

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.

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	

Table 2: Overview of performance. The left of \rightarrow denotes the source domain dataset, and the name on the right denotes the target domain dataset. N@10 and H@10 are abbreviations for the metrics ndcg@10 and hit@10, respectively. * represents the significance level p -value < 0.05 . The highest scores for each dataset and metric are emphasized in bold, while the second-best ones are underlined. Improvement in the last line denotes the relative improvement compared to the best baseline.

Models	Amazon						Douban					
	Book \rightarrow Movie		Book \rightarrow Music		Movie \rightarrow Toy		Book \rightarrow Music		Movie \rightarrow Book		Movie \rightarrow 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
EMCDR	0.0202	0.0573	0.0148	0.0453	0.0249	0.0568	0.0290	0.0833	0.0425	0.1045	0.0303	0.0833
CCDR	0.0171	0.0557	0.0118	0.0397	<u>0.0289</u>	<u>0.0605</u>	0.0194	0.0448	0.0253	0.1125	0.0242	0.1142
ART-CAT	<u>0.0236</u>	<u>0.0718</u>	<u>0.0334</u>	<u>0.1008</u>	0.0285	0.0586	<u>0.0308</u>	<u>0.1616</u>	<u>0.0462</u>	<u>0.1784</u>	<u>0.0458</u>	<u>0.1792</u>
HCTS (ours)	0.0361*	0.0969*	0.0512*	0.1279*	0.0328*	0.0645*	0.0474*	0.1898*	0.0486*	0.2045*	0.0474*	0.1845*

2 EXPERIMENTS ON LARGE DATASET

To validate the scalability of our model, we selected a larger dataset for experiments. Table 4 shows the detailed statistics of the datasets and Table 3 shows the experimental results. It is evident that our model also achieves best results in the context of large datasets. The results below show that HCTS also outperforms other models on large-scale datasets. CCDR, being a GAT-based model, underperform in most cases on smaller datasets, as is shown in Table 2. however, on larger datasets, its results are surpassed only by HCTS.

Table 3: 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
CCDR	<u>0.0368</u>	<u>0.0685</u>
HCTS (ours)	0.0372	0.0706

Table 4: 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, 2] 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

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

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 volumn 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 volumn 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 $d = \Omega\left(\sqrt{K} \ln n\right)$.

PROOF. Witout loss of generality, we suppose f is non-expanding, which means

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

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 volumn of a ball with rarius 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

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

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, $d = \Omega(\sqrt{K} \ln n)$.

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 6 and Table 5 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 6, 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 5: HCTS with trainable curvature

Table 6: 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

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] 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.