# WHAT DOES HYPERBOLIC SPACE ROLES FOR GRAPH LEARNING?

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Models, like graph neural networks, built upon hyperbolic spaces have achieved great success in tree-structured data, but it is not clear in what aspect the hyperbolic space plays an effective role. In this paper, we take the recommender system, a user-item graph-structured networks, as an example for an in-depth analysis. Given that the prevalence of the power-law distribution in user-item graph-structured networks, hyperbolic space has attracted considerable attention and achieved impressive performance recently. The advantage of hyperbolic recommendation lies in that its exponentially increasing capacity is well-suited to describe the power-law distributed user-item network whereas the Euclidean equivalent is deficient in. Nonetheless, it remains unclear which aspects are effective or counterproductive with hyperbolic models. To address the above concerns, we take the one of the most popular recommendation techniques, collaborative filtering, as the medium, to investigate the behaviors of hyperbolic and Euclidean graph models. The results reveal that tail nodes get more emphasise in the hyperbolic model than that built upon Euclidean space, but there is still ample room for improvement; head nodes receive modest attention in hyperbolic space, which could be considerably improved; and nonetheless, the hyperbolic models show more competitive performance.

## 1 INTRODUCTION

With the growth of Amazon, Netflix, TikTok, and other e-commerce or social networking services over the past several years, recommender systems are becoming ubiquitous in the digital age. Recommender systems, in a broad sense, are algorithms that try to suggest relevant or potentially preferable items to the users(items being movies to watch, articles to read, products to buy, etc).

Collaborative filtering, one of the most extensively used techniques in customized recommendation, is based on the assumption that users often get the preferable suggestions from someone with similar preferences. To provide relevant recommendations, collaborative-filtering approaches (Koren et al., 2009; Koren, 2008; Liang et al., 2018; Luo et al., 2014) rely on historical interactions between users and items, which are stored in the user-item matrix. Recently, researchers have proposed to explicitly incorporate the high-order collaborative interaction to enhance the recommendation performance. Usually, the user-item relationship is modeled as a bipartite graph with nodes representing users or items and edges representing their interactions. After then, the graph neural networks (GNNs) (Kipf & Welling, 2017; Veličković et al., 2018; Hamilton et al., 2017) are applied to extract the high-order relationships between users and items via the message propagation paradigm. By using layers of neighborhood aggregation under the graph convolutional setup to construct the final representations, these techniques (Wang et al., 2019; He et al., 2020; Sun et al., 2021) have attained state-of-the-art performance on diverse benchmark datasets.

The heavy-tailed distribution[1] occurs in most large-scale recommendation datasets where the number of popular items is liked by a large number of users accounts for the minority and the rest are the majority which is unpopular ones. In general, popular items are competitive while the long-tail

[1]Heavy-tailed distributions are substantially right-skewed, with a small number of large values in the head and a large number of small values in the tail; they are often described by a power law, a log-normal, or an exponential function.

Table 1: Statistics of the experimental data.

| Dataset | #User | #Item | | | #Interactions | Density |
|---|---|---|---|---|---|---|
| | | All | H20(%) | T80(%) | | |
| Amazon-CD | 22,947 | 18,395 | 46 | 54 | 422,301 | 0.10% |
| Amazon-Book | 52,406 | 41,264 | 47 | 53 | 1,861,118 | 0.09% |
| Yelp2020 | 71,135 | 45,063 | 62 | 37 | 1,940,014 | 0.05% |

item reflects personalized preference or something new. Both of them are critical for the recommendation. Recently, hyperbolic space has gained increasing interest in the recommendation area as the capacity of hyperbolic space exponentially increases with radius, which fits nicely with a power-law distributed user-item network. Naturally, hyperbolic graph neural network-based models achieve competitive performance in recommender systems (Sun et al., 2021; Chen et al., 2022). However, it is not clear in what respects the hyperbolic model is superior to the Euclidean counterpart? At the same time, it is unclear in which aspects hyperbolic models perform worse than Euclidean models?

