Fully Hyperbolic Neural Networks

Paper sharing

Presenter: Menglin Yang

menglin.yang@yale.edu
Department of Computer Science
Advisor: Rex Ying
Yale University

Nov 1, 2023

Fully Hyperbolic Neural Networks

Weize Chen[1,3*] **Xu Han**[1,3*] **Yankai Lin**[5] **Hexu Zhao**[1,3]
Zhiyuan Liu[1,2,3,6†] **Peng Li**[4‡] **Maosong Sun**[1,2,3,6†] **Jie Zhou**[5]
[1]Department of Computer Science and Technology, Tsinghua University
[2]International Innovation Center of Tsinghua University,
[3]Institute for Artificial Intelligence, Tsinghua University,
[4]Institute for AI Industry Research (AIR), Tsinghua University
[5]Pattern Recognition Center, WeChat AI, Tencent Inc
[6]Beijing Academy of Artificial Intelligence
{chenwz21,hanxu17}@mails.tsinghua.edu.cn
{liuzy,sms}@tsinghua.edu.cn

Figure 1: The research work was published in ACL 2022.

Data Structure and Curvature

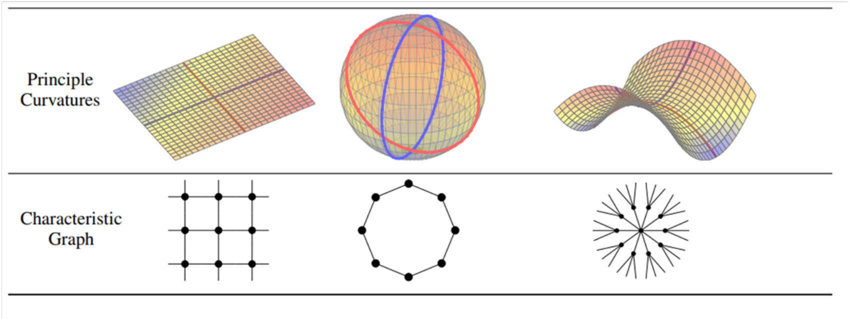


Figure 2: The quality of the representations achieved by embeddings is determined by how well the geometry of the embedding space matches the structure of the data [GSGR18].

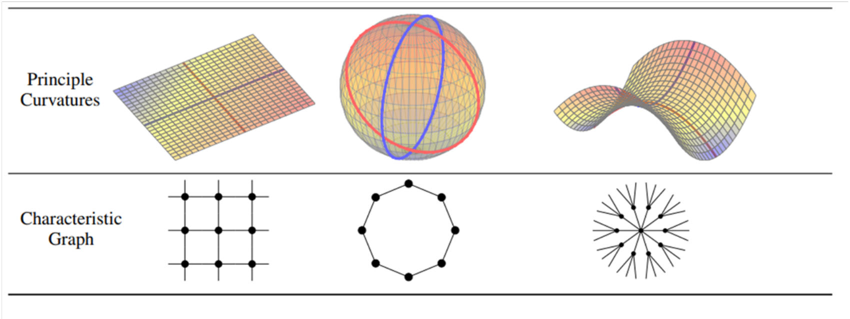


Figure 2: The quality of the representations achieved by embeddings is determined by how well the geometry of the embedding space matches the structure of the data [GSGR18].

What embedding space geometry is optimal for data?

Tree-like Data and Hyperbolic Space

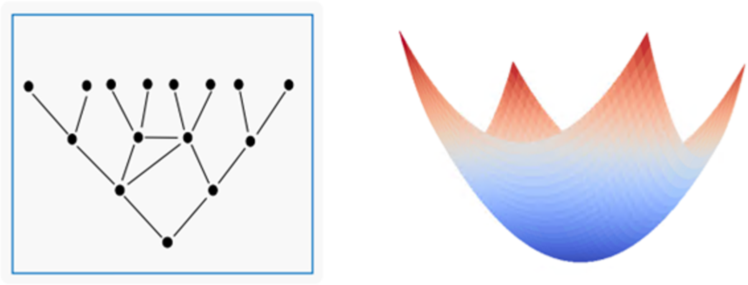


Figure 3: The (tree-like) data in the left subfigure can be considered as a discrete approximation to the (hyperbolic) manifold M in the right subfigure; on the other hand, the manifold M can also be approximated as the tree-like data.

- **Providing Geometric Prior [KPK+10].** "Trees, even infinite ones, allow nearly isometric embeddings into hyperbolic spaces".

- **Providing Geometric Prior [KPK+10].** "Trees, even infinite ones, allow nearly isometric embeddings into hyperbolic spaces".
- **Reduced Dimensionality [NK17, NK18].** "Poincaré embeddings perform very well on these datasets and especially in the low-dimensional regime (dim=10) outperform Euclidean embeddings".

- **Providing Geometric Prior [KPK$^+$10].** "Trees, even infinite ones, allow nearly isometric embeddings into hyperbolic spaces".
- **Reduced Dimensionality [NK17, NK18].** "Poincaré embeddings perform very well on these datasets and especially in the low-dimensional regime (dim=10) outperform Euclidean embeddings".
- **Improved Similarity Measures [GBH18].** Hyperbolic space embeddings tend to provide better similarity measures for tree-like data.

Embedding Tree-like Data in Hyperbolic Space

- **Providing Geometric Prior [KPK+10].** "Trees, even infinite ones, allow nearly isometric embeddings into hyperbolic spaces".
- **Reduced Dimensionality [NK17, NK18].** "Poincaré embeddings perform very well on these datasets and especially in the low-dimensional regime (dim=10) outperform Euclidean embeddings".
- **Improved Similarity Measures [GBH18].** Hyperbolic space embeddings tend to provide better similarity measures for tree-like data.
- **Lower Distortion [SWT+19].** In graph embedding setting, we can embed any tree with arbitrarily low distortion to hyperbolic space in graph embedding setting.

Embedding Tree-like Data in Hyperbolic Space

- **Providing Geometric Prior [KPK+10].** "Trees, even infinite ones, allow nearly isometric embeddings into hyperbolic spaces".
- **Reduced Dimensionality [NK17, NK18].** "Poincaré embeddings perform very well on these datasets and especially in the low-dimensional regime (dim=10) outperform Euclidean embeddings".
- **Improved Similarity Measures [GBH18].** Hyperbolic space embeddings tend to provide better similarity measures for tree-like data.
- **Lower Distortion [SWT+19].** In graph embedding setting, we can embed any tree with arbitrarily low distortion to hyperbolic space in graph embedding setting.
- **Lower Generalization Bound [SNW+21].** "In fact, if the true dissimilarity measure $\Delta*$ is given by the graph distance of a weighted tree, where Euclidean space cannot represent their metric structure well, then HOE can perform better than EOE with sufficient number S of ordinal data"

Examples of Tree-like Data

- **Networks.**
- **Words, sentences and documents.**
- **Images.**
- **Bio-structures.**
- **Neural network models.**

