2020 IEEE International Conference on Data Mining (ICDM)

PERFECT: A Hyperbolic Embedding for Joint User and Community Alignment

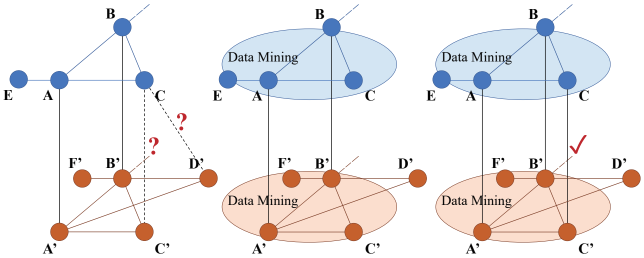
Li Sun[1], Zhongbao Zhang[1†], Jiawei Zhang[2], Feiyang Wang[1], Yang Du[1], Sen Su[1], and Philip S. Yu[3]
[1]*School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China*
[2]*IFM Lab, Department of Computer Science, Florida State University, FL, USA*
[3]*Department of Computer Science, University of Illinois at Chicago, IL, USA*
{l.sun, zhongbaozb, fywang, duyang, susen}@bupt.edu.cn, jiawei@ifmlab.org, psyu@uic.edu

***Abstract*—Social network alignment shows fundamental importance in a wide spectrum of applications. To the best of our knowledge, existing studies mainly focus on network alignment at the individual user level, requiring abundant common information between shared individual users. For the networks that cannot meet such requirements, social community structures actually provide complementary and critical information at a slightly coarse-grained level, alignment of which will provide additional information for user alignment. In turn, user alignment also reveals more clues for community alignment. Hence, in this paper, we introduce the problem of *joint social network alignment*, which aims to align users and communities across social networks simultaneously. Key challenges lie in that (1) how to learn the representations of both users and communities, and (2) how to make user alignment and community alignment benefit from each other. To address these challenges, we first elaborate on the characteristics of real-world networks with the notion of $\delta-$*hyperbolicity*, and show the superiority of hyperbolic space for representing social networks. Then, we present a novel hyperbolic embedding approach for the joint social network alignment, referred to as PERFECT, in a unified optimization. Extensive experiments on real-world datasets show the superiority of PERFECT in both user alignment and community alignment.**

***Index Terms*—Network Embedding; Network Alignment; Social Network; Data Mining**

I. INTRODUCTION

Nowadays, people join in multiple social networks to enjoy more diverse services. The alignment across these social networks benefits a wide range of applications, such as link prediction and information diffusion [1], and thus receives an increasing attention [2], [3], [4], [5]. To the best of our knowledge, existing studies mainly focus on the alignment at the individual user level, requiring abundant common information between shared individual users. However, for the networks that cannot meet such requirements, community structure plays an important role in understanding users' social patterns. Community alignment enriches the information across networks especially when individual users don't have enough common information for alignment. As illustrated in Fig. 1, compared to (a), we collect more aligned users in (c) with the additional knowledge of community alignment in (b). Furthermore, user alignment naturally reveals more clues for inferring community alignment, as shown in Fig. 1 (a) and (b). That is, user alignment and community alignment are strongly correlated. Hence, we rethink that: *can we jointly align users and communities across different social networks?*



(a) User Alignment ⟶ (b) Community Alignment ⟶ (c) User Alignment

Fig. 1. Joint social network alignment: Different networks are shown in different colors. Black lines link aligned users. Community members are grouped in the disk. Community alignment helps user alignment. It is impossible to distinguish the counterpart of C between C' and D' with network structure in (a). Further knowing both C between C' are users of data mining community in (b), C is more likely to be aligned with C' rather than D' in (c). In turn, aligned users across communities facilitate community alignment.

To this end, we introduce the problem of *joint social network alignment* in this paper. It is facing following challenges:

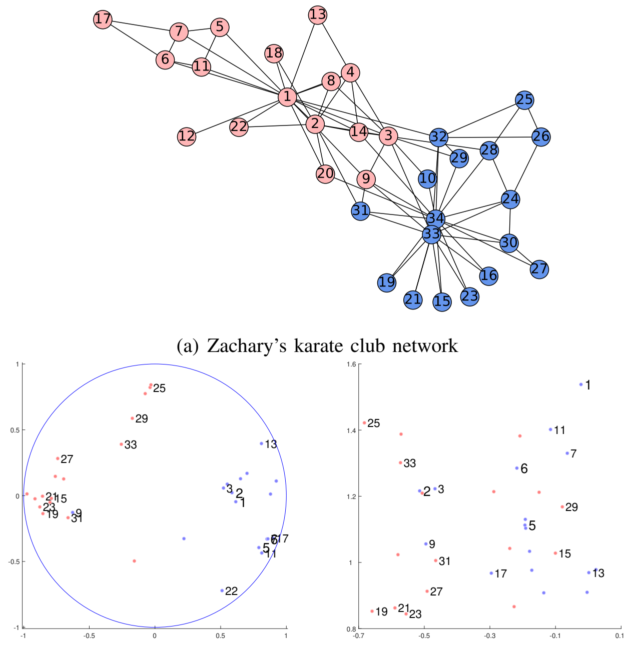
- *How to learn the representations of both users and communities in an appropriate embedding space?* Accurate alignment is possible only if embeddings can capture faithful information. Existing methods for network alignment explicitly or implicitly work with Euclidean space [6], [3], [7]. However, Euclidean space tends to render reconstruction error when embedding real-world social networks [8]. Hence, it calls for a promising embedding space for both users and communities.
- *How to make user alignment and community alignment benefit from each other?* To our knowledge, user alignment is widely studied while community alignment has rarely been touched before. Though user alignment and community alignment are strongly correlated as shown in Fig. 1, it still remains open to make user alignment and community alignment benefit from each other.

