1 COMPUTATIONAL COSTS IN THE HCTS FRAMEWORK

1.1 Time complexity of all graph neural network-based models

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: $O(B_1 dN)$
    * Item-item contrastive learning: $O(B_2 dN)$
  - 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 Actual computation time

We also calculated the average time required by each model to compute each epoch in the Douban Book-Movie experiments.

2 COMPARATIVE EXPERIMENTS WITH THE LATEST RESEARCH

2.1 Details of Baseline Models

In this appendix, we introduce the baseline models in detail as follows:

- **ART-CAT** [2] is a multi-domain cross domain recommendation model, which applies Contrastive Autoencoder for learning single domain representations and Attention-based method for transferring knowledge. To align with our experimental settings, we only one dataset as source domain and one dataset as target domain.

Table 1: Actual computation time and visulization

Method	Time (s)	Visualization
HGCF	1.41	
LightGCN	0.94	
GCF	0.99	
BiTGCF	4.36	
CoNET	2.06	
DTCDR	0.66	
CMF	0.49	
DeepAPF	0.51	
CLFM	0.56	
HCTS	4.47	

- **EMCDR** [3] combines Matrix Factorization and Bayesian Personalized Ranking with a mapping function based on Multi-Layer Perceptron to align user latent factors across different domains.
- **CCDR** [5] is a GAT based model and use contrastive learning for knowledge transfer.

Table 2: Overview of performance. The left of → 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→Movie		Book→Music		Movie→Toy		Book→Music		Movie→Book		Movie→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	0.0289	0.0605	0.0194	0.0448	0.0253	0.1125	0.0242	0.1142
ART-CAT	0.0236	0.0718	0.0334	0.1008	0.0285	0.0586	0.0308	0.1616	0.0462	0.1784	0.0458	0.1792
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***

3 EXTRA ABLATION EXPERIMENTS

We have included four additional ablation experiments and the results are shown in Table 6. To validate the effectiveness of hyperbolic space, we adapted our model to a Euclidean version by remove all hyperbolic operations and change hyperbolic similarity into cosine similarity in Euclidean space. And we seperately remove user-user, user-item and item-item contrastive learning to find out how each parts of them affects the final results of the prediction. From the table, we can find the following observations:

(1) Substituting hyperbolic space with Euclidean space has a significant impact on the final results.
(2) The ablation study of the contrastive learning strategy indicates that the effectiveness of these three strategies varies across different datasets. For example, in the ablation experiments of Amazon Book-Movie and Amazon Movie-Toy, the user-user contrastive learning shows significant effects, whereas in the experiments for Amazon Book-Music, the item-item contrastive learning is more pronounced. In the Douban Movie-Music experiments, the user-item contrastive learning is particularly significant.
(3) Removing any part of the strategy has a negative impact on the final results, suggesting that all three contrastive learning strategies are effective and that their combined use can enhance the robustness of the model.

Table 3: *HCTS-Euc* denotes the version of our model in Euclidean space. *HCTS w/o u-u* denotes HCTS without user-user contrastive learning. *HCTS w/o u-i* denotes HCTS without user-item contrastive learning and *HCTS w/o i-i* denotes HCTS without item-item contrastive learning. And * represents the significance level p−value < 0.05.

Models	Amazon						Douban					
	Book→Movie		Book→Music		Movie→Toy		Book→Music		Movie→Book		Movie→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
HCTS-Euc	0.0216	0.0685	0.0404	0.1027	0.0228	0.0441	0.0328	0.1388	0.0475	0.1900	0.0457	0.1757
HCTS w/o u-u	0.0345	0.0926	0.0503	0.1240	0.0257	0.0584	0.0471	0.1837	0.0450	0.1948	0.0475	0.1828
HCTS w/o u-i	0.0350	0.0954	0.0464	0.1114	0.0269	0.0612	0.0468	0.1837	0.0459	0.1990	0.0459	0.1775
HCTS w/o i-i	0.0356	0.0956	0.0389	0.1008	0.0271	0.0615	0.0472	0.1828	0.0462	0.1996	0.0476	0.1819
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***

4 EXPERIMENTS ON LARGE DATASET

To validate the scalability of our model, we selected a larger dataset for experiments. The

Table 4: 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
CCDR	0.0329	0.0648
ART-CAT	<u>0.0368</u>	<u>0.0685</u>
HCTS (ours)	**0.0372**	**0.0706**

Table 5: Dataset information and experiment results

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

5 DIFFERENT IMPLEMENTATIONS OF SINGLE DOMAIN METHODS

Table 6: Results of other implementations of single domain models. In Table 5, LightGCN-merge and HGCF-merge refer to the results obtained by directly merging the source and target domain data and testing solely on the target domain. LightGCN-pretrain and HGCF-pretrain denote the approaches where the model is first pre-trained on the source domain, then the embeddings of the overlapped users obtained during pre-training are used as the initial feature vectors for fine-tuning on the target domain. LightGCN-single and HGCF-single represent the outcomes of training exclusively on the target domain, which are consistent with the results reported in our paper.

Models	Amazon						Douban					
	Book→Movie		Book→Music		Movie→Toy		Book→Music		Movie→Book		Movie→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
LightGCN-merge	0.0212	0.0625	0.0274	0.0824	0.0286	0.0586	0.0383	0.1482	0.0283	0.1234	0.0426	0.1657
LightGCN-single	0.0230	0.0653	0.0449	0.1066	0.0299	0.0591	0.0409	0.1643	0.0295	0.1240	0.0419	0.1643
LightGCN-pretrain	0.0241	0.0682	0.0457	0.1092	0.0301	0.0597	0.0428	0.1684	0.0316	0.1342	0.0428	0.1672
HGCF-merge	0.0272	0.0752	0.0407	0.1095	0.0267	0.0591	0.0389	0.1503	0.0292	0.1355	0.0153	0.0835
HGCF-single	0.0347	0.0942	0.0488	0.1163	0.0253	0.0568	0.0444	0.1722	0.0481	0.1990	0.0448	0.1722
HGCF-pretrain	0.0344	0.0943	0.0492	0.1167	0.0258	0.0574	0.0436	0.1711	0.0479	0.1986	0.0458	0.1728
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***

6 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.

7 THE INFLUENCE OF CURVATURE

7.1 Explanation of curvature

The curvature of hyperbolic space is closely related to the final performance of hyperbolic neural networks. The performance of hyperbolic models is related to distortion, which is a metric that measures the extent to which embeddings can accurately represent the original data. Accurately, it is formally defined in [1, 4] as follows:
Let (X, d_x) and (Y, d_y) be any metric spaces $f : X \longrightarrow Y$ is an embedding. We say f has distortion c for $c \geq 1$ if

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

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:

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

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. We will find that the larger the curvature $\frac{1}{K}$, the faster the growth of the volumn of the hyperbolic space. 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 $s_1, \ldots, s_n$. 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 $i = 1, 2 \ldots n$. 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 $i = 1, 2 \ldots n$ 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

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

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

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

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 rarius 2, for big enough d, the following inequality holds:

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

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

7.2 Experiments for testing the influence of curvature to the final results

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] Chenglin Li, Yuanzhen Xie, Chenyun Yu, Bo Hu, Zang Li, Guoqiang Shu, Xiaohu Qie, and Di Niu. 2023. One for all, all for one: Learning and transferring user embeddings for cross-domain recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 366–374.

[3] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach.. In *IJCAI*, Vol. 17. 2464–2470.

[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.

[5] Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. 2022. Contrastive cross-domain recommendation in matching. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4226–4236.