Formally, an n-dimensional Euclidean model is the Riemannian manifold $\mathrm{E}^n_K = (\mathrm{E}^n, g^K)$. K is the zero since Euclidean space is flat. $g^K_{\mathbf{x}} = \mathrm{diag}(1, \ldots, 1)$ is the Riemannian metric tensor. Each point in E^n_K has the form $\mathbf{x} = \left(x^1, \cdots, x^n\right)$, $x^i \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^n$. E^n is a point set satisfying

$$\mathrm{E}^n_K := \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{x}, \mathbf{x} \rangle >= 0\},$$

Formally, an n-dimensional Euclidean model is the Riemannian manifold $\mathrm{E}_K^n = (\mathrm{E}^n, g^K)$. K is the zero since Euclidean space is flat. $g_{\mathbf{x}}^K = \mathrm{diag}(1, \ldots, 1)$ is the Riemannian metric tensor. Each point in E_K^n has the form $\mathbf{x} = \left(x^1, \cdots, x^n\right)$, $x^i \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^n$. E^n is a point set satisfying

$$\mathrm{E}_K^n := \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{x}, \mathbf{x} \rangle >= 0\},$$

where

$$\langle \mathbf{x}, \mathbf{y} \rangle := x^1 y^1 + \cdots + x^n y^n = \mathbf{x}^T \mathrm{diag}(1, \ldots, 1)\mathbf{y},$$

where $\langle \mathbf{x}, \mathbf{y} \rangle$ is the Euclidean inner product.

Definition of Lorentz Model

Formally, an n-dimensional Lorentz model is the Riemannian manifold $\mathcal{L}_K^n = (\mathcal{L}^n, g^K)$. $K (K < 0)$ is the constant negative curvature. $g_{\mathbf{x}}^K = \mathrm{diag}(-1, 1, \ldots, 1)$ is the Riemannian metric tensor.

Formally, an n-dimensional Lorentz model is the Riemannian manifold $\mathcal{L}_K^n = (\mathcal{L}^n, g^K)$. $K(K < 0)$ is the constant negative curvature. $g_{\mathbf{x}}^K = \mathrm{diag}(-1, 1, \ldots, 1)$ is the Riemannian metric tensor.Each point in $\mathcal{L}_K^n$ has the form $\mathbf{x} = (x^t, \mathbf{x}^s)$, $x^t \in \mathbb{R}$, $\mathbf{x}^s \in \mathbb{R}^n$. $\mathcal{L}^n$ is a point set satisfying

$$\mathcal{L}^n := \left\{ \mathbf{x} \in \mathbb{R}^{n+1} \mid \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = \frac{1}{K}, x^t > 0 \right\},$$

Formally, an n-dimensional Lorentz model is the Riemannian manifold $\mathcal{L}_K^n = (\mathcal{L}^n, g^K)$. $K (K < 0)$ is the constant negative curvature. $g_{\mathbf{x}}^K = \mathrm{diag}(-1, 1, \ldots, 1)$ is the Riemannian metric tensor.Each point in $\mathcal{L}_K^n$ has the form $\mathbf{x} = (x^t, \mathbf{x}^s)$, $x^t \in \mathbb{R}$, $\mathbf{x}^s \in \mathbb{R}^n$. $\mathcal{L}^n$ is a point set satisfying

$$\mathcal{L}^n := \left\{ \mathbf{x} \in \mathbb{R}^{n+1} \mid \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = \frac{1}{K}, x^t > 0 \right\},$$

where

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} := -x^t y^t + (\mathbf{x}^s)^T \mathbf{y}^s = \mathbf{x}^T \mathrm{diag}(-1, 1, \ldots, 1)\mathbf{y},$$

where $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}$ is the Lorentzian inner product.

Origin Point

In Euclidean space, the origin point is

$$\mathbf{0}_E = \mathbf{0}_n.$$

Origin Point

In Euclidean space, the origin point is

$$\mathbf{0}_E = \mathbf{0}_n.$$

In the Lorentz model, the origin point is defined as

$$\mathbf{0}_{\mathcal{L}} := \left(\frac{1}{\sqrt{|K|}}, \mathbf{0}_n \right).$$

Origin Point

In the Lorentz model, the defined origin point is

$$\mathbf{0}_{\mathcal{L}} = \left(\underbrace{\frac{1}{\sqrt{|K|}}}_{\text{time-like dimension}}, \underbrace{\mathbf{0}_n}_{\text{space-like dimension}} \right).$$

Origin Point

In the Lorentz model, the defined origin point is

$$\mathbf{0}_{\mathcal{L}} = \left(\underbrace{\frac{1}{\sqrt{|K|}}}_{\text{time-like dimension}}, \underbrace{\mathbf{0}_n}_{\text{space-like dimension}} \right).$$

According to the definition of Lorentz's inner product,

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} := -x^t y^t + (\mathbf{x}^s)^T \mathbf{y}^s,$$

Origin Point

In the Lorentz model, the defined origin point is

$$\mathbf{0}_{\mathcal{L}} = \left(\underbrace{\frac{1}{\sqrt{|K|}}}_{\text{time-like dimension}}, \underbrace{\mathbf{0}_n}_{\text{space-like dimension}} \right).$$

According to the definition of Lorentz's inner product,

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} := -x^t y^t + (\mathbf{x}^s)^T \mathbf{y}^s,$$

we have

$$\langle \mathbf{0}_{\mathcal{L}}, \mathbf{0}_{\mathcal{L}} \rangle_{\mathcal{L}} = -\frac{1}{\sqrt{|K|}} \cdot \frac{1}{\sqrt{|K|}} + \mathbf{0}_n^T \mathbf{0}_n = -\frac{1}{-K} = \frac{1}{K}.$$

Then we know that the defined $\mathbf{0}_{\mathcal{L}}$ is on the Lorentz manifold.

The Time-like Dimension

If a point $\mathbf{x}$ is on Lorentz model, then we have

$$\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -x^t x^t + (\mathbf{x}^s)^T \mathbf{x}^s = \frac{1}{K},$$

The Time-like Dimension

If a point $\mathbf{x}$ is on Lorentz model, then we have

$$\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -x^t x^t + (\mathbf{x}^s)^T \mathbf{x}^s = \frac{1}{K},$$

and further, we can derive that

$$(x^t)^2 = \|\mathbf{x}^s\|^2 - \frac{1}{K}.$$

The Time-like Dimension

If a point $\mathbf{x}$ is on Lorentz model, then we have

$$\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -x^t x^t + (\mathbf{x}^s)^T \mathbf{x}^s = \frac{1}{K},$$

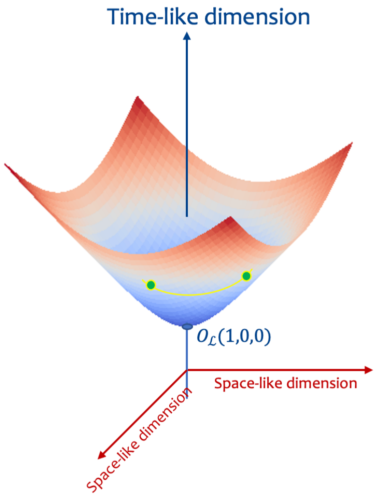
and further, we can derive that

$$(x^t)^2 = \|\mathbf{x}^s\|^2 - \frac{1}{K}.$$

Since $x^t > 0$,

$$x^t = \sqrt{\|\mathbf{x}^s\|^2 - \frac{1}{K}}.$$

Illustration of Lorentz Model



2-dimensional Lorentz Model

Suppose we have a Lorentz model denoted as $\mathcal{L}_K^n$, where $K = -1$ represents the curvature and $n = 2$ represents the dimensions.
Within this model, there are two points: $O_{\mathcal{L}} = (1, 0, 0)$ and $P = (1.732, 1, 1)$.

