). \tag{61}$$

This proves the isometry, gyroaddition, and scalar-compatibility claims.

D.2 Derivation of the PV Busemann score

We derive Theorem 2. Let $q \in \mathcal{PV}_K^d$ and $\|u\| = 1$. Under the PV–Lorentz isometry,

$$q \mapsto X_q = \left(\sqrt{\|q\|^2 - \frac{1}{K}}, q\right) = \left(\frac{\sqrt{1 - K\|q\|^2}}{\sqrt{-K}}, q\right). \tag{62}$$

The unit-speed geodesic ray from the origin in direction u lifts to

$$\Gamma_u(t) = \left((-K)^{-1/2}\cosh(\sqrt{-K}t), (-K)^{-1/2}\sinh(\sqrt{-K}t)u\right). \tag{63}$$

Using the Lorentz distance,

$$d_{\mathbb{L}}(X_q, \Gamma_u(t)) = \frac{1}{\sqrt{-K}}\operatorname{arcosh}(K\langle X_q, \Gamma_u(t)\rangle_{\mathcal{L}}). \tag{64}$$

As $t \to \infty$, the argument of arcosh is asymptotic to

$$\frac{e^{\sqrt{-K}t}}{2}\left(\sqrt{1 - K\|q\|^2} - \sqrt{-K}\langle q, u\rangle\right). \tag{65}$$

Since $\operatorname{arcosh}(s) = \log(2s) + o(1)$ as $s \to \infty$, subtracting t yields

$$B_{\mathrm{PV},K}^u(q) = \frac{1}{\sqrt{-K}}\log\left(\sqrt{1 - K\|q\|^2} - \sqrt{-K}\langle q, u\rangle\right). \tag{66}$$

The logarithm is positive: with $h = \sqrt{-K}$, Cauchy–Schwarz gives $h\langle q, u\rangle \le h\|q\| < \sqrt{1 + h^2\|q\|^2}$. Multiplying by $-\|k\|$ with $u = k/\|k\|$ gives the similarity in Equation (10). For the flat limit, let $a = \langle q, u\rangle$. Then

$$\log\left(\sqrt{1 + h^2\|q\|^2} - ha\right) = -ha + O(h^2), \tag{67}$$

so $-\|k\|h^{-1}\log(\sqrt{1 + h^2\|q\|^2} - ha) = \|k\|a + O(h) \to \langle q, k\rangle$.

D.3 Proof of the PV point-to-hyperplane key map

We prove Theorem 4. For $p \in \mathcal{PV}_K^n$ and $a \in T_p\mathcal{PV}_K^n$, Equation (48) defines the corresponding PV hyperplane and Equation (49) gives its signed margin. The unidirectional parameterization in Equation (50) sets $p_\ell = \mathrm{Exp}_0(r_\ell z_\ell/\|z_\ell\|)$ and $a_\ell = \mathrm{PT}_{0\to p_\ell}(z_\ell)$. Substituting these parameters into the signed-margin formula and simplifying the PV–Poincaré differential yields the component expression in Equation (51), which matches Equation (16). Stacking the m components gives Equation (52), hence $\Psi_\theta^{\mathrm{PV}}(x)$ is exactly the vector of signed PV point-to-hyperplane margins represented in $T_0\mathcal{PV}_K^m \simeq \mathbb{R}^m$. The flat-limit statement follows from the Taylor expansion in Section C.3.

D.4 Proof of PV midpoint aggregation

We prove Theorem 3. Let $v_j \in \mathcal{PV}_K^d$ and lift each value to the Lorentz model:

$$V_j = \left(\sqrt{\|v_j\|^2 - \frac{1}{K}}, v_j\right) = \left(\frac{\sqrt{1 - K\|v_j\|^2}}{\sqrt{-K}}, v_j\right). \tag{68}$$

The Lorentzian midpoint first forms $\widetilde{V} = \sum_j \alpha_j V_j$ and then normalizes it back to the hyperboloid. Its spatial part before normalization is $b = \sum_j \alpha_j v_j$, and its time component is $(-K)^{-1/2} r$ with

$$r = \sum_j \alpha_j \sqrt{1 - K\|v_j\|^2}. \tag{69}$$

The Lorentz norm of $\widetilde{V}$ is

$$\langle \widetilde{V}, \widetilde{V} \rangle_{\mathcal{L}} = -\frac{r^2}{-K} + \|b\|^2 = \frac{r^2 + K\|b\|^2}{K}. \tag{70}$$

