Logical Relation Modeling and Mining in Hyperbolic Space for Recommendation

Yanchao Tan[1,2,3], Hang Lv[1,2,3], Zihao Zhou[1,2,3], Wenzhong Guo[1,2,3]*, Bo Xiong[4], Weiming Liu[5], Chaochao Chen[5], Shiping Wang[1,2,3], Carl Yang[6]
[1]*College of Computer and Data Science, Fuzhou University, Fuzhou, China*
[2]*Engineering Research Center of Big Data Intelligence, Ministry of Education, Fuzhou, China*
[3]*Fujian Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou, China*
[4]*Institute for Artificial Intelligence, University of Stuttgart, Stuttgart, Germany*
[5]*College of Computer Science, Zhejiang University, Hangzhou, China*
[6]*Department of Computer Science, Emory University, Atlanta, United States*
{*yctan@fzu.edu.cn*, {*lvhangkenn, reviverkey*}*@gmail.com, guowenzhong@fzu.edu.cn, bo.xiong@ipvs.uni-stuttgart.de*,
{*21831010, zjuccc*}*@zju.edu.cn, shipingwangphd@163.com, j.carlyang@emory.edu*}

***Abstract*—The sparse interactions between users and items have aggravated the difficulty of their representations in recommender systems. Existing methods leverage tags to alleviate the sparsity problem but ignore prevalent logical relations among items and tags (e.g., membership, hierarchy, and exclusion), which can be leveraged to enhance the accuracy of modeling user preferences and conducting recommendations. To this end, we propose to extract logical relations among item tags from existing tag taxonomies and exploit the individual strengths of the Poincaré and the Lorentz models in hyperbolic space for logical relation modeling towards enhanced recommendations. Moreover, we find that the logical relations directly extracted from existing tag taxonomies can be inaccurate and coarse. Therefore, we further devise innovative consistency-based and granularity-based weighting mechanisms based on user behavior patterns for data-driven logical relation mining that can be jointly optimized along with recommendations in an end-to-end fashion. Extensive experiments on four real-world benchmark datasets show drastic performance gains brought by our proposed framework, which constantly achieves an average of 8.25% improvement over state-of-the-art competitors regarding both Recall and NDCG metrics. Insightful case studies further demonstrate that our automatically refined logical relations are highly accurate and interpretable.**

I. Introduction

Recommender systems based on traditional collaborative filtering methods suffer from the sparsity issue of user-item interactions. To address the sparsity challenge, it is a common practice to combine collaborative filtering with auxiliary data. Among them, item tags are one of the most commonly used types of auxiliary data due to their vast availability and rich semantics, which can be used to improve user modeling and recommendation [6], [22], [24], [64].

However, existing tag-based methods ignore the prevalent membership relation among items and tags, hierarchical/exclusive relations among tags. We uniformly term the above three relations as **logical relations** in taxonomies for specific recommendation scenarios, which can be helpful in producing accurate and consistent recommendations that respect the inherent logical constraints among items as indicated

*Corresponding author

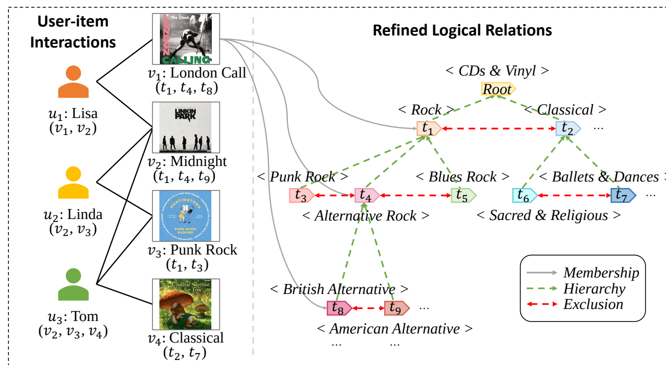


Fig. 1: **An illustrative example of recommendation with refined logical relations.** The tags are marked with <>. Note that refined logical relations are not directly derived from the existing tag taxonomies. Moreover, various levels of hierarchies and exclusions have different degrees of impact.

by the tags. As shown in Fig. 1, an item London Call should belong to tag t_1 <*Rock*> (i.e., **membership relation**) if it belongs to t_4 <*Alternative Rock*> (according to **hierarchical relation**) and should not belong to t_2 <*Classical*> (according to **exclusive relation**). Therefore, we can skip items under t_2 <*Classical*> when recommending items for Lisa or Linda since they only interact with items under t_1 <*Rock*>, which can lead to more explainable and consistent recommendations yet with significant reductions on computation cost [29].

To effectively and efficiently model the logical relations, we notice the techniques of hyperbolic representation learning. Since the volume of hyperbolic space expands exponentially, it can reliably model objects with hierarchical relations [30], [31], [40] (the detailed analysis can be found in Section III-B). However, these studies fail to explore membership and exclusive relations that are set-theoretic, which should be modeled via convex regions rather than single points in the hyperbolic space. Moreover, how to seamlessly integrate logical relation modeling with recommender systems remains unclear, where the key objective is to properly adjust the embedding spaces of both users and items while maintaining the multiple types of logical relations indicated by the item tags.

In this work, we propose to jointly model the Logical relations and perform Recommendation (`LogiRec`), which aims to make full use of tags to learn fine-grained representations and enhance recommendations (Section IV). Specifically, we propose to leverage both points and hyperplanes in the Poincaré model for logical relation modeling, where we model tags as convex regions and items as points with intuitive constraints. In this way, the logical relations among items and tags can be modeled by the geometric relations between their corresponding embeddings. Then, considering the different advantages of the Poincaré [30] and the Lorentz [31] models for the interpretability and the stable optimization of models, respectively, we innovatively exploit these two types of hyperbolic models together and design a hyperbolic graph convolutional network for user and item representation learning that can be seamlessly optimized towards the objective of recommendation.

One step further, we find that the logical relations directly derived from the existing tag taxonomies can be inaccurate and coarse [54], since the assumption that there is exclusion between tags at the same taxonomic level when they share the same parent tag and no common child tag may not always hold. For example, the exclusive relation between tag <*Heavy Metal*> and <*Metal*> can be inaccurate since the two are actually overlapping concepts and both belong to tag <*Hard Rock*>. Moreover, it is coarse to simply make tag <*Punk Rock*> and <*Alternative Rock*> be exclusive to the same degree as tag <*Punk Rock*> and <*Blues Rock*>, because <*Punk Rock*> and <*Alternative Rock*> are both rebellious music genre while <*Blues Rock*> is a soulful and emotional guitar-driven music genre. Although tags can be helpful to group items and users with similar properties and preferences, without further refining logical relations to enhance their accuracy and granularity, the model can fail to obtain accurate and fine-grained representations for users and items, which can even backfire the recommendation performance.

In light of this, we propose to refine logical relations via a data-driven approach based on user behaviors. We deem that users with consistent and specific preferences can be more helpful in refining logical relations, and their interactions should make larger impacts on the optimization of recommendations. As shown in Fig. 1, Linda who has consistent preferences over items under tag t_1 <*Rock*> can reasonably make the originally exclusive t_3 <*Punk Rock*> and t_4 <*Alternative Rock*> closer to each other; however, since her preferences are not specific into different types of t_4 <*Alternative Rock*>, her preference should not impact the closeness between t_8 <*British Alternative*> and t_9 <*American Alternative*> much. Therefore, we devise intuitive weighting mechanisms based on the novel utilization of exclusive relations and hyperbolic embeddings towards quantifications of user preference consistency and granularity, so as to adaptively adjust the impacts of different users for data-driven logical relation mining. We term this improved version of `LogiRec` with logical relation mining as `LogiRec++` (Section V).

We evaluate both `LogiRec` and `LogiRec++` with extensive experiments on four real-world benchmark datasets for recommendations. We compare them with 13 comprehensive methods focusing on state-of-the-art metric learning-based, tag-based, and graph-based recommendation methods. Extensive experimental results show that `LogiRec++` is able to significantly improve the recommendation overall baselines (e.g., with up to 14.43% improvements in Recall@10 on Book over the best baseline). More comprehensive results and discussion as well as ablation studies, hyperparameter studies, and case studies are presented and analyzed in Section VI.

In summary, we mainly make the following contributions:

- *Formulation of logic-enhanced recommendation.* `LogiRec++` is the first recommendation framework with explicit handling of logical relations, which can provide consistent and interpretable predictions and refine logical relations without additional supervision.
- *Effective model designs.* In `LogiRec`, we exploit the individual strengths of two hyperbolic models (i.e., Poincaré and Lorentz) to integrate logical relation modeling and recommendation as a whole. In `LogiRec++`, we devise intuitive consistency-based and granularity-based weighting mechanisms based on user behaviors for effective joint training of logical relation mining and recommendation.
- We conduct extensive experiments on four real-world datasets, which demonstrate significant improvements of the proposed `LogiRec++` framework on recommendation together with highly accurate and interpretable results of logical relation mining.

II. Related Work

A. Metric Learning for Collaborative Filtering

The metric learning methods for collaborative filtering use different distances to measure the similarity between users and items. Compared with methods based on matrix factorization (MF) that assumes linear relations between users and items [25], [34], [60], metric learning methods that satisfy the triangle inequality can better model the complex interactions in real-world applications [14], [39], [45], and thus can address the limitations of MF. For example, Hsieh et al. [14] proposed to learn a metric space to encode not only users' preferences but also the user-user and item-item similarities. Vinh Tran et al. [48] proposed to learn one-to-one mappings between Euclidean and hyperbolic spaces. Furthermore, to capture higher-order graph structure for user (item) representation such as neighbors-of-neighbors relations among users and items, existing methods [46], [65] measured the proximity with graph convolutional neural (GCN) networks model for collaborative filtering. For example, Tian et al. [46] proposed a recurrent graph convolutional network from both a user's clicked history and a knowledge graph. Zheng et al. [65] proposed an end-to-end diversified recommendation model based on GCN.

However, the above metric learning methods can be limited by the sparse user-item interactions [27], [63]. Some studies [14], [53] tried to alleviate the sparsity problem via leveraging auxiliary data, such as review [38] and tags [44]. For example, Shuai et al. [38] proposed a review-aware graph contrastive learning framework for recommendation. To the best of our knowledge, there is no metric learning work that has explicitly leveraged set-theoretic logical relations (e.g., membership relation and exclusive relation) for accurate and interpretable recommendations.

B. Taxonomy-based Recommendation

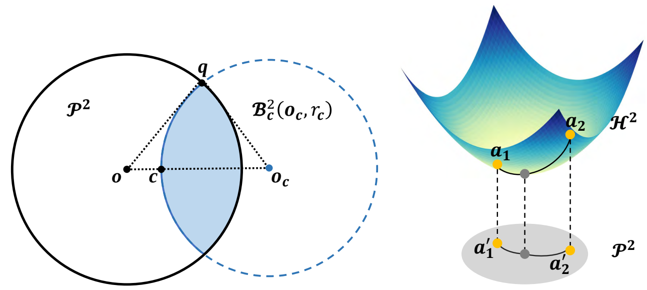
Taxonomies have garnered considerable interest across various fields for their intrinsic value and the inherently tree-like hierarchy [2], [33], [43]. In recommender systems, taxonomies are often employed to address issues of data sparsity and computational intensity. For instance, Ziegler et al. [67] exploited taxonomic background knowledge to infer users' profiling effectively. Tan et al. [44] constructed a tag taxonomy automatically to leverage structural knowledge among tags.

While taxonomies provide a structural understanding, knowledge graphs (KGs) have also been widely used to mitigate the sparsity challenge [7], [50], [56], [58]. KGs encapsulate a variety of semantic relations, whereas tag taxonomies typically embody structural relations (e.g., hierarchy). The integration of taxonomies into recommender systems presents distinct advantages over KGs. Firstly, KGs include numerous relations that may be irrelevant to recommendation tasks, which can blindly incur excessive computational costs and potentially deteriorate recommendation quality [44]. Secondly, by concentrating on structural relations, taxonomies can be mapped into hyperbolic spaces, leveraging their tree structures for effective embedding learning [1], [55], thereby enhancing the accuracy and interpretability of recommendations.