Suppose we have a Lorentz model denoted as $\mathcal{L}_K^n$, where $K = -1$ represents the curvature and $n = 2$ represents the dimensions.
Within this model, there are two points: $O_{\mathcal{L}} = (1, 0, 0)$ and $P = (1.732, 1, 1)$.
If we directly add them:

$$O_{\mathcal{L}}(1, 0, 0) + P(1.732, 1, 1) = (2.732, 1, 1).$$

The result can be easily verified not to satisfy the constraints of the Lorentz model, which is:

$$-2.732^2 + 1 + 1 \neq -1.$$

Suppose we have a Lorentz model denoted as $\mathcal{L}_K^n$, where $K = -1$ represents the curvature and $n = 2$ represents the dimensions.
Within this model, there are two points: $O_{\mathcal{L}} = (1, 0, 0)$ and $P = (1.732, 1, 1)$.
If we directly add them:

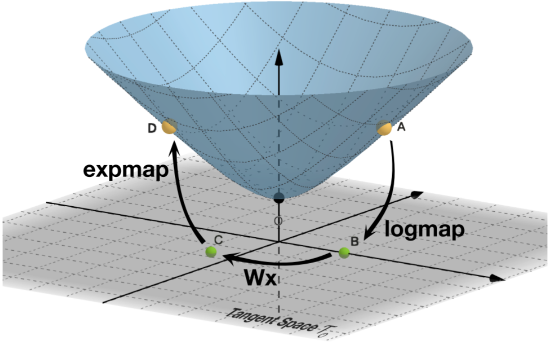
$$O_{\mathcal{L}}(1, 0, 0) + P(1.732, 1, 1) = (2.732, 1, 1).$$

The result can be easily verified not to satisfy the constraints of the Lorentz model, which is:

$$-2.732^2 + 1 + 1 \neq -1.$$

Euclidean operations do not work directly in the Lorentz model!!!

Lorentz Tangent Transformation



(a) Linear layer formalized in tangent space

logmap : $\mathcal{T}_{\mathbf{0}}\mathcal{L} \to \mathcal{L}$

expmap : $\mathcal{L} \to \mathcal{T}_{\mathbf{0}}\mathcal{L}$

In mathematics, the tangent space of a manifold is a generalization of **tangent lines to curves** in two-dimensional space and **tangent planes to surfaces** in three-dimensional space in higher dimensions.

[1] https://en.wikipedia.org/wiki/Tangent_space

Tangent Space[1]

In mathematics, the tangent space of a manifold is a generalization of **tangent lines to curves** in two-dimensional space and **tangent planes to surfaces** in three-dimensional space in higher dimensions.

In the context of physics, the tangent space to a manifold at a point can be viewed as **the space of possible velocities** for a particle moving on the manifold.

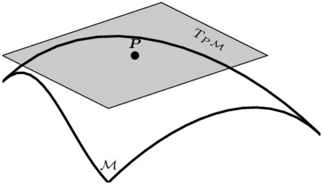
[1] https://en.wikipedia.org/wiki/Tangent_space

Tangent Space[1]

In mathematics, the tangent space of a manifold is a generalization of **tangent lines to curves** in two-dimensional space and **tangent planes to surfaces** in three-dimensional space in higher dimensions.

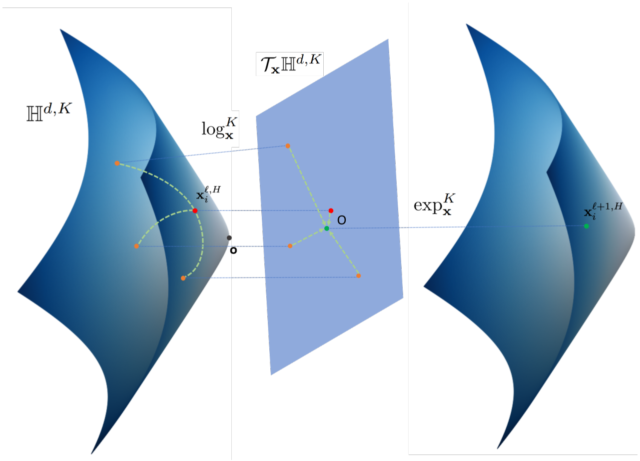
In the context of physics, the tangent space to a manifold at a point can be viewed as **the space of possible velocities** for a particle moving on the manifold.

In differential geometry, one can attach to every point $\mathbf{x}$ of a differentiable manifold a tangent spacea real vector space that intuitively **contains the possible directions** in which one can tangentially pass through $\mathbf{x}$.



[1] https://en.wikipedia.org/wiki/Tangent_space

Tangent Space Hyperbolic GNN [CYRL19]



logmap : $\mathcal{T}_{\mathbf{0}}\mathcal{L} \to \mathcal{L}$

expmap : $\mathcal{L} \to \mathcal{T}_{\mathbf{0}}\mathcal{L}$

Problems

According to the work of *Fully Hyperbolic Neural Networks*, there are two limitations:

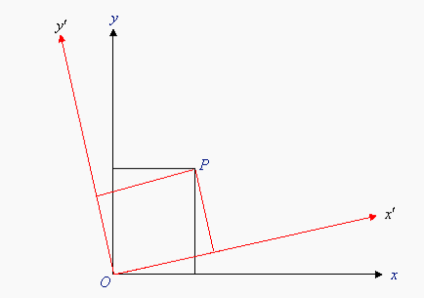
- **Unstable**. The logarithmic and exponential maps require a series of hyperbolic and inverse hyperbolic functions. The compositions of these functions are complicated and usually range to infinity, weakening the stability of models.

Problems

According to the work of *Fully Hyperbolic Neural Networks*, there are two limitations:

- **Unstable**. The logarithmic and exponential maps require a series of hyperbolic and inverse hyperbolic functions. The compositions of these functions are complicated and usually range to infinity, weakening the stability of models.
- **Limited capabilities.** Existing transformations do not include the Lorentz boost but only rotation.

Definition (Lorentz Rotation). Lorentz rotation is the rotation of the spatial coordinates. The Lorentz rotation matrices are given by $\mathbf{R} = \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \tilde{\mathbf{R}} \end{bmatrix}$, where $\tilde{\mathbf{R}}^\top \tilde{\mathbf{R}} = \mathbf{I}$ and $\det(\tilde{\mathbf{R}}) = 1$, i.e., $\tilde{\mathbf{R}} \in \mathbf{SO}(n)$ is a special orthogonal matrix.



