

1 COMPUTATIONAL COSTS OF THE HCTS FRAMEWORK

1.1 Time complexity of graph neural network-based baselines

The followings are the notations used for analyzing time complexity.

- Interactions in the source domain: $|E^S|$
- Interactions in the target domain: $|E^T|$
- Number of nodes in the source domain: $|S|$
- Number of nodes in the target domain: $|T|$
- Number of overlapped users: $|U^O|$
- Number of sampled users for contrastive learning: B_1
- Number of sampled items for contrastive learning: B_2
- Number of negative samples for contrastive learning: N
- Number of layers: L
- Latent dimension: d
- Users in testing set: U_T
- Items in testing set: I_T

Time complexity for each method:

- **LightGCN:**
 - Graph Encoding on target domain: $O(|E^T|d)$
 - Inference : $O(|U_T||I_T|d)$
- **GCF:**
 - Graph Encoding on target domain: $O(|E^T|d)$
 - Inference : $O(|U_T||I_T|d)$
- **HGCF:**
 - Exponential map: $O(|E^T|d)$
 - Graph Encoding on target domain: $O(|E^T|d)$
 - Inference : $O(|U_T||I_T|d)$
- **BiTGCF:**
 - Graph Encoding on target domain: $O(|E^T|d)$
 - Graph Encoding on source domain: $O(|E^S|d)$
 - Knowledge transfer: $O(|U^O|L)$
 - Inference : $O(|U_T||I_T|d)$
- **HCTS (ours):**
 - Exponential map: $O((|T| + |S|)d)$
 - Graph Encoding on source domain: $O(|E^S|d)$
 - Graph Encoding on target domain: $O(|E^T|d)$
 - Knowledge transfer:
 - * Manifold alignment: $O((|T| + |S|)d)$
 - * User-user contrastive learning: $O(|U^O|d)$
 - * User-item contrastive learning: $O(B_1dN)$
 - * Item-item contrastive learning: $O(B_2dN)$
 - Inference : $O(|U_T||I_T|d)$

The time complexity of HCTS primarily lies in Contrastive learning, but since the main step in calculating hyperbolic distance involves computing the Minkowski inner product, contrastive learning in hyperbolic space is similar to the operation of calculating cosine similarity in Euclidean space.





















1.2 Training and inference time cost for each epoch

We also calculated the average time cost per epoch for training/inference for each model on the Douban Book-Movie experiments and the results are in Table 1. According to the results, it can be seen that due to the incorporation of contrastive learning, HCTS has a slightly longer training time per epoch compared to other methods and is closely aligning with the training time of BiTGCF. In terms of inference time, all models are similar; however, since inference in hyperbolic space is based on hyperbolic distance, which makes HGCF and HCTS take marginally longer time than other models.

2 EXPERIMENTS ON LARGE DATASET

To validate the scalability of our model, we selected a larger dataset for experiments. Table 3 shows the detailed statistics of the datasets and Table 2 shows the experimental results. It is evident that our model also achieves best results in the context of large datasets. The results

Table 1: Time cost per epoch

Method	Training time(s)	Visualization	Inference time(s)	Visualization
HGCF	1.41		0.84	
LightGCN	0.94		0.45	
GCF	0.99		0.45	
BiTGCF	4.36		0.45	
CoNET	2.06		0.57	
DTCDR	0.66		0.62	
CMF	0.49		0.42	
DeepAPF	0.51		0.70	
CLFM	0.56		0.43	
HCTS	4.47		0.86	

below show that HCTS also outperforms other models on large-scale datasets. CDDR, being a GAT-based model, underperform in most cases on smaller datasets, as is shown in Table ??, however, on larger datasets, its results are surpassed only by HCTS.

Table 2: Comparison of model performance on the Amazon large (Movie-Toy) dataset.