To answer the above doubts, in this work, we take the simplest form of recommendation model - collaborative filtering (CF) - as an example to analyze and observe the behaviors of hyperbolic and Euclidean models. Specifically, we take LightGCN (He et al., 2020) and HGCF (Sun et al., 2021) as an example for analysis and observation, both of which are essentially the same model applied in different spaces. Specifically, we compare the recommendation effects of hyperbolic and Euclidean models, as well as their performance on the head and tail item, using the similar model configuration and running environment. The head and tail items are essentially chosen by the 20/80 rule[2], which states that all items are ranked according to their degrees, and the top 20% is considered as the head (abbreviated as H20), while the remaining 80% is referred to as the tail (abbreviated as T20). The experimental findings reveal the following facts. The tail item receives more consideration in the hyperbolic model than it does in the Euclidean model, but there is still plenty of room for improvement while the head item receives marginal attention in hyperbolic space, which might be substantially improved. Overall, the hyperbolic models outperform the Euclidean space models. These findings are of great significance to community of hyperbolic graph neural network and recommender systems, since they may help researchers better understand the advantages and disadvantages of hyperbolic models, as well as when and where to deploy them.

## 2 Models

The core idea behind Euclidean and hyperbolic graph collaborative filtering (He et al., 2020; Wang et al., 2019; Sun et al., 2021; Chen et al., 2022) is to extract high-order interactions within user and items through message aggregation and build the representation for users and items. This technique will cluster users who like the same items together, as well as items that are liked by the same users. The basic concept behind Euclidean and hyperbolic graph collaborative filtering (He et al., 2020; Wang et al., 2019; Sun et al., 2021; Chen et al., 2022) is to extract high-order dependencies between users and items via message aggregation mechanism. By graph collaborative filtering, users who like the same items, as well as items that are liked by the same users, will be grouped together. Similar to Euclidean graph collaborative filtering, hyperbolic graph collaborative filtering has three components: (1) hyperbolic encoding layer; (2) hyperbolic neighbor aggregation; and (3) prediction layer.

**Hyperbolic encoding layer.** The purpose of the hyperbolic encoding layer is to create hyperbolic initial embedding for users and items. Gaussian distribution initialization is a typical method in Euclidean space. Similarly, an embedding initialization technique is based on the Wrapped Normal Distribution (Nagano et al., 2019), which is a hyperbolic version of the Gaussian distribution (Sun et al., 2021; Wang et al., 2021; Chen et al., 2022). Formally, we use $\mathbf{x} \in \mathbb{R}^n$ to represent the Euclidean node state (including the user and item). Then the initial hyperbolic node state $\mathbf{e}_i^0$ and $\mathbf{e}_u^0$

[2]Mathematically, the 80/20 rule is roughly described by a power-law distribution (also known as a Pareto distribution)

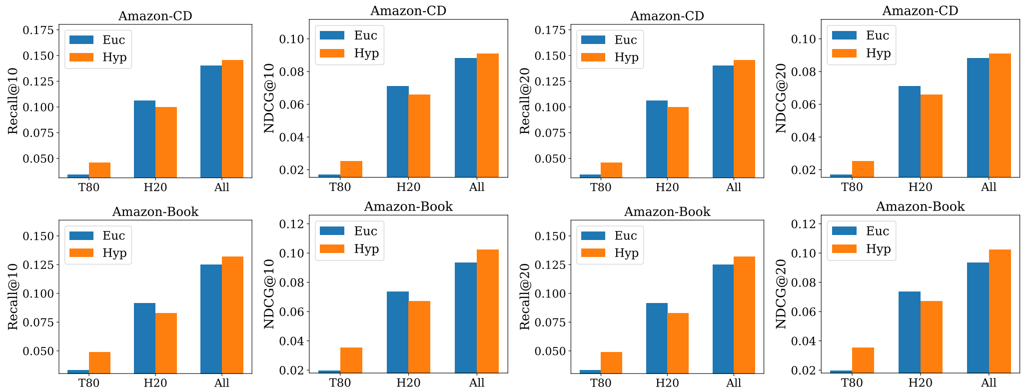


Figure 1: Comparisons of Euclidean and hyperbolic models on Amazon-CD and Amazon-Book datasets. `Euc` represents Euclidean model, LightGCN, and `Hyp` denotes hyperbolic model, HGCF.

can be obtained by:

$$\begin{aligned} \mathbf{e}_i^0 &= \exp_{\mathbf{o}}(\mathbf{z}_i^0), \qquad \mathbf{e}_u^0 = \exp_{\mathbf{o}}(\mathbf{z}_u^0) \\ \mathbf{z}_i^0 &= (0, \mathbf{x}_i), \qquad \mathbf{z}_u^0 = (0, \mathbf{x}_u) \end{aligned} \tag{1}$$

where $\mathbf{x}$ is taken from multivariate Gaussian distribution. $\mathbf{z}^0 = (0, \mathbf{x})$ denotes the operations inserting value 0 into the zero-th coordinate of $\mathbf{x}$ so that $\mathbf{z}^0$ can always live in the tangent space of origin. The superscript 0 in $\mathbf{e}^0$ and $\mathbf{z}^0$ indicates the initial or zero-th layer state.