Definition (Lorentz Boost). Lorentz boost describes relative motion with constant velocity and without rotation of the spatial coordinate axes. Given a velocity $v \in \mathbb{R}^n$ (ratio to the speed of light), $||v|| < 1$ and $\gamma = \frac{1}{\sqrt{1-||v||^2}}$, the Lorentz boost matrices are given by

$$B = \begin{bmatrix} \gamma & -\gamma v^T \\ -\gamma v & I + \frac{\gamma^2}{1+\gamma} v v^T \end{bmatrix}.$$

Definition (Lorentz Boost). Lorentz boost describes relative motion with constant velocity and without rotation of the spatial coordinate axes. Given a velocity $v \in \mathbb{R}^n$ (ratio to the speed of light), $||v|| < 1$ and $\gamma = \frac{1}{\sqrt{1-||v||^2}}$, the Lorentz boost matrices are given by

$$B = \begin{bmatrix} \gamma & -\gamma v^T \\ -\gamma v & I + \frac{\gamma^2}{1+\gamma} v v^T \end{bmatrix}.$$

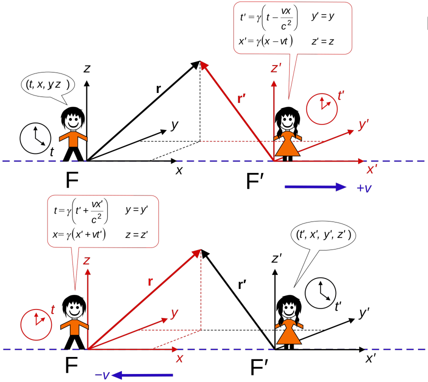
- The term γ appears in the matrix, representing the Lorentz factor.
- v is the relative velocity vector between the two observers.
- v^T is the transpose of the velocity vector.
- I is the identity matrix.
- The term $\frac{\gamma^2}{1+\gamma} v v^T$ accounts for the directionality of the boost, based on the direction of the relative velocity.

Definition (Lorentz Boost). Lorentz boost describes relative motion with constant velocity and without rotation of the spatial coordinate axes. Given a velocity $v \in \mathbb{R}^n$ (ratio to the speed of light), $||v|| < 1$ and $\gamma = \frac{1}{\sqrt{1-||v||^2}}$, the Lorentz boost matrices are given by

$$B = \begin{bmatrix} \gamma & -\gamma v^T \\ -\gamma v & I + \frac{\gamma^2}{1+\gamma} v v^T \end{bmatrix}.$$

- The term γ appears in the matrix, representing the Lorentz factor.
- v is the relative velocity vector between the two observers.
- v^T is the transpose of the velocity vector.
- I is the identity matrix.
- The term $\frac{\gamma^2}{1+\gamma} v v^T$ accounts for the directionality of the boost, based on the direction of the relative velocity.

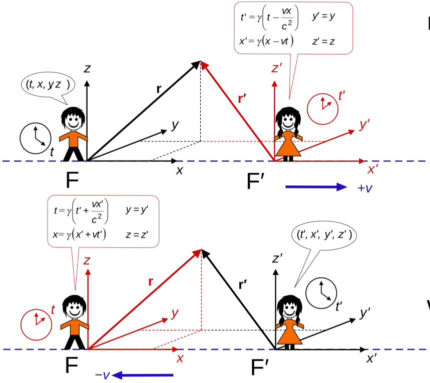
The transformations arise from the postulate of special relativity, which states that the laws of physics are the same in all inertial frames of reference.



Consider two frames, F (stationary) and F' (moving with velocity v relative to F). The Lorentz transformation for a boost in the x-direction is given by:

$$t' = \gamma\left(t - \frac{vx}{c^2}\right)$$

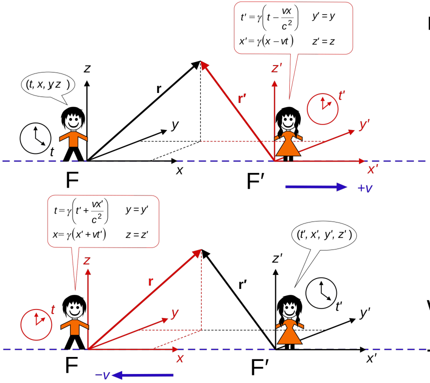
$$x' = \gamma(x - vt)$$



Consider two frames, F (stationary) and F' (moving with velocity v relative to F). The Lorentz transformation for a boost in the x-direction is given by:

$$
\begin{aligned}
t' &= \gamma\left(t - \frac{vx}{c^2}\right) \\
x' &= \gamma(x - vt) \\
y' &= y \\
z' &= z
\end{aligned},
$$

where γ is the Lorentz factor, defined as $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$.



Consider two frames, F (stationary) and F' (moving with velocity v relative to F). The Lorentz transformation for a boost in the x-direction is given by:

$$
\begin{aligned}
t' &= \gamma\left(t - \frac{vx}{c^2}\right) \\
x' &= \gamma(x - vt) \\
y' &= y \\
z' &= z
\end{aligned},
$$

where γ is the Lorentz factor, defined as $\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}}$.

This transformation indicates that time and space mix in a moving frame, leading to effects like time dilation and length contraction.

Definition (The Lorentz linear transformation)[DWGJ21]. For any $\mathbf{x} \in \mathcal{L}$, the Lorentz linear transformation is defined as

$$\mathbf{y} = \mathbf{W}\mathbf{x}$$
$$\text{s.t. } \mathbf{W} = \begin{bmatrix} 1 & \mathbf{0}^\top \\ \mathbf{0} & \widehat{\mathbf{W}} \end{bmatrix}, \widehat{\mathbf{W}}^\top \widehat{\mathbf{W}} = \mathbf{I},$$

where $\mathbf{W}$ is a transformation matrix, and $\widehat{\mathbf{W}}$ is called a transformation sub-matrix. 0 is a column vector of zeros, and $\mathbf{I}$ is an identity matrix.

Lorentz Linear Layer

In Euclidean space:

Linear Layer: $\mathbf{Wx}$

Lorentz Linear Layer

In Euclidean space:

Linear Layer: $\mathbf{W}\mathbf{x}$

In Lorentz Space:

- Tangent space method, $\mathbf{W} \otimes \mathbf{x} := \exp_{\mathbf{o}}^{K}\left(\mathbf{W}\log_{\mathbf{o}}^{K}(\mathbf{x})\right)$

In Euclidean space:

Linear Layer: $\mathbf{W}\mathbf{x}$

In Lorentz Space:

- Tangent space method, $\mathbf{W} \otimes \mathbf{x} := \exp_{\mathbf{o}}^{K}\left(\mathbf{W}\log_{\mathbf{o}}^{K}(\mathbf{x})\right)$
- Manifold-based method,

$$\mathbf{W} \otimes \mathbf{x} := f_{\mathbf{x}}(\mathbf{M})\mathbf{x} = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v}^{\top} \\ \mathbf{W} \end{bmatrix}\right)\mathbf{x} = \begin{bmatrix} \sqrt{\|\mathbf{W}\mathbf{x}\|^2 - 1/K} \\ \mathbf{W}\mathbf{x} \end{bmatrix}$$

Theorem 1. $\forall \mathbf{x} \in \mathbb{L}_K^n, \forall \mathbf{M} \in \mathbb{R}^{(m+1)\times(n+1)}$, we have $f_{\mathbf{x}}(\mathbf{M})\mathbf{x} \in \mathbb{L}_K^m$.