Model	NDCG@10	Hit@10
HGCF	0.0268	0.0674
LightGCN	0.0116	0.0368
GCF	0.0112	0.0256
BiTGCF	0.0358	0.0636
CoNet	0.0177	0.0544
DTCDR	0.0254	0.0659
CMF	0.0199	0.0401
DeepAPF	0.0257	0.0656
CLFM	0.0199	0.0401
EMCDR	0.0107	0.0262
ART-CAT	0.0329	0.0648
CDDR	0.0368	0.0685
HCTS (ours)	0.0372	0.0706

Table 3: Dataset information and experiment results

	Amazon movie (Large)	Amazon toy (Large)
Users	105027	15529
Items	44211	9697
Interactions	1406666	133837
Overlapping scale of users: 16.70%		

3 DEFINITIONS OF EVALUATION METRICS

We use $\hat{R}(u)$ to represent a ranked list of item that model produces and $R(u)$ to represent a ground truth set of items that user u interacted with. For the @10 index, the length of $\hat{R}(u)$ is 10. If there is at least one item that falls in the ground-truth set, we call it a hit.

The Hit Rate at 10 (HR@10) is defined as:

$$\text{HR@10} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \delta(\hat{R}(u) \cap R(u) \neq \emptyset).$$

where $\delta(\cdot)$ is an indicator function such that $\delta(b) = 1$ if b is true and 0 otherwise. \emptyset denotes the empty set.

The Normalized Discounted Cumulative Gain at K (NDCG@K) is defined as:

$$\text{NDCG@10} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \left(\frac{1}{\sum_{i=1}^{\min(|R(u)|, 10)} \frac{1}{\log_2(i+1)}} \sum_{i=1}^{10} \frac{\delta(i \in R(u))}{\log_2(i+1)} \right),$$

where $\delta(\cdot)$ is an indicator function.

4 IN-DEPTH ANALYSIS OF THE CURVATURE

4.1 Theoretical analysis

The curvature of hyperbolic space is closely related to the final performance of hyperbolic neural networks. The performance of hyperbolic models is related to embedding distortion, which is a metric that measures how accurate embeddings can represent the original data. Precisely, it is formally defined in [1, 4] as follows:

Definition 4.1 (Embedding Distortion). Let (X, d_X) and (Y, d_Y) be any metric spaces. A function $f : X \rightarrow Y$ is an embedding with distortion c for $c \geq 1$ if

$$d_Y(f(u), f(v)) \leq d_X(u, v) \leq cd_Y(f(u), f(v)),$$

for all $u, v \in X$.

When a graph has a hierarchical structure, the number of its nodes increases exponentially from the center. In this case, the distortion in hyperbolic space is smaller than in Euclidean space because the volume in hyperbolic space increases exponentially, whereas the volume in Euclidean space increases polynomially.

Let $\frac{1}{K}$ be the curvature of a d -dimensional hyperbolic space, then the volume of a ball in this space is:

$$V_K(r) = G_{d-1} \int_0^r \left(\sqrt{K} \sinh\left(\frac{1}{\sqrt{K}}t\right) \right)^{d-1} dt,$$

where $G_{n-1} := \frac{2\pi^{n/2}}{\Gamma(n/2)}$ is the $n-1$ dimensional area of a unit sphere in R^n .

The larger the curvature $\frac{1}{K}$, the faster the volume of the hyperbolic space grows. Therefore, for graphs where the nodes increase more rapidly, the required curvature of the hyperbolic space should be larger. The relationship between embedding distortion and curvature is shown in the proposition as follows:

PROPOSITION 1. Consider a star graph S_n with center s_0 and leaves s_1, \dots, s_n . Suppose the weight of each edge is 1 and if there exists an embedding f of $(S_n, \lambda ds)$ into d -dimensional hyperbolic space \mathbb{H}^d with constant distortion c and changeable curvature $\frac{1}{K}$, then $d = \Omega(\sqrt{K} \ln n)$.

PROOF. Without loss of generality, we suppose f is non-expanding, which means

$$d_H(f(x), f(y)) \leq d_S(x, y) \leq cd_H(f(x), f(y)), \quad \forall x, y \in S_n, c \geq 1.$$

Then $d_H(s_0, s_i) \leq 1$ for $i = 1, 2, \dots, n$. If the embedding f has multiplicative distortion c , then the balls in \mathbb{H}^d with radius $\frac{1}{c}$ centered at the points h_i for $i = 1, 2, \dots, n$ could be interior disjoint. All these balls could lie inside the ball with radius 2 centered at h_0 .