Despite the apparent benefits of taxonomies in enhancing recommendations, recent taxonomy-aware approaches [15], [23], [44], [61], [67] have overlooked some logical relations beyond mere hierarchy, such as exclusive relation. Drawing inspiration from Xiong et al. [54], we propose to capture three logical relations derived from existing tag taxonomies.

C. Hyperbolic Embedding Learning

Various data exhibit an underlying hierarchical structure that the Euclidean space embeddings suffer from distortion issues [9], [48]. To mitigate this problem, Nickel et al. [30] proposed to learn representation in the Poincaré ball model of hyperbolic space, which can naturally accommodate hierarchical structures and is convenient for visualization. By taking advantage of the Poincaré model, Hui et al. [16] proposed to learn hyperbolic embeddings based on tensor decomposition, and Iyer et al. [17] proposed to learn hyperbolic and heterogeneous relations of knowledge graphs. Furthermore, expanding on [30], Nickel et al. [31] found that learning representations based on the Lorentz formulation of the hyperbolic space are well-suited for Riemannian optimization. By constraining embedding in the Lorentz model, Dai et al. [8] proposed a hyperbolic-to-hyperbolic graph convolutional network for avoiding distortion; Sun et al. [41] proposed to learn dynamic graph representation for inferring stochastic node representations; Chen et al. [7] attempted to learn Lorentzian embeddings for knowledge graphs. Moreover, based on a hyperbolic graph attention network, Wang et al. [51] proposed the hyperbolic embedding model for knowledge graph reasoning.



(a) (b)

Fig. 2: (a) Hyperplane (light blue shallow) in Poincaré model. (b) The connection between the Poincaré and Lorentz models.

Recently, hyperbolic representation learning has also been applied to recommender systems [5], [10], [18]. For example, Mirvakhabova et al. [28] used a single-layer autoencoder in hyperbolic space to learn user and item embeddings. Chami et al. [5] proposed a weighted margin rank batch loss to learn a hyperbolic model and generated user representation by item aggregation in hyperbolic space via Einstein midpoint. Zhang et al. [62] learned geometric disentangled representations for user intentions, and Yang et al. [57] designed a geometric-aware collaborative hyperbolic regularizer. Zhou et al. [66] regarded each user-item interaction as an event in hyperbolic space and modeled the probability of event occurrence for temporal recommendation. Our approach is related to these works in that we also learn user and item representations in hyperbolic space. However, a key difference is that our approach jointly exploits the individual strengths of the different hyperbolic models to mine logical relations, so as to deliver accurate and interpretable recommendations through the existing tag taxonomies.

III. Preliminaries

A. Hyperbolic Models

The Poincaré model. The Poincaré model $\mathcal{P}^d = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| < 1\}$ is defined as a set of d-dimensional vectors with Euclidean norm smaller than 1. The Poincaré distance metric is defined as: $d_{\mathcal{P}}(\boldsymbol{x}, \boldsymbol{y}) = \cosh^{-1}\left(1 + 2\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{(1-\|\boldsymbol{x}\|_2^2)(1-\|\boldsymbol{y}\|_2^2)}\right)$. A sample in the Poincaré model $\mathcal{P}^d$ can be represented via either a point or a Poincaré hyperplane [54]. Specifically, let $\mathcal{B}^d$ denote the set of d-balls in $\mathbb{R}^d$ whose boundaries $\partial\mathcal{B}^d$ intersect the Poincaré ball $\mathcal{P}^d$ perpendicularly. Poincaré hyperplanes are defined by $\partial\mathcal{B}^d \cap \mathcal{P}^d$ plus all linear subspaces going through the origin (i.e., the light blue shallow in Fig. 2(a)). Hence, a Poincaré hyperplane can be uniquely defined by its center point $\boldsymbol{c}$ that has a minimal distance to the origin [54]. The Poincaré hyperplane can be defined as $\boldsymbol{H_c} = \{\boldsymbol{x} \in \mathcal{P}^d : g_{\mathcal{P}}(\log_{\boldsymbol{c}}(\boldsymbol{x}), \vec{\boldsymbol{c}}) = 0\}$, where $\boldsymbol{c} \neq \boldsymbol{0}$ denotes the center point,

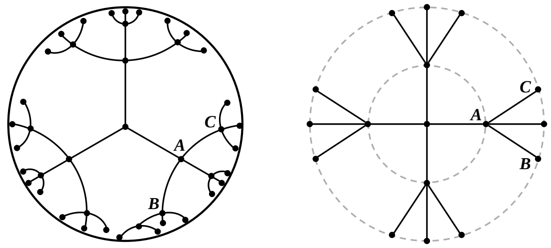


Fig. 3: **The comparison between two-dimensional hyperbolic space (left) and Euclidean space (right)**, where the tag $\boldsymbol{A}$ is <*Rock*>, $\boldsymbol{B}$ is <*Blues Rock*>, and $\boldsymbol{C}$ is <*Alternative Rock*>. Note that, all the black edges have identical lengths.

$\vec{c} \in \mathcal{T}_{\boldsymbol{c}}\mathcal{P}^d$ denotes the normal vector passing through the origin $\mathbf{0}$, and $g_{\mathcal{P}}$ is a Riemannian metric. Note that the tangent space $\mathcal{T}_{\boldsymbol{c}}\mathcal{P}^d$ is a Euclidean subspace of $\mathbb{R}^d$. An enclosing d-ball $\mathcal{B}^d\left(\boldsymbol{o}_{\boldsymbol{c}}, r_{\boldsymbol{c}}\right)$ can be defined as $\mathcal{B}^d\left(\boldsymbol{o}_{\boldsymbol{c}}, r_{\boldsymbol{c}}\right) = \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{o}_{\boldsymbol{c}}\| \leq r_{\boldsymbol{c}}\}$, where $(\boldsymbol{o}_{\boldsymbol{c}}, r_{\boldsymbol{c}}) = \left(\frac{1+\|\boldsymbol{c}\|^2}{2\|\boldsymbol{c}\|}\boldsymbol{c}, \frac{1-\|\boldsymbol{c}\|^2}{2\|\boldsymbol{c}\|}\right)$. In particular, the hyperplane $\boldsymbol{H}_{\boldsymbol{c}}$ is a subset of the d-ball $\mathcal{B}^d\left(\boldsymbol{o}_{\boldsymbol{c}}, r_{\boldsymbol{c}}\right)$.

The Lorentz model. The Lorentz model is the only unbounded hyperbolic model [59] and is defined as $\mathcal{L}^d = (\mathcal{H}^d, g_{\mathcal{L}})$ with points constrained by $\mathcal{H}^d = \{\boldsymbol{x} \in \mathbb{R}^{d+1} : \langle \boldsymbol{x}, \boldsymbol{x}\rangle_{\mathcal{L}} = 1, \boldsymbol{x}_0 \geq 0\}$, where $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}}$ is the Lorentzian scale inner product: $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}} = -\boldsymbol{x}_0\boldsymbol{y}_0 + \sum_{i=1}^{d}\boldsymbol{x}_i\boldsymbol{y}_i$, and the metric tensor is: $g_{\mathcal{L}}(\boldsymbol{x}) = \operatorname{diag}(-1, 1, \ldots, 1)$. The associated distance function in the Lorentz model is given as: $d_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y}) = \cosh^{-1}(-\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}})$.

Strengths and connections between hyperbolic models. The metric in the Poincaré model satisfies all the properties of a distance metric and is interpretable for visualization [31]. Due to its characteristic of curvature space, the Poincaré model exists infinitely many non-parallel hyperplanes (i.e., Poincaré hyperplanes) which do not intersect [54]. Furthermore, each Poincaré hyperplane can be interpreted as a convex region. In this way, the Poincaré hyperplanes can support models to capture complicated set-theoretic interactions such as implication and exclusion, which can be respectively modeled by geometric insideness and disjointness between the corresponding regions. The Lorentz model allows for an efficient closed-form computation of the geodesics on the manifold and is suited for Riemannian optimization [4], [20], [31].

Due to the equivalence of the Poincaré and the Lorentz models [31], we can exploit the models' individual strengths simultaneously. In particular, as shown in Fig. 2(b), points in the Lorentz model can be mapped into the Poincaré model via diffeomorphism p as:

$$p(\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_d) = \frac{(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d)}{\boldsymbol{x}_0 + 1}. \tag{1}$$

Furthermore, points in the Poincaré model can be mapped into the Lorentz model via diffeomorphism p^{-1} as:

$$p^{-1}\left(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d\right) = \frac{\left(1 + \|\boldsymbol{x}\|^2, 2\boldsymbol{x}_1, \cdots, 2\boldsymbol{x}_d\right)}{1 - \|\boldsymbol{x}\|^2}. \tag{2}$$

B. Comparison between Hyperbolic and Euclidean spaces

Existing studies have found flaws in Euclidean space [30], [31], [35], where the polynomial expansion has bounded the ability of the model to represent complex patterns by the dimensionality of embedding space. In particular, Euclidean space cannot fully capture the beneficial structural properties existing in tag taxonomy (e.g., hierarchy), which may cause high-distortion embeddings. Many recent studies demonstrate that hyperbolic space is capable of modeling such reliably hierarchical structure [40], [49], [44], whose volume expands exponentially. Specifically, the sum of the distances between the points and the origin in hyperbolic space is larger than that in Euclidean space, providing a clearer arrangement to separate embeddings of data points in more fine-grained levels of the hierarchy. Therefore, hyperbolic space has a stronger representation ability than Euclidean space [49].

Taking tags modeling for example, as shown in Fig. 3, with the requirement of clearly distinguishing relations among tags in the area of $\boldsymbol{BAC}$, the number of tags that hyperbolic space can carry is greater than that of Euclidean space. Note that all the black edges have identical lengths, which also demonstrates that the volume of hyperbolic space is larger. Since the limited sum of the distances between the tags and the origin in Euclidean space will make it hard to arrange all hierarchical tags properly [36], [18]. The suboptimal tag embedding optimization in Euclidean space will lead to suboptimal taxonomy construction, which may cause incorrect hierarchical relations modeling and weakly constructed tag taxonomies. As shown in the left of Fig. 3, we can observe that clear and correct hierarchical structures meet our goal of properly arranging tags in the taxonomy in hyperbolic space. The tag $\boldsymbol{B}$ <*Blues Rock*> can be closer to its immediate parent $\boldsymbol{A}$ <*Rock*>, while distant from its sibling $\boldsymbol{C}$ <*Alternative Rock*> (i.e., $\widehat{\boldsymbol{BA}} = \widehat{\boldsymbol{AC}} < \widehat{\boldsymbol{BC}}$). However, as shown in the right of Fig. 3, Euclidean space fails to model the relations among tags with $\boldsymbol{BA} = \boldsymbol{AC} = \boldsymbol{BC}$.

IV. The LogiRec Framework

In this section, we present our joint Logical relation modeling and Recommendation (`LogiRec`) framework, as shown in Fig. 4. We first give an overview of `LogiRec`. Then, we explore membership, hierarchical, and exclusive relations in the Poincaré model. Finally, we perform logic-based recommendation based on the item embeddings constrained by logical relations.