In Euclidean space:

Linear Layer: $\mathbf{Wx}$

In Lorentz Space:

- Tangent space method, $\mathbf{W} \otimes \mathbf{x} := \exp_{\mathbf{o}}^{K}\left(\mathbf{W} \log_{\mathbf{o}}^{K}(\mathbf{x})\right)$
- Manifold-based method,

$$\mathbf{W} \otimes \mathbf{x} := f_{\mathbf{x}}(\mathbf{M})\mathbf{x} = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v}^{\top} \\ \mathbf{W} \end{bmatrix}\right)\mathbf{x} = \begin{bmatrix} \sqrt{\|\mathbf{Wx}\|^2 - 1/K} \\ \mathbf{Wx} \end{bmatrix}$$

Theorem 1. $\forall \mathbf{x} \in \mathbb{L}_K^n, \forall \mathbf{M} \in \mathbb{R}^{(m+1)\times(n+1)}$, we have $f_{\mathbf{x}}(\mathbf{M})\mathbf{x} \in \mathbb{L}_K^m$.

Proof 1. One can easily verify that $\forall \mathbf{x} \in \mathbb{L}_K^n$, we have $\langle f_{\mathbf{x}}(\mathbf{M})\mathbf{x}, f_{\mathbf{x}}(\mathbf{M})\mathbf{x}\rangle_{\mathcal{L}} = 1/K$, thus $f_{\mathbf{x}}(\mathbf{M})\mathbf{x} \in \mathbb{L}_K^m$

Lorentz Transformation

$$f_{\mathbf{x}}(\mathbf{M})\mathbf{x} = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v}^\top \\ \mathbf{W} \end{bmatrix}\right)\mathbf{x} = \begin{bmatrix} \sqrt{\|\mathbf{W}\mathbf{x}\|^2 - 1/K} \\ \mathbf{W}\mathbf{x} \end{bmatrix}$$

Lorentz Transformation

$$f_{\mathbf{x}}(\mathbf{M})\mathbf{x} = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v}^\top \\ \mathbf{W} \end{bmatrix}\right)\mathbf{x} = \begin{bmatrix} \sqrt{\|\mathbf{W}\mathbf{x}\|^2 - 1/K} \\ \mathbf{W}\mathbf{x} \end{bmatrix}$$

$$f_{\mathbf{x}}(\mathbf{M}) = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v}^\top \\ \mathbf{W} \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{\|\mathbf{W}\mathbf{x}\|^2 - 1/K}}{\mathbf{v}^\top \mathbf{x}} \mathbf{v}^\top \\ \mathbf{W} \end{bmatrix}$$

$$f_{\mathbf{x}}(\mathbf{M})\mathbf{x} = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v}^\top \\ \mathbf{W} \end{bmatrix}\right)\mathbf{x} = \begin{bmatrix} \sqrt{\|\mathbf{W}\mathbf{x}\|^2 - 1/K} \\ \mathbf{W}\mathbf{x} \end{bmatrix}$$

$$f_{\mathbf{x}}(\mathbf{M}) = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v}^\top \\ \mathbf{W} \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{\|\mathbf{W}\mathbf{x}\|^2 - 1/K}}{\mathbf{v}^\top \mathbf{x}}\mathbf{v}^\top \\ \mathbf{W} \end{bmatrix}$$

Lemma 1. In the n-dimensional Lorentz model $\mathbb{L}_K^n$, we denote the set of all Lorentz boost matrices as $\mathcal{B}$, the set of all Lorentz rotation matrices as $\mathcal{R}$. Given $\mathbf{x} \in \mathbb{L}_K^n$, we denote the set of $f_{\mathbf{x}}(\mathbf{M})$ at $\mathbf{x}$ without changing the number of space dimension as $\mathcal{M}_{\mathbf{x}} = \left\{ f_{\mathbf{x}}(\mathbf{M}) \mid \mathbf{M} \in \mathbb{R}^{(n+1)\times(n+1)} \right\}.\forall \mathbf{x} \in \mathbb{L}_K^n$, we have $\mathcal{B} \subseteq \mathcal{M}_{\mathbf{x}}$ and $\mathcal{R} \subseteq \mathcal{M}_{\mathbf{x}}$

Proof

We first prove $\mathcal{M}_{\mathrm{x}}$ covers all valid transformations.

Proof

We first prove $\mathcal{M}_{\mathrm{x}}$ covers all valid transformations. Considering $\mathcal{A} = \{\mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)} \mid \forall \mathbf{x} \in \mathbb{L}_K^n : \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x}\rangle_{\mathcal{L}} = \frac{1}{K}, (\mathbf{A}\mathbf{x})_0 > 0\}$ is the set of all valid transformation matrices in the Lorentz model.

Proof

We first prove $\mathcal{M}_{\mathrm{x}}$ covers all valid transformations. Considering $\mathcal{A} = \left\{\mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)} \mid \forall \mathbf{x} \in \mathbb{L}_K^n : \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x}\rangle_{\mathcal{L}} = \frac{1}{K}, (\mathbf{A}\mathbf{x})_0 > 0\right\}$ is the set of all valid transformation matrices in the Lorentz model. Then $\forall \mathbf{A} = \begin{bmatrix} \mathbf{v}_A^\top \\ \mathbf{w}_A \end{bmatrix} \in \mathcal{A}, \mathbf{v}_A \in \mathbb{R}^{n+1}, \mathbf{W}_A \in \mathbb{R}^{n\times(n+1)}, \exists \mathbf{x} \in \mathbb{R}^{n+1} : \mathbf{v}^\top \mathbf{x} > 0$ and $\|\mathbf{W}_A \mathbf{x}\|^2 - \left(\mathbf{v}_A^\top \mathbf{x}\right)^2 = \frac{1}{K}$.