The volume of a ball with radius r in H^d is

$$V(d, r) = G_{d-1} \int_0^r \left(\sqrt{K} \sinh\left(\frac{1}{\sqrt{K}}t\right) \right)^{d-1} dt = \Omega\left(G(d)((\sqrt{K})^{d-1}2^{rd})\right)$$

And by changing the variable, $u = \sinh(\frac{1}{\sqrt{K}}t)$, and integration by parts gives

$$\begin{aligned} \int_0^r \left(\sqrt{K} \sinh\left(\frac{1}{\sqrt{K}}t\right) \right)^{d-1} dt &= \left(\sqrt{K} \right)^d \int_0^{\sinh(\frac{1}{\sqrt{K}}r)} u^{d-1} (1+u^2)^{-\frac{1}{2}} du \\ &= \left(\frac{1}{\sqrt{K}} \right)^d \left[\frac{1}{d} \left(\frac{\sinh(\frac{1}{\sqrt{K}}r)^d}{\cosh(\frac{1}{\sqrt{K}}r)} \right) + \int_0^{\sinh(\frac{1}{\sqrt{K}}r)} ru^{d+1} (1+u^2)^{-3/2} du \right] \\ &\leq \frac{\sinh\left(\frac{1}{\sqrt{K}}r\right)^d}{d \left(\frac{1}{\sqrt{K}}\right)^d \cosh\left(\frac{1}{\sqrt{K}}r\right)} \left[1 + \frac{\cosh\left(\frac{1}{\sqrt{K}}r\right) \sinh\left(\frac{1}{\sqrt{K}}r\right)^2}{d+2} \right]. \end{aligned}$$

Then we have

$$V(d, r) = O(G(d) \left(\frac{e^{\frac{1}{\sqrt{K}}(rd+2r)}}{\left(\frac{1}{\sqrt{K}}\right)^d} \right))$$

Since c is constant and n balls with radius $\frac{1}{c}$ must fit in a ball with radius 2, for big enough d , the following inequality holds:

$$nG(d) \frac{2^{\frac{d}{c}}}{\left(\frac{1}{\sqrt{K}}\right)^{d-1}} \leq G(d) \int_0^2 \left(\sqrt{K} \sinh\left(\frac{1}{\sqrt{K}}t\right) \right)^{d-1} dt \leq G(d) e^{\frac{1}{\sqrt{K}}(2d+4)} \left(\frac{1}{\sqrt{K}}\right)^d$$

Then

$$\ln n \leq \frac{1}{\sqrt{K}}(2d + 4) - \frac{d}{c} \ln 2 - \ln \left(\frac{1}{\sqrt{K}} \right)$$

From the above inequality, we can conclude in hyperbolic space, $d = \Omega \left(\sqrt{K} \ln n \right)$.

4.2 Comparative evaluation of diverse curvature settings

To investigate the impact of curvature on the final experimental results, we conducted tests for both fixed curvature and trainable curvature scenarios. Table 7 and Table 8 present the results, where rows correspond to K of the target domain, and columns correspond to K of the source domain. The values at each position indicate the prediction performance under the corresponding K values, evaluated by the metrics NDCG@10 and HR@10.

In HCTS, the value K of both the source domain and the target domain are learnable parameters. In HCTS, varying the initial values, as shown in Table 7, resulted in minimal changes to performance, demonstrating the robustness of our method. Additionally, when setting K of source domain and target domain as fixed values and conducting the same experiments, we observed more significant fluctuations in model performance. Therefore, it is evident that HCTS can approximate the optimal choice more closely compared to fixed curvature settings.