The normalization factor that restores curvature K is therefore $(r^2 + K\|b\|^2)^{-1/2}$ on the spatial part, which gives

$$\mathrm{Mid}_K^{\mathrm{PV}}\left(\{(v_j, \alpha_j)\}_{j=1}^N\right) = \frac{b}{\sqrt{r^2 + K\|b\|^2}}. \tag{71}$$

D.5 Proof of PV linear attention

For Theorem 5, let $\lambda_{ij} \geq 0$ be the unnormalized kernel weights and $Z_i = \sum_j \lambda_{ij}$. The normalized midpoint weights are $\alpha_{ij} = \lambda_{ij}/Z_i$. Substituting these weights into Theorem 3 gives

$$b_i = \sum_j \alpha_{ij} v_j = \frac{1}{Z_i} \sum_j \lambda_{ij} v_j = Z_i^{-1} B_i, \qquad r_i = \sum_j \alpha_{ij} \rho_j = \frac{1}{Z_i} \sum_j \lambda_{ij} \rho_j = Z_i^{-1} A_i, \tag{72}$$

where $B_i = \sum_j \lambda_{ij} v_j$, $A_i = \sum_j \lambda_{ij} \rho_j$, and $\rho_j = \sqrt{1 - K\|v_j\|^2}$. Inserting these expressions into the closed-form midpoint in Equation (12) yields

$$o_i^{\mathrm{PVLin}} = \frac{b_i}{\sqrt{r_i^2 + K\|b_i\|^2}} \tag{73}$$

$$= \frac{Z_i^{-1} B_i}{\sqrt{Z_i^{-2} A_i^2 + K Z_i^{-2} \|B_i\|^2}} \tag{74}$$

$$= \frac{Z_i^{-1} B_i}{Z_i^{-1} \sqrt{A_i^2 + K\|B_i\|^2}} \tag{75}$$

$$= \frac{B_i}{\sqrt{A_i^2 + K\|B_i\|^2}}. \tag{76}$$

Thus the common factor Z_i^{-1} cancels from the numerator and denominator. The Euclidean linear-attention denominator Z_i is replaced by the curvature-aware normalization $\sqrt{A_i^2 + K\|B_i\|^2}$ induced by the PV midpoint. As $K \to 0^-$, $\rho_j = \sqrt{1 - K\|v_j\|^2} \to 1$, so $A_i \to Z_i$ and the expression becomes the Euclidean linear attention output

$$\frac{\sum_j \lambda_{ij} v_j}{\sum_j \lambda_{ij}}. \tag{77}$$

D.6 Euclidean inner product from the Busemann definition

For completeness, we recall the identity used to motivate Busemann attention. Let $u \in \mathbb{R}^d$ with $\|u\| = 1$ and define

$$B^u(x) = \lim_{t \to \infty} (\|x - tu\| - t). \tag{78}$$

Then

$$\|x - tu\| - t = \frac{\|x - tu\|^2 - t^2}{\|x - tu\| + t} = \frac{\|x\|^2 - 2t\langle x, u\rangle}{\|x - tu\| + t}. \tag{79}$$

Since $\|x - tu\|/t \to 1$, the denominator divided by t tends to 2, and therefore

$$B^u(x) = -\langle x, u\rangle. \tag{80}$$

For any nonzero y, setting $u = y/\|y\|$ gives

$$\langle x, y\rangle = -\|y\| B^{y/\|y\|}(x), \tag{81}$$

which is the Euclidean identity generalized by PV Busemann attention.

E Implementation and Experimental Details

E.1 Transformer block implementation

The implemented PVFORMER block keeps the three attention primitives separated. For full attention, the query and value branches use the homomorphic PV maps in Section 4.1; the key branch uses the point-to-hyperplane map in Section C; the score is the PV Busemann logit in Equation (11); and values are aggregated by the PV midpoint in Equation (12). For large graphs, the block switches to the PV linear attention branch in Section 4.4. This branch uses point-valued PV queries, keys, and values and avoids materializing the $N \times N$ Busemann score matrix.

For numerical stability, implementation uses a small ε in key normalization and logarithm arguments, as shown in Equation (11). Orthogonal maps are parameterized through skew-symmetric generators so that the resulting matrices remain in $SO(d_h)$. This preserves the geometric interpretation of the homomorphic Q/V updates while keeping all trainable parameters unconstrained in Euclidean memory.

E.2 Graph, text, and vision protocols

The experiments follow the Hypformer-style protocol used in the main text. Small and medium graph experiments report F1-score on Disease and accuracy on Airport, Cora, Citeseer, and PubMed. Large graph experiments report ROC-AUC on ogbn-proteins and accuracy on the large-scale accuracy benchmarks shown in Table 1. Text and vision experiments evaluate Mini-ImageNet and 20News-Groups under k-nearest-neighbor graphs with $k \in \{5, 10, 15, 20\}$ and a graph-free variant.