Proof

We first prove $\mathcal{M}_{\mathrm{x}}$ covers all valid transformations. Considering $\mathcal{A} = \left\{\mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)} \mid \forall \mathbf{x} \in \mathbb{L}_K^n : \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x}\rangle_{\mathcal{L}} = \frac{1}{K}, (\mathbf{A}\mathbf{x})_0 > 0\right\}$ is the set of all valid transformation matrices in the Lorentz model. Then $\forall \mathbf{A} = \begin{bmatrix} \mathbf{v}_A^\top \\ \mathbf{W}_A \end{bmatrix} \in \mathcal{A}, \mathbf{v}_A \in \mathbb{R}^{n+1}, \mathbf{W}_A \in \mathbb{R}^{n\times(n+1)}, \exists \mathbf{x} \in \mathbb{R}^{n+1} : \mathbf{v}^\top \mathbf{x} > 0$ and $\|\mathbf{W}_A\mathbf{x}\|^2 - \left(\mathbf{v}_A^\top \mathbf{x}\right)^2 = \frac{1}{K}$.
Furthermore, $\forall \mathbf{A} \in \mathcal{A}$, we have

$$f_{\mathbf{x}}(\mathbf{A}) = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v_A}^\top \\ \mathbf{W_A} \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{\|\mathbf{W_A}\mathbf{x}\|^2 - 1/K}}{\mathbf{v}_\mathbf{A}^\top \mathbf{x}} \mathbf{v}_\mathbf{A}^\top \\ \mathbf{W_A} \end{bmatrix} = \mathbf{A}$$

Hence, we can see that $\mathcal{A} \subseteq \mathcal{M}_{\mathrm{x}}$. Since $\mathcal{B} \subseteq \mathcal{A}$ and $\mathcal{R} \subseteq \mathcal{A}$, therefore $\mathcal{B} \subseteq \mathcal{M}_{\mathbf{x}}$ and $\mathcal{R} \subseteq \mathcal{M}_{\mathbf{x}}$.

Proof

We first prove $\mathcal{M}_{\mathrm{x}}$ covers all valid transformations. Considering $\mathcal{A} = \left\{ \mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)} \mid \forall \mathbf{x} \in \mathbb{L}_K^n : \langle \mathbf{A}\mathbf{x}, \mathbf{A}\mathbf{x} \rangle_{\mathcal{L}} = \frac{1}{K}, (\mathbf{A}\mathbf{x})_0 > 0 \right\}$ is the set of all valid transformation matrices in the Lorentz model. Then $\forall \mathbf{A} = \begin{bmatrix} \mathbf{v}_A^\top \\ \mathbf{w}_A \end{bmatrix} \in \mathcal{A}, \mathbf{v}_A \in \mathbb{R}^{n+1}, \mathbf{W}_A \in \mathbb{R}^{n\times(n+1)}, \exists \mathbf{x} \in \mathbb{R}^{n+1} : \mathbf{v}^\top \mathbf{x} > 0$ and $\|\mathbf{W}_A \mathbf{x}\|^2 - \left(\mathbf{v}_A^\top \mathbf{x}\right)^2 = \frac{1}{K}$. Furthermore, $\forall \mathbf{A} \in \mathcal{A}$, we have

$$f_{\mathbf{x}}(\mathbf{A}) = f_{\mathbf{x}}\left(\begin{bmatrix} \mathbf{v_A}^\top \\ \mathbf{W_A} \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{\|\mathbf{W_A}\mathbf{x}\|^2 - 1/K}}{\mathbf{v_A}^\top \mathbf{x}} \mathbf{v_A}^\top \\ \mathbf{W_A} \end{bmatrix} = \mathbf{A}$$

Hence, we can see that $\mathcal{A} \subseteq \mathcal{M}_{\mathrm{x}}$. Since $\mathcal{B} \subseteq \mathcal{A}$ and $\mathcal{R} \subseteq \mathcal{A}$, therefore $\mathcal{B} \subseteq \mathcal{M}_{\mathbf{x}}$ and $\mathcal{R} \subseteq \mathcal{M}_{\mathbf{x}}$.

According to Theorem 1 and Lemma 1, both Lorentz boost and rotation can be covered by the proposed linear layer.

Tangent Method

$$\exp_{\mathbf{0}}\left(\begin{bmatrix} * & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{W} \end{bmatrix} \log_{\mathbf{0}}\left(\begin{bmatrix} x_t \\ \mathbf{x}_s \end{bmatrix}\right)\right)$$

Tangent Method

$$\exp_{\mathbf{0}}\left(\begin{bmatrix} * & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{W} \end{bmatrix} \log_{\mathbf{0}}\left(\begin{bmatrix} x_t \\ \mathbf{x}_s \end{bmatrix}\right)\right) = \begin{bmatrix} \frac{\cosh(\beta)}{\sqrt{-K}x_t} & \mathbf{0}^\top \\ \mathbf{0} & \frac{\sinh(\beta)\mathbf{w}}{\sqrt{-K}\|\mathbf{W}\mathbf{x}_s\|} \end{bmatrix} \begin{bmatrix} x_t \\ \mathbf{x}_s \end{bmatrix}$$

$$\text{where } \beta = \frac{\sqrt{-K}\cosh^{-1}\left(\sqrt{-K}x_t\right)}{\sqrt{-Kx_t^2 - 1}} \|\mathbf{W}\mathbf{x}_s\|.$$

Tangent Method

$$\exp_{\mathbf{0}}\left(\begin{bmatrix} * & \mathbf{0}^\top \\ 0 & \mathbf{W} \end{bmatrix} \log_{\mathbf{0}}\left(\begin{bmatrix} x_t \\ \mathbf{x}_s \end{bmatrix}\right)\right) = \begin{bmatrix} \frac{\cosh(\beta)}{\sqrt{-K}x_t} & \mathbf{0}^\top \\ \mathbf{0} & \frac{\sinh(\beta)\mathbf{w}}{\sqrt{-K}\|\mathbf{W}\mathbf{x}_s\|} \end{bmatrix} \begin{bmatrix} x_t \\ \mathbf{x}_s \end{bmatrix}$$

$$\text{where } \beta = \frac{\sqrt{-K}\cosh^{-1}\left(\sqrt{-K}x_t\right)}{\sqrt{-Kx_t^2 - 1}}\|\mathbf{W}\mathbf{x}_s\|.$$

Lemma 2. $\forall \mathbf{x} \in \mathbb{L}_K^n$, we define the set of the outcomes of Eq.(2) as

$$\mathcal{H}_{\mathbf{x}} = \left\{ \begin{bmatrix} \frac{\cosh(\beta)}{\sqrt{-K}x_t} & \mathbf{0}^\top \\ \mathbf{0} & \frac{\sinh(\beta)}{\sqrt{-K}\|\mathbf{W}_{\mathbf{x}_s}\|}\mathbf{W} \end{bmatrix} \mid \mathbf{W} \in \mathbb{R}^{n\times n} \right\}$$

then we have $\mathcal{H}_{\mathbf{x}} \subseteq \mathcal{P}_{\mathbf{x}}$ and $\mathcal{H}_{\mathbf{x}} \cap \mathcal{B} = \{\mathbf{I}\}$

Connection with Tangent Method

Formally, at the point $\mathbf{x} \in \mathbb{L}_K^n$, all pseudo-rotation matrices make up the set $\mathcal{P}_{\mathbf{x}} = \left\{ f_{\mathbf{x}}\left(\begin{bmatrix} w & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{W} \end{bmatrix} \right) \mid w \in \mathbb{R}, \mathbf{W} \in \mathbb{R}^{n \times n} \right\}$. As we no longer require the submatrix $\mathbf{W}$ to be a special orthogonal matrix, this setting is a relaxation of the Lorentz rotation.