Table 4: HCTS with trainable curvature

Table 5: HCTS with fixed curvature

K(Source/Target)	0.1	0.5	1.0	K(Source/Target)	0.1	0.5	1.0
0.1	0.0468/0.1854	0.0469/0.1863	0.0460/0.1837	0.1	0.0369/0.1726	0.0382/0.1782	0.0298/0.1626
0.5	0.0460/0.1837	0.0463/0.1854	0.0467/0.1854	0.5	0.0316/0.1763	0.0392/0.1754	0.0368/0.1728
1.0	0.0467/0.1854	0.0472/0.1896	0.0474/0.1898	1.0	0.0423/0.1792	0.0368/0.1728	0.0428/0.1792

5 ADDITIONAL EXPERIMENTS

In addition to the experiments presented in the Table ??, we conducted extra experiments to verify the performance of our model on datasets with weaker correlations. In the experimental results, we find that although the improvement is less significant than experiments in Table ??, our model also outperforms the baseline models such as Music→Book (+6.79% on NDCG@10, +4.76% on HIT@10) and Toy→Music (+8.19% on NDCG@10, +3.35% on HIT@10).

Table 6: Extra experiments: . The performance of each model is evaluated by the metrics ndcg@10 and hit@10, denoted in this table as N@10 and H@10. * represents the significance level p -value < 0.05 . The best-performing model for every dataset and metric is emphasized in bold, while the second-best is underlined. Improvement in the last line means the relative improvement compared to the second-best model.

Model	Amazon											
	Music→Movie		Music→Book		Toy→Book		Music→Toy		Book→Toy		Toy→Music	
	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10
HGCF	<u>0.0347</u>	<u>0.0942</u>	<u>0.0280</u>	<u>0.1093</u>	<u>0.0280</u>	<u>0.1093</u>	0.0253	0.0568	0.0253	0.0568	<u>0.0488</u>	<u>0.1163</u>
LightGCN	0.0230	0.0653	0.0186	0.0858	0.0186	0.0858	0.0297	0.0563	0.0297	0.0563	0.0391	0.0959
GCF	0.0232	0.0681	0.0172	0.0746	0.0172	0.0746	0.0293	0.0572	0.0293	0.0572	0.0409	0.1008
BiTGCF	0.0280	0.0808	0.0203	0.0874	0.0193	0.0910	<u>0.0297</u>	<u>0.0605</u>	0.0290	0.0563	0.0386	0.1056
CoNet	0.0121	0.0388	0.0066	0.0290	0.0071	0.0307	<u>0.0097</u>	0.0237	0.0089	0.0249	0.0129	0.0349
DTCDR	0.0108	0.0322	0.0077	0.0383	0.0084	0.0362	0.0136	0.0317	0.0156	0.0335	0.0163	0.0484
CMF	0.0267	0.0735	0.0182	0.0833	0.0190	0.0864	0.0233	0.0483	0.0301	0.0605	0.0395	0.1095
DeepAPF	0.0194	0.0607	0.0169	0.0762	0.0138	0.0577	0.0286	0.0596	<u>0.0307</u>	<u>0.0636</u>	0.0310	0.0843
CLFM	0.0139	0.0464	0.0088	0.0481	0.0105	0.0488	0.0147	0.0345	0.0161	0.0331	0.0194	0.0572
HCTS	0.0358*	0.0952*	0.0299*	0.1145*	0.0290*	0.1131*	0.0304*	0.0619*	0.032*	0.0654*	0.0528*	0.1202*
Improvement	+3.17%	+1.06%	+6.79%	+4.76%	+3.57%	+3.48%	+2.36%	+2.31%	+4.23%	+2.83%	+8.19%	+3.35%

6 HYPERBOLIC GEOMETRY

In this appendix, we present details about the hyperbolic concepts mentioned in this work.