A. Method Overview

To explicitly model logical relations for recommendations, we first initialize tag embeddings in the Poincaré model and denote them as $\boldsymbol{T} = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_S\}$. Specifically, $\boldsymbol{t}_i \in \mathcal{P}^d$ is assigned as a corresponding Poincaré hyperplane and S is the number of tags. According to the item-tag matrix $\boldsymbol{Q}$, we propose to model the membership relations among items and tags via a membership objective function $\mathcal{L}_{Mem}$, where we denote the item embeddings in the Poincaré model as $\boldsymbol{v}^{\mathcal{P}} \in \mathcal{P}^d$. Then, we transform the logical relations between tags into geometric relations between their corresponding Poincaré hyperplanes, where the hierarchical relation is modeled by the geometric insideness via $\mathcal{L}_{Hie}$ while the exclusive relation is modeled

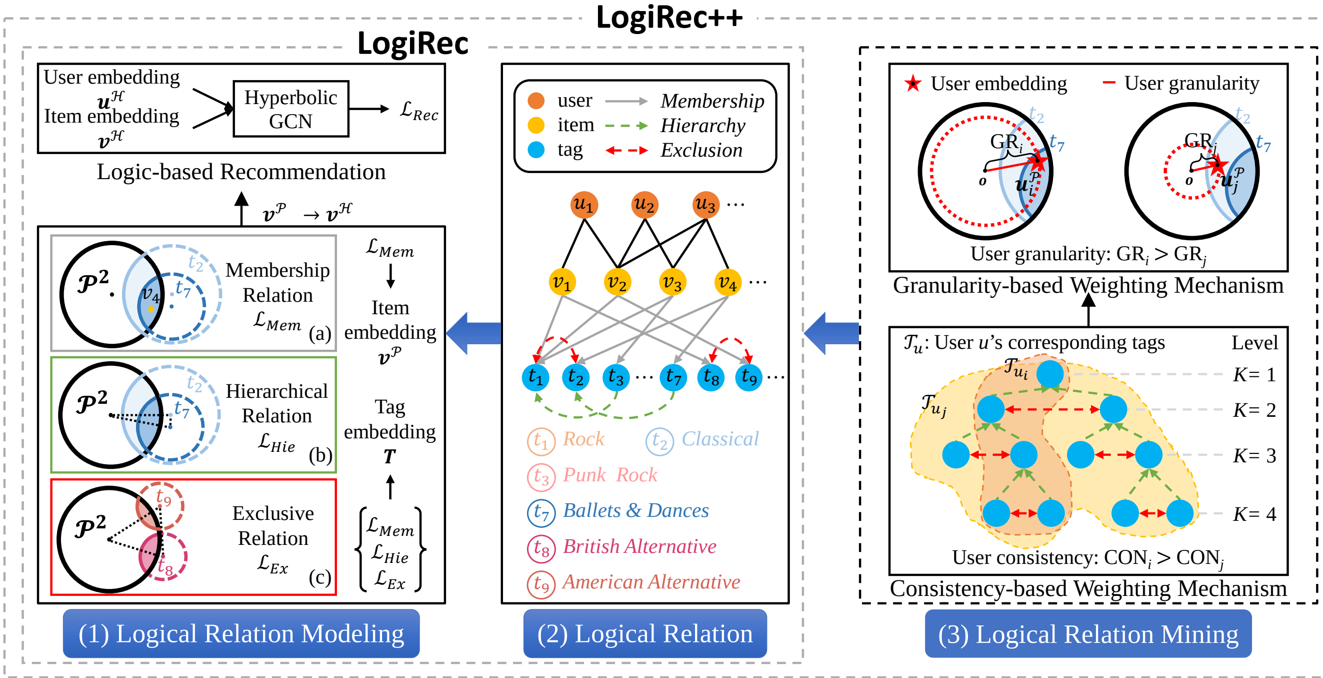


Fig. 4: **Overview of our proposed `LogiRec` and `LogiRec++` framework.** Specifically, `LogiRec` explicitly models three logical relations (i.e., membership, hierarchy, and exclusion) in Poincaré model. Then we perform logic-based recommendation based on the item embeddings constrained by logical relations and user embeddings in the Lorentz model. `LogiRec++` further devises consistency-based and granularity-based weighting mechanisms based on the novel utilization of exclusive relations and hyperbolic embeddings in (3), so as to adaptively adjust the impacts of different users in logical relation mining.

by the geometric disjointness via $\mathcal{L}_{Ex}$. Furthermore, we map the learned item embedding $\boldsymbol{v}^{\mathcal{P}}$ into the Lorentz model as $\boldsymbol{v}^{\mathcal{H}}$ and denote user embedding $\boldsymbol{u}^{\mathcal{H}}$, so as to stably optimize the representation of users and items. Finally, we perform logic-based recommendation via the objective function $\mathcal{L}_{Rec}$ based on the user embedding $\boldsymbol{u}^{\mathcal{H}}$ and the item embedding $\boldsymbol{v}^{\mathcal{H}}$.

B. Logical Relation Modeling

As motivated in Section I, it is important to model logical relations for interpretable recommendations, where membership, hierarchical, and exclusive relations can be extracted from the existing tag taxonomies together with item-tag relations according to [54]. However, the above relations come from set-theoretic semantics, which should be measured via the region-based rather than point-based representations [47], [54], and thus the recent studies in hyperbolic space [36], [18] based on point-based embeddings fail to accurately model such set-theoretic logical relations.

Recall that Poincaré hyperplanes are interpreted as convex regions to layout sets in Section III, we propose to leverage both points and hyperplanes in the Poincaré model for the logical relation modeling of our `LogiRec`. For example, an item point London Call can be included by a tag hyperplane <*Alternative Rock*>, where the region of <*Alternative Rock*> can be further included by a larger hyperplane <*Rock*> that represents more abstract concepts than <*Alternative Rock*>. Fig. 4 shows how these three relations are transformed into soft geometric constraints in hyperbolic space and we describe each relation in detail.

*1) **Membership Relation Modeling**:* Since an item can have multiple tags that are recorded in the item-tag matrix $\boldsymbol{Q}$, we propose to leverage both points and hyperplanes for modeling the membership relations among items and tags.

Lemma 1: (Membership property). An instance $\boldsymbol{v}$ is inside a d-ball $\mathcal{B}^d_{\boldsymbol{t}}(\boldsymbol{o_t}, r_{\boldsymbol{t}})$ if and only if $\|\boldsymbol{v} - \boldsymbol{o_t}\| < r_{\boldsymbol{t}}$.

In particular, items are modeled as points and tags are modeled as hyperplanes. For example, an item $\boldsymbol{v}_i$ can be described by a tag $\boldsymbol{t}$, hence the corresponding geometric relation is a point $\boldsymbol{v}_i^{\mathcal{P}} \in \mathcal{P}^d$ being inside a d-ball $\mathcal{B}^d_{\boldsymbol{t}}$. Then, we define the membership objective function $\mathcal{L}_{Mem}$ by measuring the geometric membership (i.e., $\|\boldsymbol{v}_i^{\mathcal{P}} - \boldsymbol{o_t}\| < r_{\boldsymbol{t}}$) as follows:

$$\mathcal{L}_{Mem}(\boldsymbol{v}_i^{\mathcal{P}}, \mathcal{B}^d_{\boldsymbol{t}}) = \max\{0, \|\boldsymbol{v}_i^{\mathcal{P}} - \boldsymbol{o_t}\| - r_{\boldsymbol{t}}\}, \tag{3}$$

where $\boldsymbol{v}_i^{\mathcal{P}}$ denotes the item embedding and $\mathcal{B}^d_{\boldsymbol{t}} = (\boldsymbol{o_t}, r_{\boldsymbol{t}})$.

*2) **Hierarchical Relation Modeling**:* Since tags can provide abstract concepts for multiple items and a parent tag can include its children geometrically, we propose to leverage the geometric insideness between the Poincaré hyperplanes of corresponding d-ball $\mathcal{B}^d(\boldsymbol{o_c}, r_{\boldsymbol{c}})$ for hierarchical relations.

Lemma 2: (Hierarchical property). A d-ball $\mathcal{B}^d_{\boldsymbol{t}_i}(\boldsymbol{o}_{\boldsymbol{t}_i}, r_{\boldsymbol{t}_i})$ contains a $\mathcal{B}^d_{\boldsymbol{t}_j}(\boldsymbol{o}_{\boldsymbol{t}_j}, r_{\boldsymbol{t}_j})$ if and only if $\|\boldsymbol{o}_{\boldsymbol{t}_i} - \boldsymbol{o}_{\boldsymbol{t}_j}\| + r_{\boldsymbol{t}_j} < r_{\boldsymbol{t}_i}$.

According to the hierarchical relations in the hyperbolic hyperplane, we transform the logical constrain into soft geometric constraint in the embedding space, where we propose a hierarchy loss as follows:

$$\mathcal{L}_{Hie}(\mathcal{B}^d_{\boldsymbol{t}_i}, \mathcal{B}^d_{\boldsymbol{t}_j}) = \max\{0, \|\boldsymbol{o}_{\boldsymbol{t}_i} - \boldsymbol{o}_{\boldsymbol{t}_j}\| + r_{\boldsymbol{t}_j} - r_{\boldsymbol{t}_i}\}, \tag{4}$$

where d-ball $\mathcal{B}^d_{\boldsymbol{t}_i}(\boldsymbol{o}_{\boldsymbol{t}_i}, r_{\boldsymbol{t}_i})$ contains d-ball $\mathcal{B}^d_{\boldsymbol{t}_j}(\boldsymbol{o}_{\boldsymbol{t}_j}, r_{\boldsymbol{t}_j})$.

*3) **Exclusive Relation Modeling**:* Similarly, to properly model the exclusions between tags, we interpret the exclusion of tags as geometric disjointness between $\mathcal{B}^d(\boldsymbol{o_c}, r_{\boldsymbol{c}})$.

Lemma 3: (Exclusive property). A d-ball $\mathcal{B}^d_{\boldsymbol{t}_i}(\boldsymbol{o}_{\boldsymbol{t}_i}, r_{\boldsymbol{t}_i})$ disconnects from a d-ball $\mathcal{B}^d_{\boldsymbol{t}_j}(\boldsymbol{o}_{\boldsymbol{t}_j}, r_{\boldsymbol{t}_j})$ if and only if $r_{\boldsymbol{t}_i} + r_{\boldsymbol{t}_j} < \|\boldsymbol{o}_{\boldsymbol{t}_i} - \boldsymbol{o}_{\boldsymbol{t}_j}\|$.

Then, we propose an exclusion loss for exclusive relation modeling as follows:

$$\mathcal{L}_{Ex}(\mathcal{B}^d_{\boldsymbol{t}_i}, \mathcal{B}^d_{\boldsymbol{t}_j}) = \max\{0, r_{\boldsymbol{t}_i} + r_{\boldsymbol{t}_j} - \|\boldsymbol{o}_{\boldsymbol{t}_i} - \boldsymbol{o}_{\boldsymbol{t}_j}\|\}, \tag{5}$$

where d-ball $\mathcal{B}^d_{\boldsymbol{t}_i}(\boldsymbol{o}_{\boldsymbol{t}_i}, r_{\boldsymbol{t}_i})$ disjoints from d-ball $\mathcal{B}^d_{\boldsymbol{t}_j}(\boldsymbol{o}_{\boldsymbol{t}_j}, r_{\boldsymbol{t}_j})$.

C. Logic-based Recommendation

Since the logical modeling based on the existing tag taxonomies has not considered the modeling of complex relations between users and items, how to seamlessly integrate logical relation modeling with recommender systems is still an endeavor. Note that the logical relation modeling leverages the Poincaré model to capture complex set-theoretic interactions, while the Lorenz model is stable for numeric optimization of sparse interactions [31], [4], [20]. In light of this, we propose to leverage the individual strengths of both the Poincaré and the Lorentz models for the logic-based recommendation, where a hyperbolic graph convolution network (i.e., Hyperbolic GCN) is devised to capture complicated effects for both logical relation modeling and user decision-making. We name such an integrated framework as `LogiRec`.

In particular, we first denote a learnable user embedding in the Lorentz model as $\boldsymbol{u}^{\mathcal{H}} \in \mathcal{H}^d$. Then, we map item embeddings $\boldsymbol{v}^{\mathcal{P}} \in \mathcal{P}^d$ learned via Eq. 3 in the logical relation modeling module (cf., Section IV-B) into the Lorentz model via diffeomorphism p^{-1} in Eq. 2 (cf., Section III), where we have $\boldsymbol{v}^{\mathcal{H}} = p^{-1}(\boldsymbol{v}^{\mathcal{P}}) \in \mathcal{H}^d$. In this way, we can jointly perform recommendation and logical relation modeling in a unified framework.