Formally, at the point $\mathbf{x} \in \mathbb{L}_K^n$, all pseudo-rotation matrices make up the set $\mathcal{P}_{\mathbf{x}} = \left\{ f_{\mathbf{x}}\left(\begin{bmatrix} w & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{W} \end{bmatrix} \right) \mid w \in \mathbb{R}, \mathbf{W} \in \mathbb{R}^{n \times n} \right\}$. As we no longer require the submatrix $\mathbf{W}$ to be a special orthogonal matrix, this setting is a relaxation of the Lorentz rotation.

Therefore, a conventional hyperbolic linear layer can be considered as a special rotation where the time axis is changed according to the space axes to ensure that the output is still in the Lorentz model

General Form of Linear Layer

A More General Formula Here, we give a more general formula of the above hyperbolic linear layer, by adding activation, dropout, bias, and normalization,

General Form of Linear Layer

A More General Formula Here, we give a more general formula of the above hyperbolic linear layer, by adding activation, dropout, bias, and normalization,

$$\mathbf{y} = \mathrm{HL}(\mathbf{x}) = \begin{bmatrix} \sqrt{\|\phi(\mathbf{W}\mathbf{x}, \mathbf{v})\|^2 - 1/K} \\ \phi(\mathbf{W}\mathbf{x}, \mathbf{v}) \end{bmatrix}$$

General Form of Linear Layer

A More General Formula Here, we give a more general formula of the above hyperbolic linear layer, by adding activation, dropout, bias, and normalization,

$$\mathbf{y} = \mathrm{HL}(\mathbf{x}) = \begin{bmatrix} \sqrt{\|\phi(\mathbf{Wx}, \mathbf{v})\|^2 - 1/K} \\ \phi(\mathbf{Wx}, \mathbf{v}) \end{bmatrix}$$

where $\mathbf{x} \in \mathbb{L}_K^n, \mathbf{v} \in \mathbb{R}^{n+1}, \mathbf{W} \in \mathbb{R}^{m\times(n+1)}$, and ϕ is an operation function: for the dropout, the function is $\phi(\mathbf{Wx}, \mathbf{v}) = \mathbf{W}$ dropout $(\mathbf{x})$; for the activation and normalization $\phi(\mathbf{Wx}, \mathbf{v}) =$

$$\frac{\lambda\sigma\left(\mathbf{v}^\top \mathbf{x} + b'\right)}{\|\mathbf{W}h(\mathbf{x}) + \mathbf{b}\|}(\mathbf{W}h(\mathbf{x}) + \mathbf{b}),$$

where σ is the sigmoid function, $\mathbf{b}$ and b' are bias terms, $\lambda > 0$ controls the scaling range, h is the activation function. We elaborate $\phi(\cdot)$ we use in practice in the appendix.

Lorentz Attention Layer

Specifically, we consider the weighted aggregation of a point set $\mathcal{P} = \left\{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{P}|}\right\}$ as calculating the centroid, whose expected (squared) distance to $\mathcal{P}$ is minimum, i.e., $\arg\min_{\boldsymbol{\mu} \in \mathbb{L}_K^n} \sum_{i=1}^{|\mathcal{P}|} \nu_i d_{\mathcal{L}}^2 (\mathbf{x}_i, \boldsymbol{\mu})$, where ν_i is the weight of the i-th point.

Specifically, we consider the weighted aggregation of a point set $\mathcal{P} = \left\{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{P}|}\right\}$ as calculating the centroid, whose expected (squared) distance to $\mathcal{P}$ is minimum, i.e., $\arg\min_{\boldsymbol{\mu}\in\mathbb{L}_K^n} \sum_{i=1}^{|\mathcal{P}|} \nu_i d_{\mathcal{L}}^2\left(\mathbf{x}_i, \boldsymbol{\mu}\right)$, where ν_i is the weight of the i-th point.

Law et al. (2019) prove that, with squared Lorentzian distance defined as $d_{\mathcal{L}}^2(\mathbf{a}, \mathbf{b}) = 2/K - 2\langle\mathbf{a}, \mathbf{b}\rangle_{\mathcal{L}}$, the centroid w.r.t. the squared Lorentzian distance is given as

Specifically, we consider the weighted aggregation of a point set $\mathcal{P} = \left\{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{P}|}\right\}$ as calculating the centroid, whose expected (squared) distance to $\mathcal{P}$ is minimum, i.e., $\arg\min_{\boldsymbol{\mu} \in \mathbb{L}_K^n} \sum_{i=1}^{|\mathcal{P}|} \nu_i d_{\mathcal{L}}^2\left(\mathbf{x}_i, \boldsymbol{\mu}\right)$, where ν_i is the weight of the i-th point.

Law et al. (2019) prove that, with squared Lorentzian distance defined as $d_{\mathcal{L}}^2(\mathbf{a}, \mathbf{b}) = 2/K - 2\langle \mathbf{a}, \mathbf{b}\rangle_{\mathcal{L}}$, the centroid w.r.t. the squared Lorentzian distance is given as

$$\begin{aligned}\boldsymbol{\mu} &= \operatorname{Centroid}\left(\left\{\nu_1, \ldots, \nu_{|\mathcal{P}|}\right\}, \left\{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{P}|}\right\}\right) \\ &= \frac{\sum_{j=1}^{|\mathcal{P}|} \nu_j \mathbf{x}_j}{\sqrt{-K} \mid \left\|\sum_{i=1}^{|\mathcal{P}|} \nu_i \mathbf{x}_i\right\|_{\mathcal{L}} \mid}\end{aligned}$$

Given the query set $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_{|\mathcal{Q}|}\}$, key set $\mathcal{K} = \{\mathbf{k}_1, \ldots, \mathbf{k}_{|\mathcal{K}|}\}$, and value set $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_{|\mathcal{V}|}\}$, where $|\mathcal{K}| = |\mathcal{V}|$, we exploit the squared Lorentzian distance between points to calculate weights. Attention is defined as

$$\mathrm{ATT}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \left\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|\mathcal{Q}|}\right\}$$

Given the query set $\mathcal{Q} = \left\{\mathbf{q}_1, \ldots, \mathbf{q}_{|\mathcal{Q}|}\right\}$, key set $\mathcal{K} = \left\{\mathbf{k}_1, \ldots, \mathbf{k}_{|\mathcal{K}|}\right\}$, and value set $\mathcal{V} = \left\{\mathbf{v}_1, \ldots, \mathbf{v}_{|\mathcal{V}|}\right\}$, where $|\mathcal{K}| = |\mathcal{V}|$, we exploit the squared Lorentzian distance between points to calculate weights. Attention is defined as

$$\mathrm{ATT}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \left\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|\mathcal{Q}|}\right\}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{|\mathcal{K}|} \nu_{ij} \mathbf{v}_j}{\sqrt{-K} \left| \left\| \sum_{k=1}^{|\mathcal{K}|} \nu_{ik} \mathbf{v}_k \right\|_{\mathcal{L}} \right|},$$

$$\nu_{ij} = \frac{\exp\left(\frac{-d_{\mathcal{L}}^2(\mathbf{q}_i, \mathbf{k}_j)}{\sqrt{n}}\right)}{\sum_{k=1}^{|\mathcal{K}|} \exp\left(\frac{-d_{\mathcal{L}}^2(\mathbf{q}_i, \mathbf{k}_k)}{\sqrt{n}}\right)},$$

where n is the dimension of points.