**Hyperbolic neighbor aggregation.** Hyperbolic neighbor aggregation is to extract explicit user-item interaction. The hyperbolic neighbor aggregation is computed by aggregating neighboring representations of user and item from previous aggregation. Given the neighbors $\mathcal{N}_i$ and $\mathcal{N}_u$ of $i$ and $u$, respectively, the embedding of user $u$ and $i$ is updated utilizing the tangent state $\mathbf{z}$ and the $k$-th (k¿0) aggregation is given by:

$$\mathbf{z}_i^k = \mathbf{z}_i^{k-1} + \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|} \mathbf{z}_u^{k-1}, \qquad \mathbf{z}_u^k = \mathbf{z}_u^{k-1} + \sum_{i \in \mathcal{N}_u} \frac{1}{|\mathcal{N}_u|} \mathbf{z}_i^{k-1}. \tag{2}$$

where $|\mathcal{N}_u|$ and $|\mathcal{N}_i|$ are the number of one-hop neighbors of $u$ and $i$, respectively. For high-order aggregation, the sum-pooling are applied on these $K$ tangential states:

$$\mathbf{z}_i = \sum_{k=1}^{K} \mathbf{z}_i^k, \qquad \mathbf{z}_u = \sum_{k=1}^{K} \mathbf{z}_u^k. \tag{3}$$

Note that $\mathbf{z}$ is on the tangent space of origin. For the hyperbolic state, it is projected back to the hyperbolic space with the exponential map,

$$\mathbf{e}_i = \exp_{\mathbf{o}}(\mathbf{z}_i), \qquad \mathbf{e}_u = \exp_{\mathbf{o}}(\mathbf{z}_u), \tag{4}$$

where $\mathbf{e}_i$ and $\mathbf{e}_u$ represents the final hyperbolic embeddings.

**Prediction layer**. Through hyperbolic neighbor propagation, explicitly structural information is embedded in the user and item embeddings. To infer the preference of a user to an item, the hyperbolic distance $d_{\mathcal{H}}$ can be utilized for the prediction, $p(u,i) = 1/d_{\mathcal{H}}^2(\mathbf{e}_u, \mathbf{e}_i)$. Since we concerned with the rank of preferred items, the negative form can likewise be used for prediction, i.e, $p(u,i) = -d_{\mathcal{H}}^2(\mathbf{e}_u, \mathbf{e}_i)$.

## 3 Investigation

In this work, we use three public recommendation datasets, namely Amazon-CD[1], Amazon-Book[3] and Yelp2020[4]. Note that we only use user-item interactions to maintain consistency with the comparison models. The statistics of the dataset are in Table 1, where H20 and T80 denote the average ratio of the head items and tail items appear in users preference. They are calculated by

[3]https://jmcauley.ucsd.edu/data/amazon/

[4]https://www.yelp.com/dataset

$\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\#\{\mathcal{N}_u \cap \mathcal{I}_{H20}\}$ and $\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\#\{\mathcal{N}_u \cap \mathcal{I}_{B80}\}$, respectively. Each dataset is split into 80% and 20% training and test sets for training and evaluation, respectively. In these datasets, the ratings are converted into binary preferences by applying a threshold $\geq 4$ which simulates the implicit feedback settings. We employ two standard evaluation metric to assess the performance of top-K recommendation and preference ranking: Recall and NDCG (Ying et al., 2018).

According to the previous research (He et al., 2020; Sun et al., 2021), the hyperbolic model (Sun et al., 2021) performs more competitively than that built upon Euclidean space (He et al., 2020) using models with essentially the same structure. However, it is unclear in what aspects the hyperbolic model excels above its Euclidean equivalent. Simultaneously, it's uncertain in which places hyperbolic models worse than Euclidean models. These issues obstruct our understanding of hyperbolic recommendation models and hinder their applications in real-world scenarios.