The ablations in Tables 4 and 5 are designed to isolate geometric components. The PVNN-FC ablation replaces only the homomorphic query/value maps. The similarity/key ablation keeps the graph branch, optimizer, curvature, dropout, homomorphic Q/V maps, and midpoint aggregation fixed while changing either the similarity function or the key map. The attention scalability experiment in Figure 2 uses synthetic attention states with hidden dimension 128 and one head, and reports forward/backward time per attention step and peak GPU memory.

E.3 Reproducibility notes

The reported tables use the values in the main manuscript as the experimental record. For small and medium graph ablations, the explicit ablation seeds are five different ones when the table reports a five-seed mean and standard deviation. Large-graph experiments use the scalable PV linear attention path, because full Busemann attention is quadratic in the number of nodes and is not feasible at the large-graph scale shown in Figure 2. Absolute local paths, machine-specific process identifiers, and private infrastructure details are omitted from the manuscript to keep the supplementary material portable and anonymous. All experiments were conducted on A100 80GB GPUs.

F Numerical Stability of PV

Tables 7 to 9 summarize stability at the model, attention-block, and training levels. The model-level results from PVNN [10] show no reported PV failures across the tested radii, with constraint violations inapplicable to its unconstrained representation. Our attention and training diagnostics

use FP32 and the official stabilized Hypformer baseline. The attention probe feeds both models the same 512 embeddings at each radius. The Lorentz baseline retains its native safeguards, including time-coordinate reconstruction and argument clamping. PVFORMER retains nonzero input gradients with a spread no greater than 1.20, whereas the Lorentz baseline has 67.6% to 88.9% exactly zero gradients for $r \geq 5$. Increasing $k = |K|$ from 0.5 to 64 reduces the Lorentz singularity margins by approximately 72× on Airport and 93× on Citeseer, while the PV margins remain comparatively stable. The fixed-radius probe measures robustness under synthetic stress, and no Lorentz training run diverged in the curvature sweep.

Table 7: Model-level failure and manifold-constraint violation rates (%), reproduced from PVNN [10]. N/A denotes an inapplicable constraint check. These rates do not measure boundary gradient saturation.

Metric (%)	Model	Geodesic radius r									
		1	5	10	20	50	75	100	150	200	1000
Failure rate	PV	0	0	0	0	0	0	0	0	0	0
Failure rate	Poincaré	0	0	0	0	0	0	0	0	0	0
Failure rate	Hyperboloid	0	0	0	4.23	64.42	79.63	88.26	96.43	100	100
Violation rate	PV	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Violation rate	Poincaré	0	0	0	0	0	0	0	0	0	0
Violation rate	Hyperboloid	32.50	92.36	99.76	100	100	100	100	100	100	100

Table 8: Attention-block stability with 512 embeddings at each radius. Max/min is the gradient spread over nonzero input gradients, and zero-grad is the percentage of exactly zero input gradients. Lorentz denotes the official stabilized Hypformer implementation.

Input radius r	PV max/min	PV zero-grad	Lorentz max/min	Lorentz zero-grad
0.5	1.01	0%	3.35	0%
1	1.03	0%	23.9	0%
2	1.06	0%	1.2×10^3	0%
5	1.06	0%	1.7×10^{28}	67.6%
10	1.12	0%	1.9×10^{10}	88.6%
50	1.20	0%	2.0×10^9	88.9%

Table 9: Minimum distance of the Lorentz arcosh argument or the PV Busemann logarithm argument from its singularity during training as $k = |K|$ varies. The collapse row summarizes the change from $k = 0.5$ to $k = 64$. Margins are interpreted within each geometry.

$k = \|K\|$	Airport: Lorentz	Airport: PV	Citeseer: Lorentz	Citeseer: PV
0.5	2.466	0.526	0.574	0.952
1	1.530	0.462	0.350	0.939
4	0.476	0.642	0.098	0.886
16	0.125	0.570	0.024	0.722
64	0.0344	0.547	0.00614	0.763
Collapse	72×	flat	93×	1.25×

G Additional Experiments

G.1 Efficiency on real large-scale graphs

We extend the attention efficiency comparison in Figure 2 to real large-scale graphs. Table 10 reports the attention-branch time per step on ogbn-arxiv with the full graph and on Amazon2M with two batch sizes. Compared with Hypformer, PVFORMER reduces this time by 33.9% on ogbn-arxiv and by 29.7% and 11.7% on Amazon2M with batches of 900K and 100K nodes, respectively.