Minkowski (pseudo) inner product. Consider the bilinear map $\langle \cdot, \cdot \rangle_{\mathcal{M}} : \mathbb{R}^{n+1} \times \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ defined by

$$\langle u, v \rangle_{\mathcal{M}} := -u_0 v_0 + \sum_{i=1}^n u_i v_i = u^T J v,$$

where $J = \text{diag}(-1, 1, \dots, 1) \in \mathbb{R}^{(n+1) \times (n+1)}$. It is called Minkowski (pseudo) inner product on \mathbb{R}^{n+1} . This is not an inner product on \mathbb{R}^{n+1} because J has a negative eigenvalue, but it is a pseudo-inner product because all eigenvalues of J are nonzero. Given a constant $K > 0$, the equality $\langle x, x \rangle_{\mathcal{M}} = -K$ implies that

$$x_0^2 = K + \sum_{i=1}^n x_i^2 \geq K.$$

Hyperbolic manifold $\mathcal{H}^{n,K}$ and Tangent space $T_x \mathcal{H}^{n,K}$. Consider the subset of \mathbb{R}^{n+1} defined as follows:

$$\begin{aligned} \mathcal{H}^{n,K} &:= \{x \in \mathbb{R}^{n+1} : \langle x, x \rangle_{\mathcal{M}} = -K \text{ and } x_0 > 0\} \\ &= \{x \in \mathbb{R}^{n+1} : x_0^2 = K + x_1^2 + \dots + x_n^2 \text{ and } x_0 > 0\} \\ &= \{x \in \mathbb{R}^{n+1} : h(x) := \langle x, x \rangle_{\mathcal{M}} + K = 0 \text{ with } x_0 > 0\}. \end{aligned}$$

Here, the defining function $h(x) = \langle x, x \rangle_{\mathcal{M}} + K$ has differential

$$Dh(x)[u] = 2\langle x, u \rangle_{\mathcal{M}} = (2Jx)^T u.$$

Notice that $x_0 \neq 0$ for all $x \in \mathcal{H}^{n,K}$. $2Jx \neq 0$ for all $x \in \mathcal{H}^{n,K}$ since J is invertible, and $Jx = 0$ if and only if $x = 0$. This implies that differential $Dh(x): \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ is surjective (i.e., $\text{rank } Dh(x) = 1$) for all $x \in \mathcal{H}^{n,K}$. By [2, Definition 3.10 & Theorem 3.15], we conclude the next definition.

Definition 6.1. Given any constant $K > 0$, the set $\mathcal{H}^{n,K}$ is an n -dimensional embedded submanifold of \mathbb{R}^{n+1} with tangent space:

$$\begin{aligned} T_x \mathcal{H}^{n,K} &= \ker Dh(x) = \{u \in \mathbb{R}^{n+1} : \langle x, u \rangle_{\mathcal{M}} = 0\} \\ &= \{u \in \mathbb{R}^{n+1} : x_0 u_0 = \sum_{i=1}^n x_i u_i\} \end{aligned}$$

which is an n -dimensional subspace of \mathbb{R}^{n+1} .

The restriction of $\langle \cdot, \cdot \rangle_{\mathcal{M}}$ to each tangent space $T_x \mathcal{H}^{n,K}$ defines a Riemannian metric on $\mathcal{H}^{n,K}$, turning it into a Riemannian manifold. With this Riemannian structure, we call $\mathcal{H}^{n,K}$ a *hyperbolic manifold*. The main geometric trait of $\mathcal{H}^{n,K}$ with $n \geq 2$ is that its *sectional curvatures* are negative constant, equal to $-\frac{1}{K}$.

North pole point on $\mathcal{H}^{n,K}$. The point $o := (\sqrt{K}, 0, \dots, 0) \in \mathcal{H}^{n,K}$ is called the north pole point of $\mathcal{H}^{n,K}$. We observe that

$$\begin{aligned} T_o \mathcal{H}^{n,K} &= \{u \in \mathbb{R}^{n+1} : \langle o, u \rangle_{\mathcal{M}} = -\sqrt{K}, u_0 = 0\} \\ &= \{u \in \mathbb{R}^{n+1} : u_0 = 0\} \\ &= \{(0, u') \in \mathbb{R}^{n+1} : u' \in \mathbb{R}^n\} \\ &\cong \mathbb{R}^n. \end{aligned}$$

Thus, if we fix the dimension n , then for different curvatures $-\frac{1}{K}$, the manifolds $\mathcal{H}^{n,K}$ have different north pole points but share the same tangent space at their north pole points.