Since we cannot apply Euclidean mean aggregation in hyperbolic space, it is needed to project the embeddings to the corresponding tangent space as [40] to obtain the initialization for graph convolution as $\boldsymbol{z}_u^{\mathcal{H},0}$ and $\boldsymbol{z}_v^{\mathcal{H},0}$. Taking user embedding for example, we project $\boldsymbol{u}^{\mathcal{H}}$ to tangent space $\mathcal{T}_{\mathbf{o}}\mathcal{H}^d$ via the logarithmic map $\log_{\mathbf{o}} : \mathcal{H}^d \to \mathcal{T}_{\mathbf{o}}\mathcal{H}^d$ as follows:

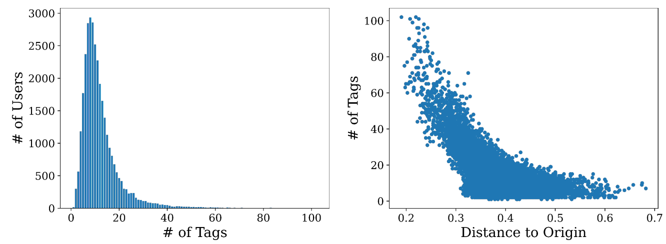
$$\begin{aligned}\boldsymbol{z}_u^{\mathcal{H},0} &= \log_{\mathbf{o}}(\boldsymbol{u}^{\mathcal{H}}) \\ &= \operatorname{arcosh}\left(-\langle \mathbf{o}, \boldsymbol{u}^{\mathcal{H}}\rangle_{\mathcal{L}}\right) \frac{\boldsymbol{u}^{\mathcal{H}} + \langle \mathbf{o}, \boldsymbol{u}^{\mathcal{H}}\rangle_{\mathcal{L}}\mathbf{o}}{\|\boldsymbol{u}^{\mathcal{H}} + \langle \mathbf{o}, \boldsymbol{u}^{\mathcal{H}}\rangle_{\mathcal{L}}\mathbf{o}\|_{\mathcal{L}}},\end{aligned} \tag{6}$$

where $\mathbf{o} = (1, 0, \ldots, 0) \in \mathcal{H}^d$ is the referred origin, $\|\boldsymbol{u}^{\mathcal{H}}\|_{\mathcal{L}} = \sqrt{\langle \boldsymbol{u}^{\mathcal{H}}, \boldsymbol{u}^{\mathcal{H}}\rangle_{\mathcal{L}}}$, and $\mathcal{T}_{\mathbf{o}}\mathcal{H}^d$ is a Euclidean subspace of $\mathbb{R}^{d+1}$.

Then, we can aggregate neighborhood representations and further aggregate them from all intermediate layers:

$$\begin{aligned}\boldsymbol{z}_u^{\mathcal{H},l+1} &= \boldsymbol{z}_u^{\mathcal{H},l} + \sum_{v \in \mathcal{N}_u} \frac{1}{|\mathcal{N}_u|}\boldsymbol{z}_v^{\mathcal{H},l}, \ \boldsymbol{z}_u^{\mathcal{H}} = \sum_{l=1}^{L} \boldsymbol{z}_u^{\mathcal{H},l}, \\ \boldsymbol{z}_v^{\mathcal{H},l+1} &= \boldsymbol{z}_v^{\mathcal{H},l} + \sum_{u \in \mathcal{N}_v} \frac{1}{|\mathcal{N}_v|}\boldsymbol{z}_u^{\mathcal{H},l}, \ \boldsymbol{z}_v^{\mathcal{H}} = \sum_{l=1}^{L} \boldsymbol{z}_v^{\mathcal{H},l},\end{aligned} \tag{7}$$

where $\mathcal{N}_u = \{v | R_{uv} = 1\} \in \mathcal{V}$ is the item set that user u interacts with. $\mathcal{N}_v = \{u | R_{uv} = 1\} \in \mathcal{U}$ is the user set who interact with item v. L is the total number of graph layers.



(a) (b)

Fig. 5: (a) User distribution across different numbers of tag types on the CD dataset. (b) The relation between the number of users' interacted tag types and the corresponding distance to the origin on the CD dataset.

To project the final embedding back into the Lorentz model, we apply an exponential map as follows:

$$\begin{aligned}\boldsymbol{u}^{\mathcal{H}} &= \exp_{\mathbf{o}}(\boldsymbol{z}_u^{\mathcal{H}}) \\ &= \cosh\left(\|\boldsymbol{z}_u^{\mathcal{H}}\|_{\mathcal{L}}\right)\mathbf{o} + \sinh\left(\|\boldsymbol{z}_u^{\mathcal{H}}\|_{\mathcal{L}}\right)\frac{\boldsymbol{z}_u^{\mathcal{H}}}{\|\boldsymbol{z}_u^{\mathcal{H}}\|_{\mathcal{L}}}.\end{aligned} \tag{8}$$

Similarly, we can obtain final item embeddings by replacing the inputs of Eq. 6 with $\boldsymbol{v}^{\mathcal{H}}$, and then apply the exponential map as Eq. 8.

Finally, we utilize the largest margin nearest neighbor algorithm (LMNN) to perform logic-based recommendation as follows:

$$\mathcal{L}_{Rec} = \sum_{(u,v_p)\in\mathcal{I}} \sum_{(u,v_q)\notin\mathcal{I}} \left[m + d(\boldsymbol{u}^{\mathcal{H}}, \boldsymbol{v}_p^{\mathcal{H}}) - d(\boldsymbol{u}^{\mathcal{H}}, \boldsymbol{v}_q^{\mathcal{H}})\right]_+, \tag{9}$$

where $d(\boldsymbol{x}, \boldsymbol{y}) = \cosh^{-1}(\boldsymbol{x}_0\boldsymbol{y}_0 - \sum_{i=1}^{d}\boldsymbol{x}_i\boldsymbol{y}_i)$ is the Lorentzian distance measurement. $\mathcal{I}$ is the set of positive user-item pairs derived from the implicit feedback data $\mathbf{X}$. m is a margin to enforce the difference between triplets which we empirically set to 0.1 by default in our experiments. $[(x)]_+ = \max(x, 0)$ is a standard hinge loss.

The objective function of `LogiRec` is calculated as follows:

$$\min_{\boldsymbol{u}^{\mathcal{H}}, \boldsymbol{v}^{\mathcal{H}}, \boldsymbol{T}} \mathcal{L}_{Rec} + \lambda(\mathcal{L}_{Mem} + \mathcal{L}_{Hie} + \mathcal{L}_{Ex}), \tag{10}$$

where λ is a weight hyperparameter to control the regularization for logical relation modeling.

V. LogiRec with Logical Relation Mining based on User Behaviors (LogiRec++)

The proposed `LogiRec` essentially performs recommendation with logical relations modeling via Eq. 10. However, the logical relations directly extracted from the existing tag taxonomy can be inaccurate and coarse. For example, the exclusive relation between tag <*Heavy Metal*> and <*Metal*> is inaccurate for actually overlapping concepts; making (<*Punk Rock*>, <*Alternative Rock*>) and (<*Punk Rock*> and <*Blues Rock*>) the same degree of exclusion is coarse, since <*Punk Rock*> and <*Alternative Rock*> are both rebellious music genre while <*Blues Rock*> is soulful and emotional. In this way, it is hard to ensure that the accurate logical relations are always available (shown in the right of Fig. 1).

To deal with the above problems, we propose to leverage users' preferences based on their behaviors to further mine logical relations. Specifically, we propose user consistency and granularity for adjusting users' impacts on the optimization of recommendations.

The first insight is: users with consistent preferences should have a large impact on adjusting the logical relations. For example, as shown in Fig. 1, rather than leveraging Tom's diverse preferences across six tags (e.g., t_1 *<Rock>* and t_2 *<Classical>*), we can leverage Linda's consistent preferences with t_1 *<Rock>* to make the interacted t_3 *<Punk Rock>* and t_4 *<Alternative Rock>* be close. As shown in Fig. 5(a), the number of tag types that users interact with is around 10. Although most users have consistent preferences with specific tag types, there are still many long-tail users who interact with more than 20 tag types. This diversity of preferences makes it challenging to accurately profile these users, as well as to train logical relations based on user preferences without being swayed by their extreme diversity. Therefore, we propose to measure the consistency of users by counting the frequency of users' exclusive tags at different levels so as to weigh users' impacts towards the mining of accurate logical relations.

The second insight is: As shown in Fig. 5(b), users with specific preferences (i.e., with less # of tags) are embedded far away from the origin in the hyperbolic space. Due to the exponential growth of the volume of hyperbolic space along the distance to the origin [30], [31], they need large weights to properly rearrange the fine-grained tag embeddings. For example, as shown in Fig. 1, compared with Linda who has coarse-granularity preferences with t_1 *<Rock>* and makes the third level t_3 *<Punk Rock>* and t_4 *<Alternative Rock>* close, Lisa who has finer-granularity preferences with t_4 *<Alternative Rock>* needs more effort to make the forth level t_8 *<British Alternative>* and t_9 *<American Alternative>* close in the larger fine-grained space. Therefore, we propose to measure user granularity via the distance between each user embedding and the space origin, and then pay more attention to users with specific preferences for mining fine-grained logical relations. We name our framework of `LogiRec` with logical relation mining based on user behaviors as `LogiRec++` (shown in the right of Fig. 2).

A. Consistency-based Weighting Mechanism

We observe that users with diverse preferences are usually not constrained by tags, and thus can exist many exclusive relations among their interacted tags. As shown in the bottom right of Fig. 4, the orange area represents the tag of user u_i while the yellow one is for u_j, where $\mathcal{T}_u$ denotes the list of tags that user u interacts with. We can observe that u_i only has one pair of exclusive tags at the fourth level; while u_j contains 5 pairs of exclusions, where the levels range from 2 to 4. It is reasonable that u_i is regarded as a more consistent user than u_j. Such observation also implies that there exists a negative correlation between user consistency and the users' interacted exclusive tags. In particular, the less frequent and lower level of exclusive tags that the user has interacted with, the user has more consistent preferences.

Inspired by the above argument, we propose a consistency-based weighting mechanism as CON_u based on exclusive relations of tags. Specifically, we first define $\mathrm{TF}(t_i, \mathcal{T}_u)$ as the normalized frequency of each tag $t_i \in \mathcal{T}_u$ as follows:

$$\mathrm{TF}(t_i, \mathcal{T}_u) = \frac{\log(|\mathcal{T}_{u,i}| + 1)}{\log(|\mathcal{T}_u|)}, \tag{11}$$

where $\mathcal{T}_u$ denotes the list of tags that user u interacts with. $|\mathcal{T}_u|$ denotes the total number of tags in $\mathcal{T}_u$, and $|\mathcal{T}_{u,i}|$ denotes the number of occurrences of tag t_i in $\mathcal{T}_u$. Then, we consider both the level and frequency of the pairwise tag exclusive relations, and define user consistency CON_u as follows:

$$\begin{aligned}\mathrm{CON}_u = \exp(-\sum_{k=1}^{\eta} \sum_{t_i, t_j \in \mathcal{T}_u} \mathbb{I}(t_i, t_j) \\ * \mathrm{TF}(t_i, \mathcal{T}_u) * \mathrm{TF}(t_j, \mathcal{T}_u) * \exp((\eta - k))),\end{aligned} \tag{12}$$

where $\mathbb{I}(t_i, t_j)$ is an indicator function to judge whether there exists exclusion between t_i and t_j or not. η denotes the total number of levels in tag taxonomy and is empirically set as 4.

The idea behind Eq. 12 is to leverage the tripartite user-item-tag graph to calculate the ratio and level of the exclusive tags in the whole interacted tag list, i.e., user u has the less and the lower level of exclusive tags, user u with a larger CON_u has more consistent preferences towards these tags, and thus the more weights of user u's impacts should be considered for mining accurate logical relations.

B. Granularity-based Weighting Mechanism

To provide intuitive visualization and find the insightful correlation between users and tags, we project the learned user embedding $u^{\mathcal{H}}$ in the Lorentz model into the Poincaré model. As shown in the upper right of Fig. 4, tag t_7 *<Ballets & Dances>* with a specific concept (i.e., fine-grained tag) has small radiuses and the shortest distance from t_7 *<Ballets & Dances>* hyperplane to the space origin is large; while tag t_2 *<Classical>* that has a more abstract concept than t_7 *<Ballets & Dances>* (i.e., coarse-grained tag) has a large radius and its shortest distance to the origin is small. According to an inversely proportional relation between the tag granularity and the distance between tag embedding to the space origin, we can infer that users who have fine-granularity preferences that interacted with fine-grained tags are also far away from the origin. In the upper right of Fig. 4, we can observe that the distance between the user embedding $\boldsymbol{u}_i^{\mathcal{P}}$ and the origin $\boldsymbol{o}$ is larger than the distance between $\boldsymbol{u}_j^{\mathcal{P}}$ and $\boldsymbol{o}$, which shows that user u_i has finer-granularity preferences than user u_j. Considering a larger moving distance of fine-granularity users due to the exponential growth of the volume of hyperbolic space along the distance to the origin [30], [31], a greater optimization effort is required to properly rearrange the fine-grained tag embeddings.