Setup Similar to Balazevic et al. (2019a), they design a score function for each triplet as

$$s(h, r, t) = -d_{\mathcal{L}}^2\left(f_r\left(\mathbf{e}_h\right), \mathbf{e}_t\right) + b_h + b_t + \delta$$

where $\mathbf{e}_h, \mathbf{e}_t \in \mathbb{L}_K^n$ are the Lorentz embeddings of the head entity h and the tail entity t, $f_r(\cdot)$ is a Lorentz linear transformation of the relation r and δ is a margin hyper-parameter.

Setup Similar to Balazevic et al. (2019a), they design a score function for each triplet as

$$s(h, r, t) = -d_{\mathcal{L}}^2 \left(f_r\left(\mathbf{e}_h\right), \mathbf{e}_t\right) + b_h + b_t + \delta$$

where $\mathbf{e}_h, \mathbf{e}_t \in \mathbb{L}_K^n$ are the Lorentz embeddings of the head entity h and the tail entity t, $f_r(\cdot)$ is a Lorentz linear transformation of the relation r and δ is a margin hyper-parameter. For each triplet, they randomly corrupt its head or tail entity with k entities and calculate the probabilities for triplets as $p = \sigma(s(h, r, t))$, where σ is the sigmoid function. Finally, they minimize the binary cross-entropy loss

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left(\log p^{(i)} + \sum_{j=1}^{k} \log \left(1 - \tilde{p}^{(i,j)}\right) \right)$$

where $p^{(i)}$ and $\tilde{p}^{(i,j)}$ are the probabilities for correct and corrupted triplets respectively, N is the triplet number.

Model	WN18RR #Dims	MRR	H@10	H@3	H@1	FB15k-237 #Dims	MRR	H@10	H@3	H@1
TRANSE (Bordes et al., 2013)	180	22.7	50.6	38.6	3.5	200	28.0	48.0	32.1	17.7
DISTMULT (Yang et al., 2015)	270	41.5	48.5	43.0	38.1	200	19.3	35.3	20.8	11.5
COMPLEX (Trouillon et al., 2017)	230	43.2	50.0	45.2	39.6	200	25.7	44.3	29.3	16.5
CONVE (Dettmers et al., 2018)	120	43.5	50.0	44.6	40.1	200	30.4	49.0	33.5	21.3
ROTATE (Sun et al., 2019)	1,000	47.3	55.3	48.8	43.2	1,024	30.1	48.5	33.1	21.0
TUCKER (Balazevic et al., 2019b)	200	46.1	53.5	47.8	42.3	200	34.7	53.3	38.4	25.4
MURP (Balazevic et al., 2019a)	32	46.5	54.4	48.4	42.0	32	32.3	50.1	35.3	23.5
ROTH (Chami et al., 2020a)	32	47.2	<u>55.3</u>	49.0	42.8	32	31.4	49.7	34.6	22.3
ATTH (Chami et al., 2020a)	32	46.6	55.1	48.4	41.9	32	32.4	50.1	35.4	23.6
HYBONET	32	<u>48.9</u>	<u>55.3</u>	<u>50.3</u>	<u>45.5</u>	32	<u>33.4</u>	<u>51.6</u>	<u>36.5</u>	<u>24.4</u>
MURP (Balazevic et al., 2019a)	β	48.1	56.6	49.5	44.0	β	33.5	51.8	36.7	24.3
ROTH (Chami et al., 2020a)	β	49.6	<u>**58.6**</u>	51.4	44.9	β	34.4	53.5	38.0	24.6
ATTH (Chami et al., 2020a)	β	48.6	57.3	49.9	44.3	β	34.8	<u>**54.0**</u>	38.4	25.2
HYBONET	β	<u>**51.3**</u>	56.9	<u>**52.7**</u>	<u>**48.2**</u>	β	<u>**35.2**</u>	52.9	<u>**38.7**</u>	<u>**26.3**</u>

Task2: Machine Translation

"We use OpenNMT (Klein et al., 2017) to build Euclidean Transformer and our Lorentz one. Following previous hyperbolic work (Shimizu et al., 2021), "We use OpenNMT (Klein et al., 2017) to build Euclidean Transformer and our Lorentz one. Following previous hyperbolic work (Shimizu et al., 2021), we conduct experiments in lowdimensional settings. To show that our framework can be applied to high-dimensional settings, we additionally train a Lorentz Transformer of the same size as Transformer base, and compare their performance on WMT'14. "

	IWSLT'14	WMT'14		
Model	d=64	d=64	d=128	d=256
CONVSEQ2SEQ	23.6	14.9	20.0	21.8
TRANSFORMER	23.0	17.0	21.7	25.1
HYPERNN++	22.0	17.0	19.4	21.8
HATT	23.7	18.8	22.5	25.5
HYBONET	**25.9**	**19.7**	**23.3**	**26.2**

Summary

Main contents:

- Lorentz Model
- Tangent space Linear Transformation
- Fully Linear Transformation
- KG Compilation
- Machine Translation

Strengths and weakness:

- Strengths
- Weakness

[CYRL19] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec.
Hyperbolic graph convolutional neural networks.
Advances in neural information processing systems, 32, 2019.

[DWGJ21] Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia.
A hyperbolic-to-hyperbolic graph convolutional network.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 154–163, 2021.

[GBH18] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann.
Hyperbolic neural networks.
Advances in neural information processing systems, 31, 2018.

[GSGR18] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré.
Learning mixed-curvature representations in product spaces.
In *International conference on learning representations*, 2018.

[KPK+10] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná.
Hyperbolic geometry of complex networks.
Physical Review E, 82(3):036106, 2010.

[NK17] Maximillian Nickel and Douwe Kiela.
Poincaré embeddings for learning hierarchical representations.
Advances in neural information processing systems, 30, 2017.

[NK18] Maximillian 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.

[SNW+21] Atsushi Suzuki, Atsushi Nitanda, Jing Wang, Linchuan Xu, Kenji Yamanishi, and Marc Cavazza.
Generalization error bound for hyperbolic ordinal embedding.
In *International Conference on Machine Learning*, pages 10011–10021. PMLR, 2021.

[SWT+19] Atsushi Suzuki, Jing Wang, Feng Tian, Atsushi Nitanda, and Kenji Yamanishi.
Hyperbolic ordinal embedding.
In *Asian Conference on Machine Learning*, pages 1065–1080. PMLR, 2019.

Thanks!