Table 10: Attention-branch time per step (ms) on real large-scale graphs. Time reduction is measured relative to Hypformer.

Method	ogbn-arxiv (full graph, 169K)	Amazon2M (batch 900K)	Amazon2M (batch 100K)
Hypformer	204.1 ± 1.2	929.8 ± 2.2	89.5 ± 0.2
PVFORMER	135.0 ± 0.3	653.6 ± 2.2	79.0 ± 0.3
Time reduction	**33.9%**	**29.7%**	**11.7%**

During Cora training, Hypformer performs an average of 12.5 constraint-restoration operations per forward pass, whereas PVFORMER requires none because PV coordinates are unconstrained. This count includes time-coordinate reconstruction, hyperboloid projection, and midpoint renormalization. Tangent-space maps themselves are excluded, while projections invoked inside them are counted. The fractional average reflects dropout being active only during training.

G.2 Gradient stability compared with PVNN

Table 11 reports the within-batch gradient spread under different Q/V transformations and similarity functions in the synthetic radius probes. At $r = 50$, replacing our homomorphic Q/V map with PVNN-FC while retaining Busemann similarity increases the spread from 1.20 to 1.14×10^3. With PVNN-FC fixed, replacing Busemann similarity with distance increases it further to 1.71×10^3. With homomorphic Q/V maps, the two similarities yield spreads of 1.20 and 1.25, respectively. The homomorphic maps preserve norms and constrain expressiveness, so these results characterize gradient balance for the tested designs and should be interpreted together with the expressiveness discussion in Section H.

Table 11: Within-batch gradient spread of the PV attention block under different Q/V transformations and similarity functions.

Input radius	Homomorphic (ours) + Busemann (ours)	PVNN-FC + Busemann (ours)	Homomorphic (ours) + distance	PVNN-FC + distance
5	1.07	12.8	1.15	18.0
10	1.13	551	1.20	726
50	1.20	1.14×10^3	1.25	1.71×10^3

H Discussion of Our Homomorphic Transformations

Current maximality and the choice of $SO(d)$. In Euclidean space, linear homomorphisms admit the general and tractable form $x \mapsto Ax$. Our construction identifies PV gyro-homomorphisms through the sufficient condition of being an origin-preserving isometry. For equal-dimensional PV transformations, the maximal family satisfying this condition is the orthogonal group $\{x \mapsto Rx \mid R \in O(d)\}$. This maximality concerns origin-preserving isometries, while a classification of more general PV gyro-homomorphisms is outside the scope of this work. Theorem 3.1 of Helfrich et al. [19] provides a scaled Cayley parameterization of $O(d)$ through an additional diagonal sign matrix D, with the number ρ of its negative entries treated as a hyperparameter. For simpler optimization, we use $SO(d)$.

Expressiveness, dimension-changing cases, and future work. Our sufficient-condition construction identifies a tractable subclass of PV gyro-homomorphisms. Its rotation constraint preserves norms and may limit expressiveness relative to more general transformations. A dimension-reducing map with $d_k < d$ is not covered by our current homomorphic construction, since the present work considers only equal-dimensional transformations. Characterizing broader PV gyro-homomorphisms and developing dimension-changing homomorphic maps are two important directions for future work.

I Limitations

Due to limited computational resources, we have not trained PVFORMER at the scale of large language models. We plan to apply it to large language models in future work. Our homomorphic Q/V maps are restricted to equal-dimensional rotations, with expressiveness and dimension-changing extensions discussed in Section H. The numerical stability diagnostics characterize the tested operators and stress regimes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the native PV-space Transformer claim with the qualifier "to the best of our knowledge" and distinguish positive table results from the remaining Mini-ImageNet and reproducibility caveats.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: See Appendix I.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: In Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The code repository is `https://github.com/Longchentong/PVFormer`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The code repository is `https://github.com/Longchentong/PVFormer`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: In Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: The main PVFormer table rows report mean and standard deviation.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: A100 GPU.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work uses public benchmark-style datasets and standard representation-learning experiments, and the manuscript avoids deanonymizing information.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

Justification: In conclusion.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The work does not release high-risk generative models, scraped datasets, or dual-use systems.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The paper cites benchmark and method papers where available, but the paper adopts CC BY 4.0.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The paper introduces a method and experimental results but does not release a new dataset or standalone asset package.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The work does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The work does not involve human-subject experiments.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLM usage is not part of the core PVFORMER method or experimental claims in this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.