Based on the above argument, we propose a granularity-based weighting mechanism and associate it with the distance between user embedding $\boldsymbol{u}^{\mathcal{H}}$ and the origin $\boldsymbol{o}$ as follows:

$$\mathrm{GR}_u = \cosh^{-1}(-\langle \boldsymbol{o}, \boldsymbol{u}^{\mathcal{H}} \rangle_{\mathcal{L}}). \tag{13}$$

In this way, the farther distance of u away from the origin, the user u with a larger value of user granularity GR_u is regarded as a fine-granularity user, thus the more weights of user u's interactions should be considered for refining logical relations.

To ensure the weighting strategy satisfies both user consistency and granularity, we have the following design similar to [37], [44]:

$$\alpha_u = \sqrt{\mathrm{CON}_u \cdot \mathrm{GR}_u}, \tag{14}$$

where CON_u can be referred in Eq. 12, and GR_u can be referred in Eq. 13. With such personalized weight α_u, we rewrite Eq. 10 to obtain the objective function of LogiRec++ as follows:

$$\min_{\boldsymbol{u}^{\mathcal{H}}, \boldsymbol{v}^{\mathcal{H}}, \boldsymbol{T}} \sum_u \alpha_u \mathcal{L}^u_{Rec} + \lambda(\mathcal{L}_{Mem} + \mathcal{L}_{Hie} + \mathcal{L}_{Ex}), \tag{15}$$

where λ is a weight hyperparameter to control the regularization for logical relation modeling. In this way, the user with a large impact α_u can hold a large proportion of the gradient feedback, which can make full use of their consistent and fine-granularity preferences for reorganizing the embedding space and further refining the logical relations; while the user with a low value of α_u can make a small impact on the optimization of model, which can reduce the user's impact on the mining of accurate logical relations.

C. Riemannian Optimization

Different from traditional Euclidean gradient descent optimization, we apply the Riemannian SGD [3] for optimization. The Riemannian gradient $\mathrm{grad}(\mathcal{L}(\mathcal{X}_t))$ can be obtained by

$$\mathrm{grad}(\mathcal{L}(\mathcal{X}_t)) = (\boldsymbol{I} - \mathcal{X}_t \mathcal{X}_t^T)\nabla(\mathcal{L}(\mathcal{X}_t)). \tag{16}$$

Denoting the $\mathcal{X}$ as the variable set, the parameters are updated by $\mathcal{X}_{t+1} = \exp_{\mathcal{X}_t}(-\beta_t \, \mathrm{grad}(\mathcal{L}(\mathcal{X}_t)))$, where the exp operations in the Poincaré model and the Lorentz model are different and will be introduced as follows:

Optimizating $\mathcal{L}_{Hie}$, $\mathcal{L}_{Ex}$, and $\mathcal{L}_{In}$. In this scenario, items and tags are embedded in the Poincaré model. Therefore, we use Möbius exponential map and we take tag embeddings $\boldsymbol{T}$ as an example:

$$\exp_{\boldsymbol{T}}(\eta) = \boldsymbol{T} \oplus \left(\tanh\left(\frac{\|\eta\|}{2}\right)\frac{\eta}{\|\eta\|}\right) = \boldsymbol{T} \oplus \boldsymbol{y}, \tag{17}$$

where $\boldsymbol{T} \oplus \boldsymbol{y} = \frac{(1+2\langle \boldsymbol{T}, \boldsymbol{y}\rangle + \|\boldsymbol{y}\|^2)\boldsymbol{T} + (1-\|\boldsymbol{T}\|^2)\boldsymbol{y}}{1+2\langle \boldsymbol{T}, \boldsymbol{y}\rangle + \|\boldsymbol{T}\|^2\|\boldsymbol{y}\|^2}$ is the Möbius addition, and $\boldsymbol{y} = \tanh\left(\frac{\|\eta\|}{2}\right)\frac{\eta}{\|\eta\|}$.

Optimizating $\mathcal{L}_{Rec}$. In this scenario, the embeddings are computed in the Lorentz model. We take $\boldsymbol{v}^{\mathcal{H}}$ for example and show how to optimize. The exponential map is defined as:

$$\exp_{\boldsymbol{v}^{\mathcal{H}}}(\eta) = \cosh(\|\eta\|_{\mathcal{L}})\boldsymbol{v}^{\mathcal{H}} + \sinh(\|\eta\|_{\mathcal{L}})\frac{\eta}{\|\eta\|_{\mathcal{L}}}. \tag{18}$$

VI. Experiments

In this section, we evaluate our proposed LogiRec and LogiRec++ framework focusing on the following four research questions:

- **RQ1:** How do LogiRec and LogiRec++ framework perform in comparison to state-of-the-art recommendation methods?
- **RQ2:** What are the effects of the model components?
- **RQ3:** How do the hyperparameters affect the recommendation performance and how to choose optimal values?
- **RQ4:** How does LogiRec++ improve recommendations with interpretability via logical relation mining?

TABLE I: Statistics of the datasets used in our experiments.

	Ciao	CD	Clothing	Book
# User	5180	32589	63986	79368
# Item	8836	20559	19727	62385
# Interaction	104905	515562	704325	4657501
Density(%)	0.2292	0.0769	0.0558	0.0941
# Tag	28	379	3051	510
# Membership	8900	45976	86639	124394
# Hierarchy	16	361	4804	636
# Exclusion	22	1572	195004	5392

A. Experimental Setup

1) Datasets: In order to comprehensively verify the effectiveness of compared methods, we use four real-world datasets from different application domains with different sizes and interaction densities, i.e., Ciao[1], Amazon CDs & Vinyl (CD)[2], Amazon Clothing (Clothing)[2], and Amazon Books (Book)[2]. These datasets have been widely adopted in previous literature [32], [48], [42], and their statistics are summarized in Table I.

2) Evaluation protocols: We split the data into training, validation, and testing sets based on timestamps given in the datasets to provide a recommendation evaluation setting. For each user, we use the first 60% of data as the training set, 20% data as validation set, and 20% data as the testing set. We evaluate the recommendation performance using two metrics: Recall@K and NDCG@K instead of sampled metrics as suggested in [19]. Intuitively, the Recall metric considers whether the ground-truth is ranked amongst the top K items while the NDCG metric is a position-aware ranking metric.

3) Methods for comparison: The following representative state-of-the-art baselines can be divided into four groups: (1) **general recommendation methods** (BPRMF [34], NeuMF [12]), (2) **metric learning methods** (CML [14], SML [21], HyperML [48]), (3) **tag-based methods** (CMLF [14], AMF [13], TransC [26], AGCN [53]), and (4) **graph-based methods** (LightGCN [11], HGCF [40], GDCF [62], HRCF [57]).

4) Implementation Details: We implement our LogiRec and LogiRec++ framework with Pytorch. The full code for this work is available[3]. Implementations of the general recommendation methods are either from open-source projects or the original authors (BPRMF/CML[4], NeuMF[5], SML[6], HyperML[7], LightGCN[8], HGCF[9], GDCF[10], and HRCF[11]). Implementations of the tag-based methods are constrained to leverage item tags

[1] https://www.cse.msu.edu/ tangjili/datasetcode/truststudy.htm
[2] http://jmcauley.ucsd.edu/data/amazon/
[3] https://github.com/Melinda315/LogiRec
[4] https://github.com/cheungdaven/DeepRec
[5] https://github.com/hexiangnan/neural_collaborative_filtering
[6] https://github.com/MingmingLie/SML
[7] https://github.com/lucasvinhtran/hyperml
[8] https://github.com/gusye1234/LightGCN-PyTorch
[9] https://github.com/layer6ai-labs/HGCF
[10] https://github.com/ydzhang-stormstout/GDCF
[11] https://github.com/marlin-codes/HRCF

TABLE II: Experimental results (%) on four benchmark datasets, where * denotes a significant improvement according to the Wilcoxon signed-rank test [52]. The best performances are in **boldface** and the second runners are underlined.

Method	Recall@10	Recall@20	NDCG@10	NDCG@20	Recall@10	Recall@20	NDCG@10	NDCG@20
	Ciao				CD			
BPRMF	3.18±0.13	4.90±0.15	2.26±0.10	3.15±0.16	6.18±0.21	9.55±0.26	4.42±0.20	5.37±0.24
NeuMF	3.27±0.18	5.13±0.20	2.73±0.19	3.26±0.20	6.06±0.21	8.44±0.23	4.19±0.23	4.96±0.22
CML	3.67±0.23	5.84±0.26	2.68±0.19	3.40±0.21	6.22±0.15	9.60±0.17	4.55±0.14	5.66±0.19
SML	3.60±0.17	5.76±0.19	2.75±0.16	3.44±0.15	6.33±0.25	9.83±0.22	4.92±0.18	6.06±0.20
HyperML	3.81±0.21	6.17±0.26	2.96±0.16	3.74±0.21	7.89±0.25	12.03±0.21	5.79±0.21	7.11±0.26
CMLF	3.73±0.21	5.92±0.24	2.79±0.18	3.52±0.18	6.32±0.25	9.71±0.23	4.72±0.24	5.79±0.24
AMF	3.56±0.21	5.46±0.23	2.65±0.24	3.41±0.28	6.25±0.27	9.61±0.25	4.72±0.22	5.82±0.24
TransC	4.68±0.14	7.40±0.13	3.87±0.15	4.77±0.11	8.10±0.16	12.89±0.14	6.32±0.14	7.33±0.18
AGCN	6.10±0.08	9.14±0.11	4.99±0.06	5.86±0.10	9.07±0.13	13.63±0.12	7.11±0.10	8.35±0.11
LightGCN	5.17±0.18	7.86±0.14	4.17±0.15	5.10±0.15	9.77±0.14	14.22±0.15	7.47±0.14	9.02±0.17
HGCF	5.98±0.13	9.35±0.11	4.80±0.13	5.90±0.11	10.01±0.12	14.56±0.12	7.58±0.16	9.21±0.17
GDCF	6.06±0.07	9.50±0.09	4.85±0.06	6.07±0.04	10.14±0.11	14.79±0.13	7.91±0.07	9.46±0.08
HRCF	6.25±0.06	9.72±0.11	4.87±0.04	6.18±0.05	10.49±0.13	15.14±0.16	8.07±0.08	9.65±0.10
`LogiRec`	6.54±0.08	10.11±0.10	5.11±0.05	6.30±0.06	10.81±0.10	15.59±0.08	8.26±0.11	9.87±0.13
`LogiRec++`	**6.67±0.05***	**10.30±0.06***	**5.21±0.04***	**6.39±0.03***	**11.04±0.12***	**15.83±0.11***	**8.49±0.09***	**10.04±0.07***
	Clothing				Book			
BPRMF	12.04±0.15	13.43±0.12	10.32±0.13	12.19±0.10	4.14±0.14	7.26±0.13	5.34±0.12	6.23±0.18
NeuMF	12.38±0.16	14.06±0.17	12.47±0.14	12.62±0.14	4.22±0.15	7.28±0.16	5.41±0.14	6.31±0.15
CML	13.90±0.15	16.01±0.16	12.24±0.15	13.16±0.13	4.53±0.11	7.64±0.15	5.85±0.09	6.92±0.06
SML	13.63±0.14	15.79±0.13	12.56±0.13	13.31±0.11	4.42±0.18	7.57±0.11	5.65±0.11	6.62±0.14
HyperML	14.42±0.19	16.91±0.17	13.52±0.15	14.47±0.16	4.79±0.21	7.94±0.23	6.18±0.17	7.20±0.18
CMLF	14.13±0.13	16.23±0.16	12.75±0.09	13.62±0.10	4.63±0.14	7.66±0.13	5.87±0.07	6.95±0.12
AMF	13.48±0.15	14.96±0.18	12.11±0.13	13.20±0.12	4.57±0.13	7.60±0.19	5.79±0.18	6.73±0.14
TransC	20.32±0.13	23.18±0.15	18.59±0.11	19.51±0.16	4.82±0.09	7.76±0.11	6.09±0.12	7.12±0.08
AGCN	23.10±0.09	25.05±0.08	21.80±0.07	22.72±0.07	4.63±0.12	7.67±0.13	5.92±0.11	7.01±0.13
LightGCN	19.58±0.11	21.54±0.12	19.05±0.10	19.74±0.12	4.36±0.10	7.11±0.09	5.53±0.09	6.44±0.12
HGCF	22.34±0.14	24.58±0.13	20.69±0.11	21.84±0.10	4.84±0.12	7.99±0.11	6.15±0.15	7.18±0.15
GDCF	22.61±0.13	24.97±0.14	21.14±0.11	22.52±0.09	4.88±0.13	8.01±0.12	6.26±0.12	7.25±0.09
HRCF	22.76±0.14	25.36±0.10	21.36±0.13	22.58±0.06	5.06±0.05	8.06±0.06	6.39±0.05	7.36±0.04
`LogiRec`	24.53±0.12	27.62±0.14	23.17±0.10	24.04±0.11	5.49±0.10	8.83±0.09	7.04±0.07	8.08±0.08
`LogiRec++`	**25.26±0.14***	**28.23±0.12***	**23.71±0.11***	**24.73±0.09***	**5.79±0.07***	**9.10±0.09***	**7.27±0.06***	**8.33±0.08***