To solve the aforementioned doubts, we undertake a quantitative analysis that aims to experimentally study the behaviors of hyperbolic and Euclidean recommendation models by disentangling their performance on the tail and head items. In particular, we first sort items by their degree, which is similar to the popularity, and then split into the head 20%(denoted as H20, or $\mathcal{I}_{H20}$) and the tail 80%, (denoted as T80, $\mathcal{I}_{T80}$). Next, we investigate the recommendation effect via the metric Recall@K and NDCG@K on the H20 and T80 items, respectively, using the Euclidean graph collaborative filter model, LightGCN, and the corresponding hyperbolic model, HGCF. The results are shown in Figure 1. From the experimental results, we have the following observations:

- The overall recommendation performance of the hyperbolic model is better than that of the Euclidean model;
- Tail items get greater emphasis in the hyperbolic model as the results on tail items are far beyond that of the Euclidean counterpart ;
- Head items receive moderate attention in the hyperbolic model as the performance of HGCF is sightly lower than that of LightGCN.

The above findings provide valuable insights on the benefits of hyperbolic space for recommender systems: the exponentially increased capacity of hyperbolic space enables the model to pay more attention to tail items, which is beneficial for personalized recommendation and increasing market diversity.[5] The hyperbolic model is a strong contender, but there are still two main shortage in the present hyperbolic model. (1) Despite the fact that hyperbolic model produces better overall outcomes and has a greater recommendation effect on tail items, there is still large room for improvement. The reason is that tail items account more user's interests in Amazon-CD (54% T80 vs 46% H20) and Amazon-Book (53% T80 vs 47%) as given in Table 1, but the recommendation effect of tail item is much lower than that of head items as shown in Figure 1. (2) Besides, compared with Euclidean space, hyperbolic space reduces the attention of the model on head items to a certain extent. Thus, there is an urgent need to improve the recommendation ability of head items.

## 4 Conclusion

Hyperbolic models have received increasing attention in the recommendation community, while their pros and cons over the Euclidean counterparts have not been explicitly studied. In this work, we attempt to initiate the investigation by further separately comparing their performance on head and tail items against the Euclidean equivalents. Overall, the hyperbolic model shows apparent superiority. It is also observed that the hyperbolic model performs substantially better on Tail items than the Euclidean equivalent, but there is still sufficient room for improvement. For the head item, the hyperbolic model place modest attention.

The above observations shed more light on the role of hyperbolic models in Recommender system. Note that the exponentially increased capacity of hyperbolic space allows the model to pay more attention to tail items, which is beneficial for personalized recommendation and increasing market diversity; In future work, we aim to analyze the advantages and disadvantages of hyperbolic spaces from a more theoretical perspective.

[5] As we know, the head item is popular and liked by a large number of users while the tail item is either personalized reflecting the unique preference of the user or something fresh increasing the diversity of the market.

## REFERENCES

Yankai Chen, Menglin Yang, Yingxue Zhang, Mengchen Zhao, Ziqiao Meng, Jianye Hao, and Irwin King. Modeling scale-free graphs for knowledge-aware recommendation. *WSDM*, 2022.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *ICONIP*, pp. 1025–1035, 2017.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. LightGCN: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pp. 639–648, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pp. 426–434, 2008.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *WWW*, pp. 689–698, 2018.

Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *TII*, 10(2):1273–1284, 2014.

Yoshihiro Nagano, Shoichiro Yamaguchi, Yasuhiro Fujita, and Masanori Koyama. A wrapped normal distribution on hyperbolic space for gradient-based learning. In *ICML*, pp. 4693–4702. PMLR, 2019.

Jianing Sun, Zhaoyue Cheng, Saba Zuberi, Felipe Pérez, and Maksims Volkovs. HGCF: Hyperbolic graph convolution networks for collaborative filtering. In *WWW*, pp. 593–601, 2021.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

Hao Wang, Defu Lian, Hanghang Tong, Qi Liu, Zhenya Huang, and Enhong Chen. Hypersorec: Exploiting hyperbolic user and item representations with multiple aspects for social-aware recommendation. *TOIS*, pp. 1–28, 2021.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *SIGIR*, pp. 165–174, 2019.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, pp. 974–983, 2018.