Riemannian distance on $\mathcal{H}^{n,K}$. The distance function induced by Riemannian metric $\langle \cdot, \cdot \rangle_{\mathcal{M}}$ is

$$d_{\mathcal{M}}^K(x, y) = \sqrt{K} \operatorname{arcosh} \left(-\frac{\langle x, y \rangle_{\mathcal{M}}}{K} \right)$$

for all $x, y \in \mathcal{H}^{n,K}$.

Exponential and logarithmic maps. For $x \in \mathcal{H}_K^n$, $v \in T_x \mathcal{H}_K^n$ and $y \in \mathcal{H}_K^n$ such that $v \neq 0$ and $y \neq x$, the exponential and logarithmic maps are given by:

$$\operatorname{Exp}_x^K(v) = \cosh \left(\frac{\|v\|_x}{\sqrt{K}} \right) \cdot x + \sqrt{K} \sinh \left(\frac{\|v\|_x}{\sqrt{K}} \right) \cdot \frac{v}{\|v\|_x}$$

and

$$\begin{aligned} \operatorname{Log}_x^K(y) &= \frac{d_{\mathcal{M}}^K(x, y)}{\|y + \frac{1}{K} \langle x, y \rangle_{\mathcal{M}} \cdot x\|_{\mathcal{M}}} \cdot \left(y + \frac{1}{K} \langle x, y \rangle_{\mathcal{M}} \cdot x \right) \\ &= \frac{d_{\mathcal{M}}^K(x, y)}{\|\operatorname{Proj}_x(y)\|_{\mathcal{M}}} \cdot \operatorname{Proj}_x(y). \end{aligned}$$

Example. (Mapping from Euclidean space to hyperbolic manifold [3])

Let $x^E \in \mathbb{R}^n$ denote input Euclidean features. Let $o := (\sqrt{K}, 0, \dots, 0)$ denote the north pole in \mathcal{H}_K^n , which we use as a reference point to perform tangent space operations. We interpret (o, x^E) as a point in $T_o \mathcal{H}_K^n$ and have

$$\begin{aligned}
 x^H &:= \exp_o^K((o, x^E)) \\
 &= \cosh\left(\frac{\|(o, x^E)\|_{\mathcal{M}}}{\sqrt{K}}\right) \cdot o + \sqrt{K} \sinh\left(\frac{\|(o, x^E)\|_{\mathcal{M}}}{\sqrt{K}}\right) \cdot \frac{(o, x^E)}{\|(o, x^E)\|_{\mathcal{M}}} \\
 &= \cosh\left(\frac{\|x^E\|_2}{\sqrt{K}}\right) \cdot o + \sqrt{K} \sinh\left(\frac{\|x^E\|_2}{\sqrt{K}}\right) \cdot \frac{(0, x^E)}{\|x^E\|_2} \\
 &= \left(\sqrt{K} \cosh\left(\frac{\|x^E\|_2}{\sqrt{K}}\right), \sqrt{K} \sinh\left(\frac{\|x^E\|_2}{\sqrt{K}}\right) \cdot \frac{x^E}{\|x^E\|_2}\right).
 \end{aligned}$$

For the last equality, notice the position of the zero elements in o and $(0, x^E)$ as vectors of \mathbb{R}^n .

REFERENCES

- [1] Ittai Abraham, Yair Bartal, and Ofer Neimany. 2006. Advances in metric embedding theory. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. 271–286.
- [2] Nicolas Boumal. 2023. *An introduction to optimization on smooth manifolds*. Cambridge University Press. <https://doi.org/10.1017/9781009166164>
- [3] Ines Chami, Zhitaoying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems* 32 (2019).
- [4] Kevin Verbeek and Subhash Suri. 2014. Metric embedding, hyperbolic space, and social networks. In *Proceedings of the thirtieth annual symposium on Computational geometry*. 501–510.