according to the original authors (CMLF[4], AMF[12], TransC[13], and AGCN[14], where TransC is constrained to model tag-tag, item-tag, and user-item relations. We optimize the compared Euclidean baselines with standard SGD and the hyperbolic ones with Riemannian SGD. We tune all hyperparameters through grid search. In particular, learning rate in {1e-5, 5e-5, 1e-4, 5e-4, 1e-3}, the number of graph layer L in {1, 2, 3, 4}, the weight λ in {0, 0.01, 0.1, 1.0, 1.5}, margin m in {0, 0.1, 0.2, 0.3}, and the embedding dimension d in {32, 64, 128}. The batch size is set to 10000. We also carefully tuned the hyperparameters of all baselines through cross-validation as suggested in the original papers to achieve their best performance.

B. Overall Performance Comparison (RQ1)

We compare the performance of our `LogiRec` and `LogiRec++` frameworks to those of the baselines and have the following observations:

In general, `LogiRec` and `LogiRec++` both outperform all 12 baselines across all evaluation metrics on all datasets. This answers RQ1, showing that the recommender systems with explicit handling of the logical relations are capable of effective collaborative ranking. The ranking of many baselines is fluctuating across datasets, where the model of second-best performance scatters between AGCN and HRCF. Compared with the second-best performance, the performance gains of `LogiRec++` on Ciao, CD, Clothing, and Book range from reasonably large (3.40% achieved with NDCG@20 on Ciao) to significantly large (14.43% achieved with Recall@10 on Book). Note that the improvements of `LogiRec++` are more significant when the number of logical relations is larger and the interactions of users and items are sparse, like with Clothing and Book, which supports the appropriate design of our model to make full use of tags and alleviate the sparsity problem in recommendations.

In particular, by considering latent hierarchies in hyperbolic space, HRCF performs better than AGCN in many cases. Compared with HRCF in the Lorentz model, `LogiRec++` exploits the individual strengths of the Poincaré and the Lorentz models for interpretability and optimization. In this way, `LogiRec++` can leverage set-theoretic relations rather than the only point-based hierarchical relation of HRCF for jointly logical relation mining and recommendation. Therefore, `LogiRec++` outperforms HRCF by up to 14.43% in Recall@10 on Book.

However, the learned hierarchies do not always perfectly match reality without the help of tag information, and thus AGCN can sometimes achieve better performance by considering flat item tags directly. Compared with AGCN, our `LogiRec++` not only considers hierarchical tags but also models exclusive tags and membership relations among

[12]https://github.com/cthurau/pymf
[13]https://github.com/davidlvxin/TransC
[14]https://github.com/yimutianyang/AGCN

TABLE III: Ablation results (%) of our proposed `LogiRec++` on the four datasets.

Method	Ciao				CD			
	Recall@10	Recall@20	NDCG10	NDCG@20	Recall@10	Recall@20	NDCG@10	NDCG@20
`LogiRec++`	6.67 ± 0.05	10.30 ± 0.06	5.21 ± 0.04	6.39 ± 0.03	11.04 ± 0.12	15.83 ± 0.11	8.49 ± 0.09	10.04 ± 0.07
- w/o. $\mathcal{L}_{Mem}$	5.71 ± 0.05	9.08 ± 0.07	4.73 ± 0.11	5.86 ± 0.09	9.73 ± 0.13	14.27 ± 0.12	7.75 ± 0.10	9.24 ± 0.09
- w/o. $\mathcal{L}_{Hie}$	6.43 ± 0.07	9.95 ± 0.08	4.98 ± 0.06	6.04 ± 0.06	10.49 ± 0.11	15.25 ± 0.10	8.16 ± 0.08	9.76 ± 0.09
- w/o. $\mathcal{L}_{Ex}$	6.49 ± 0.08	10.04 ± 0.10	5.06 ± 0.05	6.22 ± 0.07	10.68 ± 0.09	15.48 ± 0.07	8.23 ± 0.10	9.82 ± 0.06
- w/o. HGCN	5.29 ± 0.06	8.47 ± 0.09	4.04 ± 0.10	5.03 ± 0.08	8.74 ± 0.14	13.06 ± 0.11	6.59 ± 0.13	7.87 ± 0.12
- w/o. LRM	6.54 ± 0.08	10.11 ± 0.10	5.11 ± 0.05	6.30 ± 0.06	10.81 ± 0.10	15.59 ± 0.08	8.26 ± 0.11	9.87 ± 0.13
- w/o. Hyper	5.32 ± 0.10	8.60 ± 0.07	4.24 ± 0.08	5.34 ± 0.08	10.28 ± 0.11	14.68 ± 0.12	7.91 ± 0.09	9.33 ± 0.08
Method	Clothing				Book			
	Recall@10	Recall@20	NDCG10	NDCG@20	Recall@10	Recall@20	NDCG@10	NDCG@20
`LogiRec++`	25.26 ± 0.14	28.23 ± 0.12	23.71 ± 0.11	24.73 ± 0.09	5.79 ± 0.07	9.10 ± 0.09	7.27 ± 0.06	8.33 ± 0.08
- w/o. $\mathcal{L}_{Mem}$	22.05 ± 0.10	24.52 ± 0.09	20.64 ± 0.13	21.72 ± 0.08	5.12 ± 0.08	8.20 ± 0.13	6.64 ± 0.10	7.67 ± 0.08
- w/o. $\mathcal{L}_{Hie}$	23.54 ± 0.13	26.68 ± 0.12	21.87 ± 0.09	23.26 ± 0.10	5.32 ± 0.11	8.66 ± 0.10	6.85 ± 0.07	7.93 ± 0.11
- w/o. $\mathcal{L}_{Ex}$	24.12 ± 0.11	27.36 ± 0.10	22.76 ± 0.08	23.98 ± 0.12	5.41 ± 0.12	8.74 ± 0.08	6.92 ± 0.13	8.01 ± 0.10
- w/o. HGCN	17.85 ± 0.13	21.09 ± 0.11	16.07 ± 0.12	17.21 ± 0.10	4.57 ± 0.12	7.52 ± 0.10	5.68 ± 0.09	6.53 ± 0.12
- w/o. LRM	24.53 ± 0.12	27.62 ± 0.14	23.17 ± 0.10	24.04 ± 0.11	5.49 ± 0.10	8.83 ± 0.09	7.04 ± 0.07	8.08 ± 0.08
- w/o. Hyper	23.10 ± 0.11	25.46 ± 0.13	21.07 ± 0.10	22.13± 0.09	5.36 ± 0.13	8.44 ± 0.12	6.79 ± 0.08	7.76 ± 0.10

items and tags in hyperbolic space. Moreover, the proposed `LogiRec++` refines logical relations based on user behaviors, and thus makes full use of tags for accurate and fine-grained user preferences, where `LogiRec++` can outperform AGCN ranging from 4.41% in NDCG@10 on Ciao to 25.05% in Recall@10 on Book.

C. Model Ablation (RQ2)

To study the effectiveness of components, we compare 6 `LogiRec++` variants on four datasets, which can be divided into six components as follows:

- `LogiRec++` w/o. $\mathcal{L}_{Mem}$ is our proposed `LogiRec++` without membership relation modeling.
- `LogiRec++` w/o. $\mathcal{L}_{Hie}$ is our proposed `LogiRec++` without hierarchical relation modeling.
- `LogiRec++` w/o. $\mathcal{L}_{Ex}$ is our proposed `LogiRec++` without exclusive relation modeling.
- `LogiRec++` w/o. HGCN is our proposed `LogiRec++` without the hyperbolic graph convolutional network.
- `LogiRec++` w/o. LRM is our proposed `LogiRec++` without logical relation mining based on user behaviors (i.e, `LogiRec`).
- `LogiRec++` w/o. Hyper is our proposed `LogiRec++` projected from hyperbolic to Euclidean space.

From Table III, we have the following observations:

Compared with `LogiRec++` w/o. logical relation modeling including $\mathcal{L}_{Mem}$, $\mathcal{L}_{Hie}$, and $\mathcal{L}_{Ex}$, `LogiRec++` leads to the performance gains ranging from 2.24% (achieved in NDCG@20 on CD) to 16.81% (achieved in Recall@10 on Ciao). Such results are consistent with those in Table II, showing the effectiveness of modeling logical relations among items and tags. Since the logical relations between tags can be inaccurate and coarse, we observe that the impact of removing exclusion loss is marginal, where `LogiRec++` w/o. $\mathcal{L}_{Ex}$ achieves the best performance among all the three variants that remove different logical relations.

Furthermore, the impact of removing the Hyperbolic GCN is the greatest compared with other ablations, where the performance gains of `LogiRec++` over `LogiRec++` w/o. HGCN ranges from 21.01% (achieved in Recall@20 on Book) to 47.54% (achieved in NDCG@10 on Clothing). This indicates that higher-order relations are helpful to learn complex relations among users, items, and tags.

TABLE IV: Hyperparameter studies (%) on CD and Clothing.

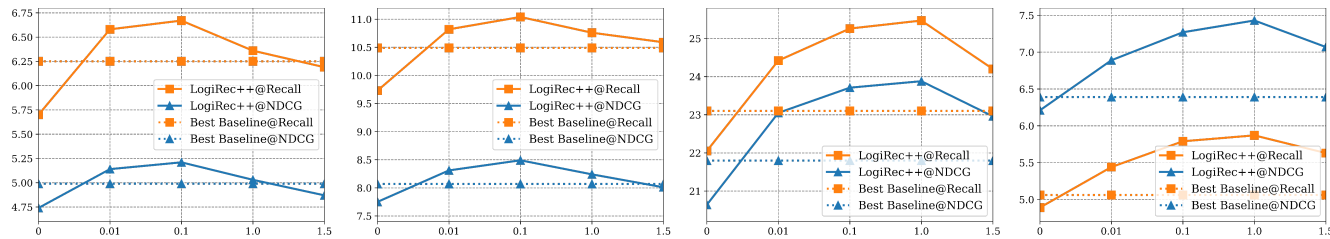
Param.	Recall@10	NDCG@10	Recall@10	NDCG@10
	CD		Clothing	
L = 1	10.29±0.11	7.94±0.10	22.85±0.13	21.86±0.10
L = 2	10.58±0.12	8.27±0.11	24.20±0.13	22.96±0.12
L = 3	**11.04±0.12**	**8.49±0.09**	**25.26±0.14**	**23.71±0.11**
L = 4	10.63±0.14	8.30±0.10	24.03±0.15	23.23±0.13
λ = 0.0	9.73±0.12	7.75±0.11	22.05±0.12	20.64±0.09
λ = 0.01	10.82±0.09	8.31±0.08	24.42±0.11	23.05±0.10
λ = 0.1	**11.04±0.12**	**8.49±0.09**	25.26±0.14	23.71±0.11
λ = 1.0	10.76±0.13	8.24±0.12	**25.47±0.10**	**23.88±0.13**
λ = 1.5	10.59±0.10	8.01±0.09	24.20±0.13	22.96±0.11
m = 0.0	10.81±0.10	8.32±0.11	23.94±0.12	21.82±0.14
m = 0.1	**11.04±0.12**	**8.49±0.09**	**25.26±0.14**	**23.71±0.11**
m = 0.2	10.89±0.10	8.41±0.13	24.87±0.11	23.46±0.08
m = 0.3	10.78±0.11	8.35±0.09	23.80±0.12	21.63±0.12
d = 32	7.86±0.09	5.74±0.10	16.92±0.13	15.60±0.11
d = 64	11.04±0.12	8.49±0.09	25.26±0.14	23.71±0.11
d = 128	**12.38±0.12**	**9.64±0.08**	**26.73±0.10**	**25.59±0.08**

Moreover, since inaccurate logical relations can affect `LogiRec++` w/o. LRM (i.e., `LogiRec`), `LogiRec++` overperforms `LogiRec++` w/o. LRM by refining logical relations without additional supervision. Specifically, `LogiRec++` measures user consistency and granularity only based on the exclusive relations and the learned embedding, which can effectively improve performance ranging from 1.43% in NDCG@20 on Ciao to 5.46% in Recall@10 on Book.

Compared with `LogiRec++` w/o. Hyper, `LogiRec++` leads to significant performance gains ranging from 7.07% in NDCG@10 on Book to 25.38% in Recall@10 on Ciao. Such observation strongly indicates that metric learning in hyperbolic space can effectively and efficiently model the logical relations via Poincaré hyperplanes, which is consistent with those in Table II.

D. Effect of Hyperparameters (RQ3)

Our proposed `LogiRec++` framework mainly introduces three hyperparameters, i.e., L, λ and m. Here we show how these three hyperparameters impact the performance and clarify how to set them.



(a) Varying λ on Ciao (b) Varying λ on CD (c) Varying λ on Clothing (d) Varying λ on Book

Fig. 6: Performance regarding Recall@10 and NDCG@10 of the best baseline and `LogiRec++` with varying weights on logical relation modeling regularizer on the four datasets.

TABLE V: Examples of tag-based user profiles modeled by the proposed `LogiRec++` with user consistency **CON**, user granularity **GR**, and personalized weights $\boldsymbol{\alpha}$. The corresponding items are also recommended by our `LogiRec++`.

	User		Tag	Item
CD	User1	$\text{CON}_1 = 0.91$ $\text{GR}_1 = 0.83$ $\alpha_1 = 0.87$	*<Alternative Rock>*; *<Goth & Industrial>*; *<Industrial>*; *<Industrial Dance>*; ...	Industrial Dance; Industrial Strength Machine Music: 1978-1995; Gothic Industrial: The Remixed Collection; Metal Dance 2: Industrial New Wave & EBM Classics; ...
CD	User2	$\text{CON}_2 = 0.25$ $\text{GR}_2 = 0.39$ $\alpha_2 = 0.31$	*<Pop>*; *<Jazz>*; *<Latin Music>*; *<Rock>*; ...	Best of the Cardigans; Long Gone Before Daylight; Gran Turismo; Other Side of the Moon; Life; First Band On The Moon; Gran Turismo Overdrive; Super Extra Gravity; ...
Book	User3	$\text{CON}_3 = 0.84$ $\text{GR}_3 = 0.78$ $\alpha_3 = 0.81$	*<Teen & Young Adult>*; *<Romance>*; *<Romantic Comedy>*; ...	First Comes Revenge; Best Frenemies; Things We Left Behind (Knockemout Book 3); Wildfire: A Novel (The Maple Hills Series); The Way I Hate Him; Happy Place; ...
Book	User4	$\text{CON}_4 = 0.75$ $\text{GR}_4 = 0.46$ $\alpha_4 = 0.59$	*<Teen & Young Adult>*; *<Romance>*; *<Romantic Suspense>*; *<Fantasy Romance>*; *<Romantic Comedy>*; ...	Never Fall For The Fake Boyfriend: A Grumpy Sunshine Romance (Never Say Never Book 3); Hands Off: A Steamy Romantic Suspense; The Second Chance Shoppe; ...

From Table IV, we have the following observations:

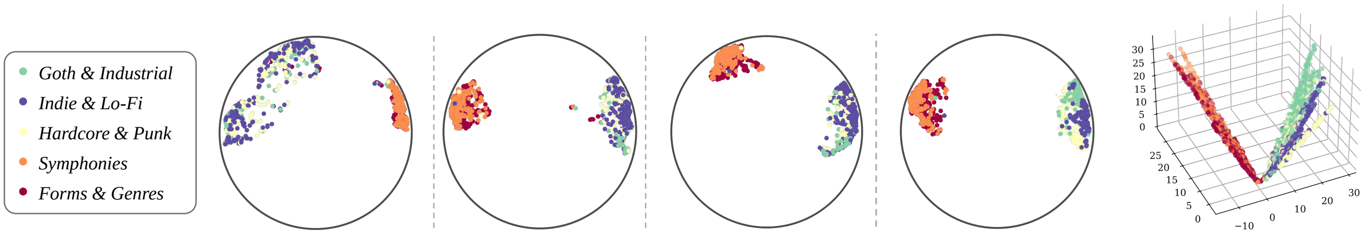
- L is the layer of hyperbolic GCN. `LogiRec++` achieves the best performance with $L = 3$. Since both CD and Clothing have sparse interactions with 0.0769% and 0.0558% density, more neighbor aggregation can be helpful to alleviate the data sparsity issue. When L continues to increase to 4, too many neighbors can lead to an over-smoothing problem.
- λ is the weight of controlling the regularization for $\mathcal{L}_{Mem}$, $\mathcal{L}_{Hie}$, and $\mathcal{L}_{Ex}$. Too small λ will cause inadequate modeling logical relations and cause the model to ignore the inherent logical relations among tags, while too large λ will likely ignore the complex interactions between users and items and cause a decrease in the performance of item recommendation. As shown in Fig. 6, we can observe that the optimal λ values on Ciao, CD, Clothing and Book are about 0.1, 0.1, 1.0 and 1.0, respectively. The choice of λ should depend on the number of tags and logical relations numbers on the dataset. Particularly, for the dataset that has many tags and logical relations (e.g., Clothing), $\lambda = 1.0$ should be chosen. Whereas, $\lambda = 0.1$ should be chosen (e.g., CD). The same can be done for a new dataset.
- m is the margin to enforce the difference between positive and negative triplets. The optimal m value on CD and Clothing is about 0.1, and m can be obtained by slight tuning, which is consistent with [40].
- d is the embedding dimension. Overall, we can observe that `LogiRec++` achieves an average of 47.41% improvement on CD and Clothing when d increases from 32 to 64; whereas, as d increases to 128, `LogiRec++` shows a modest average gain of 9.85%. Since a larger embedding dimension d may cause more storage costs and waste the high-dimensional representation capability of hyperbolic space [40], we set $d = 64$ as a balanced choice.

E. Interpretable Case Studies (RQ4)

To provide more insights into the advantages of `LogiRec++` in providing interpretable recommendations, we demonstrate four example users on CD and Book. All of the recommended tags and the corresponding items are retrieved based on the user-tag relations and user-item interactions, which are learned by `LogiRec++`.

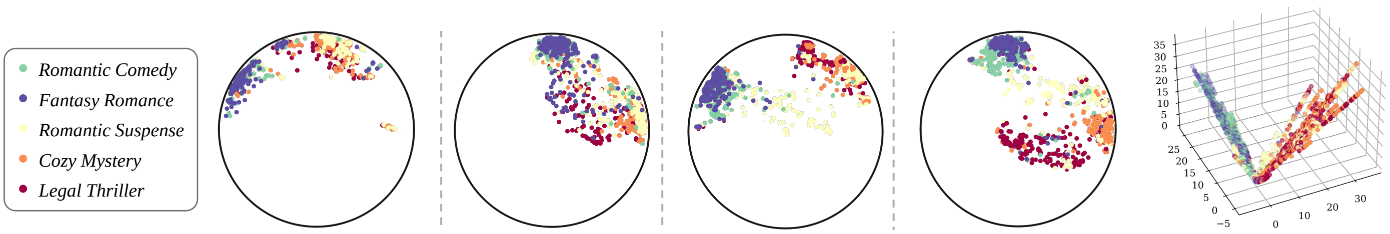
Mining User Consistency for Accurate Recommendation. As shown in Table V, we can perform accurate recommendations via user consistency **CON**, where the higher value of CON, and the item included in more specific tags is recommended. For example, User 1's preference is consistent with tag *<Industrial Dance>* from historical interactions, where user consistency CON_1 is 0.91. As a consequence, we profile User 1 via *<Industrial Dance>*, and the recommendations are highly related, such as Metal Dance 2: Industrial New Wave & EBM Classics. However, User 2 that interacts with diverse CDs is with a low value of CON_2, where we reduce the optimization weight α_2 to avoid modeling inaccurate logical relations derived from User 2. In this way, we pay more attention to his interactions that are mostly from the band The Cardigans with diverse musical styles, and it is reasonable that we recommended the CDs of The Cardigans (e.g., Super Extra Gravity) to User 2.

Mining User Granularity for Fine-grained Recommendation. As shown in Table V, when user consistency of User 3 and User 4 are relatively close, we can further leverage user



(a) AGCN (b) HRCF (c) LogiRec (d) LogiRec++

Fig. 7: Visualizations of item embeddings learned by AGCN, HRCF, LogiRec, and LogiRec++ on CD.



(a) AGCN (b) HRCF (c) LogiRec (d) LogiRec++

Fig. 8: Visualizations of item embeddings learned by AGCN, HRCF, LogiRec, and LogiRec++ on Book.

granularity **GR** for distinguishing user preferences, so as to achieve fine-grained recommendation. Specifically, the value of GR_3 is higher than GR_4, indicating that the preference of User 3 is more specific than User 4 and needs more effort α to make the interacted tags closer. Therefore, we profile User 3 with a fine-grained tag <*Romantic Comedy*> and recommend items such as Best Frenemies and Happy Place to him/her; while we profile User 4 with a coarse-grained tag <*Romance*> and make recommendations from <*Romantic Suspense*>, <*Fantasy Romance*>, or <*Romantic Comedy*>.

Refining Logical Relations for Interpretable Recommendation. To demonstrate the advantages of logical relation mining for recommendations, we visualize the item embedding vectors on CD and Book learned by AGCN, HRCF, LogiRec, and LogiRec++, respectively (shown in Fig. 7 and Fig 8). The items with the same color spectrum represent that they belong to the same tags. In AGCN, HRCF, LogiRec, and LogiRec++, the items from more exclusive tag pairs are well separated (e.g., <*Forms & Genres*> and <*Hardcore & Punk*> on CD, <*Fantasy Romance*> and <*Cozy Mystery*> on Book). However, the separation becomes challenging in the embedding spaces of AGCN, HRCF, and LogiRec when dealing with less exclusive tag pairs (e.g., <*Goth & Industrial*>, <*Indie & Lo-Fi*>, and <*Hardcore & Punk*> on CD, <*Romantic Suspense*>, <*Cozy Mystery*>, and <*Legal Thriller*> on Book). With our proposed logical relation mining, the items from less exclusive tag pairs are also well separated in the embedding spaces of LogiRec++.

Moreover, the tag relation after refinement can also well retain the originally accurate relations. For example, we can observe that <*Romantic Suspense*> learned by LogiRec and LogiRec++ locates in the middle of romantic books and mystery books, which is consistent with the hierarchical relations that <*Romantic Suspense*> belongs to both <*Romance*> and <*Mystery*>. However, this distinctive positioning isn't observed in the embedding spaces of AGCN and HRCF. Finally, implement a hyperbolic projection into a 3D plane to decompose the embeddings learned by LogiRec++. This offers a clearer delineation among items from different tags on the CD and Book datasets (as shown in the right-hand part of Fig.7(d) and Fig.8(d)). The combined use of 2D and 3D visualizations enhances our understanding of the hyperbolic embedding space. It clearly illustrates LogiRec++'s unique strength in refining logical relations among tags by alleviating the bias from data projection. The two visual perspectives together underscore the potential of LogiRec++ to improve the organization of tags.

VII. Conclusion

In this paper, we propose to enhance recommendations via explicitly logical relation modeling and mining in hyperbolic space, which can effectively alleviate the sparsity problem and provide interpretable prediction results. Specifically, by exploiting the individual strengths of two hyperbolic models, we leverage both points and hyperplanes to learn fine-grained representations for users, items, and tags. Then, we achieve logical relation refinement based on user behaviors via innovative consistency-based and granularity-based weighting mechanisms. Extensive experiments demonstrate the clear improvements of LogiRec++ over the state-of-the-art baselines and insightful case studies show the accuracy and interpretability of our logical relation mining.

In the future, it would be interesting to further consider more complicated set-theoretic logical relations (e.g., intersection) from geometric insight, the linguistic information (e.g., tag's name) from semantic insight, and the application of fine-grained user-item-tag relations for tasks such as accurate user profiling and personalized recommendation.

VIII. Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 6230071268, U21A20472, and 62276065, and the National Key Research and Development Plan of China under Grant 2021YFB3600503. Carl Yang was not supported by any fund from China.

References

[1] R. Aly, S. Acharya, A. Ossa, A. Köhn, C. Biemann, and A. Panchenko. Every child should have parents: A taxonomy refinement algorithm based on hyperbolic term embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4811–4817, 2019.

[2] I. Arous, L. Dolamic, and P. Cudré-Mauroux. Taxocomplete: Self-supervised taxonomy completion leveraging position-enhanced semantic matching. In *Proceedings of the ACM Web Conference 2023*, pages 2509–2518, 2023.

[3] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[4] J. Bose, A. Smofsky, R. Liao, P. Panangaden, and W. Hamilton. Latent variable modelling with hyperbolic normalizing flows. In *International Conference on Machine Learning*, pages 1045–1055, 2020.

[5] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, 2020.

[6] C. Chen, X. Zheng, Y. Wang, F. Hong, and D. Chen. Capturing semantic correlation for item recommendation in tagging systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[7] Y. Chen, M. Yang, Y. Zhang, M. Zhao, Z. Meng, J. Hao, and I. King. Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 94–102, 2022.

[8] J. Dai, Y. Wu, Z. Gao, and Y. 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.

[9] S. Feng, L. V. Tran, G. Cong, L. Chen, J. Li, and F. Li. Hme: A hyperbolic metric embedding approach for next-poi recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1429–1438, 2020.

[10] C. Gulcehre, M. Denil, M. Malinowski, A. Razavi, R. Pascanu, K. M. Hermann, P. Battaglia, V. Bapst, D. Raposo, A. Santoro, et al. Hyperbolic attention networks. In *International Conference on Learning Representations*, 2018.

[11] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[12] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[13] Y. Hou, N. Yang, Y. Wu, and P. S. Yu. Explainable recommendation with fusion of aspect information. *World Wide Web*, 22:221–240, 2019.

[14] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, pages 193–201, 2017.

[15] J. Huang, Z. Ren, W. X. Zhao, G. He, J.-R. Wen, and D. Dong. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *WSDM*, 2019.

[16] B. Hui and W.-S. Ku. Low-rank nonnegative tensor decomposition in hyperbolic space. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 646–654, 2022.

[17] R. G. Iyer, Y. Bai, W. Wang, and Y. Sun. Dual-geometric space embedding model for two-view knowledge graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 676–686, 2022.

[18] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020.

[19] W. Krichene and S. Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1748–1757, 2020.

[20] M. Law, R. Liao, J. Snell, and R. Zemel. Lorentzian distance learning for hyperbolic representations. In *International Conference on Machine Learning*, pages 3672–3681, 2019.

[21] M. Li, S. Zhang, F. Zhu, W. Qian, L. Zang, J. Han, and S. Hu. Symmetric metric learning with adaptive margin for recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4634–4641, 2020.

[22] T. Li, Y. Ma, J. Xu, B. Stenger, C. Liu, and Y. Hirate. Deep heterogeneous autoencoders for collaborative filtering. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1164–1169, 2018.

[23] B. Liu, W. Guo, D. Niu, C. Wang, S. Xu, J. Lin, K. Lai, and Y. Xu. A user-centered concept mining system for query and document understanding at tencent. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1831–1841, 2019.

[24] W. Liu, J. Su, C. Chen, and X. Zheng. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. *Advances in Neural Information Processing Systems*, 34:19223–19234, 2021.

[25] Y. Liu, X. Xia, L. Chen, X. He, C. Yang, and Z. Zheng. Certifiable robustness to discrete adversarial perturbations for factorization machines. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 419–428, 2020.

[26] X. Lv, L. Hou, J. Li, and Z. Liu. Differentiating concepts and instances for knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1979, 2018.

[27] W. Meng, D. Yang, and Y. Xiao. Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in Information Retrieval*, pages 1091–1100, 2020.

[28] L. Mirvakhabova, E. Frolov, V. Khrulkov, I. Oseledets, and A. Tuzhilin. Performance of hyperbolic geometry models on top-n recommendation tasks. In *RecSys*, 2020.

[29] F. Mirzazadeh, S. Ravanbakhsh, N. Ding, and D. Schuurmans. Embedding inference for structured multilabel prediction. *Advances in Neural Information Processing Systems*, 28, 2015.

[30] M. Nickel and D. Kiela. Poincar\'e embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

[31] M. Nickel and D. Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, pages 3779–3788, 2018.

[32] C. Park, D. Kim, X. Xie, and H. Yu. Collaborative translational metric learning. In *2018 IEEE international conference on data mining (ICDM)*, pages 367–376, 2018.

[33] J. Park, B. J. Hescott, and D. K. Slonim. Towards a more molecular taxonomy of disease. *Journal of biomedical semantics*, 2017.

[34] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[35] F. Sala, C. De Sa, A. Gu, and C. Ré. Representation tradeoffs for hyperbolic embeddings. In *ICML*, 2018.

[36] R. Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366, 2011.

[37] J. Shang, X. Zhang, L. Liu, S. Li, and J. Han. Nettaxo: Automated topic taxonomy construction from text-rich network. In *Proceedings of The Web Conference 2020*, pages 1908–1919, 2020.

[38] J. Shuai, K. Zhang, L. Wu, P. Sun, R. Hong, M. Wang, and Y. Li. A review-aware graph contrastive learning framework for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1283–1293, 2022.

[39] K. Song, F. Nie, J. Han, and X. Li. Parameter free large margin nearest neighbor for distance metric learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[40] J. Sun, Z. Cheng, S. Zuberi, F. Pérez, and M. Volkovs. Hgcf: Hyperbolic graph convolution networks for collaborative filtering. In *Proceedings of the Web Conference 2021*, pages 593–601, 2021.

[41] L. Sun, Z. Zhang, J. Zhang, F. Wang, H. Peng, S. Su, and S. Y. Philip. Hyperbolic variational graph neural network for modeling dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4375–4383, 2021.

[42] C.-Y. Tai, M.-R. Wu, Y.-W. Chu, S.-Y. Chu, and L.-W. Ku. Mvin: Learning multiview items for recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 99–108, 2020.

[43] K. Takeoka, K. Akimoto, and M. Oyamada. Low-resource taxonomy enrichment with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2747–2758, 2021.

[44] Y. Tan, C. Yang, X. Wei, C. Chen, L. Li, and X. Zheng. Enhancing recommendation with automated tag taxonomy construction in hyperbolic space. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1180–1192, 2022.

[45] Y. Tan, C. Yang, X. Wei, Y. Ma, and X. Zheng. Multi-facet recommender networks with spherical optimization. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1524–1535, 2021.

[46] Y. Tian, Y. Yang, X. Ren, P. Wang, F. Wu, Q. Wang, and C. Li. Joint knowledge pruning and recurrent graph convolution for news recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 51–60, 2021.

[47] J. Van Heijenoort. Set-theoretic semantics. In *Studies in Logic and the Foundations of Mathematics*, volume 87, pages 183–190. Elsevier, 1977.

[48] L. Vinh Tran, Y. Tay, S. Zhang, G. Cong, and X. Li. Hyperml: A boosting metric learning approach in hyperbolic space for recommender systems. In *Proceedings of the 13th international conference on web search and data mining*, pages 609–617, 2020.

[49] H. Wang, D. Lian, H. Tong, Q. Liu, Z. Huang, and E. Chen. Hypersorec: Exploiting hyperbolic user and item representations with multiple aspects for social-aware recommendation. *ACM Transactions on Information Systems (TOIS)*, 40(2):1–28, 2021.

[50] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua. Kgat: Knowledge graph attention network for recommendation. In *KDD*, 2019.

[51] Y. Wang, H. Wang, W. Lu, and Y. Yan. Hygge: Hyperbolic graph attention network for reasoning over knowledge graphs. *Information Sciences*, 630:190–205, 2023.

[52] R. F. Woolson. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3, 2007.

[53] L. Wu, Y. Yang, K. Zhang, R. Hong, Y. Fu, and M. Wang. Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 679–688, 2020.

[54] B. Xiong, M. Cochez, M. Nayyeri, and S. Staab. Hyperbolic embedding inference for structured multi-label prediction. In *Advances in Neural Information Processing Systems*, 2022.

[55] Y. Xu, D. Wang, B. Chen, R. Lu, Z. Duan, M. Zhou, et al. Hyperminer: Topic taxonomy mining with hyperbolic embedding. *Advances in Neural Information Processing Systems*, 35:31557–31570, 2022.

[56] H. Xuan, Y. Liu, B. Li, and H. Yin. Knowledge enhancement for contrastive multi-behavior recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 195–203, 2023.

[57] M. Yang, M. Zhou, J. Liu, D. Lian, and I. King. Hrcf: Enhancing collaborative filtering via hyperbolic geometric regularization. In *Proceedings of the ACM Web Conference 2022*, pages 2462–2471, 2022.

[58] Y. Yang, C. Huang, L. Xia, and C. Huang. Knowledge graph self-supervised rationalization for recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3046–3056, 2023.

[59] C. Zhang and J. Gao. Hype-han: Hyperbolic hierarchical attention network for semantic embedding. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3990–3996, 2021.

[60] K. Zhang, H. Qian, Q. Cui, Q. Liu, L. Li, J. Zhou, J. Ma, and E. Chen. Multi-interactive attention network for fine-grained feature learning in ctr prediction. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 984–992, 2021.

[61] Y. Zhang, A. Ahmed, V. Josifovski, and A. Smola. Taxonomy discovery for personalized recommendation. In *WSDM*, 2014.

[62] Y. Zhang, C. Li, X. Xie, X. Wang, C. Shi, Y. Liu, H. Sun, L. Zhang, W. Deng, and Q. Zhang. Geometric disentangled collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 80–90, 2022.

[63] F. Zhao and Y. Guo. Learning discriminative recommendation systems with side information. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3469–3475, 2017.

[64] X. Zheng, M. Wang, C. Chen, Y. Wang, and Z. Cheng. Explore: Explainable item-tag co-recommendation. *Information Sciences*, 474:170–186, 2019.

[65] Y. Zheng, C. Gao, L. Chen, D. Jin, and Y. Li. Dgcn: Diversified recommendation with graph convolutional networks. In *Proceedings of the Web Conference 2021*, pages 401–412, 2021.

[66] Y. Zhou, Y. Cao, Y. Shang, C. Zhou, S. Pan, Z. Lin, and Q. Li. Explainable hyperbolic temporal point process for user-item interaction sequence generation. *ACM Transactions on Information Systems*, 41(4):1–26, 2023.

[67] C.-N. Ziegler, G. Lausen, and L. Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *CIKM*, 2004.