To address these challenges, in this paper, we propose a novel unified hyPERbolic embedding approach For the joint usEr and Community alignmenT, referred to as PERFECT. Its essential novelty lies in that we for the first time close the loop of user alignment and community alignment so that they benefit from each other in a unified optimization.

To address the first challenge mentioned above, we work with hyperbolic space. To elaborate on the choice of represen-

[†] Corresponding Author

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00059



(a) Zachary's karate club network

(b) Hyperbolic Embedding

(c) Euclidean Embedding

Fig. 2. We embed the Zachary karate club network in (a) via the proposed hyperbolic embedding approach and the corresponding Euclidean one in $2D$ space with the same experimental settings, shown in (b) and (c), respectively. In (b), nodes of high degree, e.g., nodes 1 and 2, reside close to the origin and nodes of low degree, e.g., nodes 5 and 6, are positioned close to the boundary of the disk, revealing the latent hierarchy, while this does not hold in the Euclidean space in (c). Moreover, communities are separable in (b).

tation space, we first give a toy example in Fig. 2: we embed the example network in (a) with both a hyperbolic embedding approach and the corresponding Euclidean one in $2D$ space, whose results are shown in (b) and (c), respectively. Compared against Euclidean space, hyperbolic space tends to present the latent hierarchies among nodes in the input network. We observe that *the hierarchical characteristic* is common for social networks [9] and, more importantly, has shown to be crucial for user alignment [2] and benefits community discovery [10]. Furthermore, we demonstrate the hierarchical characteristic of several real-world networks with the metric of Gromovs *δ-hyperbolicity* [8], [11]. Fortunately, the hyperbolic space is well-suited to embed the latent hierarchical structures [12], [13], [14]. Thus, in PERFECT, we embed both users and communities of each network in hyperbolic space. We then construct a common hyperbolic subspace, and finally formulate the unified optimization to jointly align users and communities across social networks.

Meanwhile, to address the second challenge, we propose an alternating Riemannian optimization algorithm so that user embeddings and community embeddings are mutually refined in the common hyperbolic subspace for the joint alignment. Specifically, we update community embeddings and user embeddings in an alternating approach. When updating community embeddings, we incorporate the knowledge of user embeddings via expectation-maximization in Riemannian manifold for community alignment. On the other hand, when updating user embeddings, we incorporate the knowledge of community embeddings by conducting exponential map with Riemannian gradient for user alignment. Furthermore, we give solid theoretical analyses on the proposed algorithm.

Finally, we summarize the key contributions as follows:

- To our knowledge, this is the first attempt to jointly align users and communities across social networks.
- To this end, we work with hyperbolic space and propose a novel hyperbolic embedding approach with a unified optimization, PERFECT, closing the loop of community alignment and user alignment.
- To address this optimization, in PERFECT, we design a novel Riemannian alternating optimization algorithm with solid theoretical analyses.
- We empirically evaluate the hyperbolicity of several real-world social networks and show the superiority of PERFECT. Our code is available at https://github.com/NetAligner/perfect.

II. Problem Statement

Notations: We use lowercase x, bold lowercase $\boldsymbol{x}$ and bold uppercase $\boldsymbol{X}$ to denote scalar, vector and matrix, respectively. $\boldsymbol{x}^T$ denotes the transpose of $\boldsymbol{x}$. $||\cdot||$ and $\langle\cdot,\cdot\rangle$ denote the usual Euclidean norm and inner product throughout this paper.

A social network is described as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{(v_i)\}$ is the user set of size $N = |\mathcal{V}|$ and $\mathcal{E} = \{(v_i, v_j)\}$ is the edge set. A community is a subset of users $\mathcal{C}_p \subset \mathcal{V}$ with the same community label, where $\mathcal{C}_p \bigcap \mathcal{C}_q = \emptyset$ for any $\mathcal{C}_p$ and $\mathcal{C}_q$, and $\bigcup_p \mathcal{C}_p = \mathcal{V}$. We consider a pair of social networks: $\mathcal{G}^s$ is the source network and $\mathcal{G}^t$ is the target network. We use superscript x to indicate variables associated with $\mathcal{G}^x$, $x \in \{s, t\}$. We use subscripts i, j, k and n to denote indexes of the users, and subscripts p and q to denote indexes of the communities. The source and target networks are linked by anchor users, whose definition is given as follows:

Definition 1 (Anchor User). *The user who has accounts v_i^s in the source network $\mathcal{G}^s$ and v_k^t in the target network $\mathcal{G}^t$ is termed as anchor user, and (v_i^s, v_k^t) is called an anchor link.*

The set of anchor users known in advance between $\mathcal{G}^t$ and $\mathcal{G}^s$ is referred to as $\mathcal{A}$, which can be collected from user profiles or third-party platforms.

Definition 2 (Anchor Community). *Community $\mathcal{C}_p^s$ discovered from $\mathcal{G}^s$ and community $\mathcal{C}_q^t$ discovered from $\mathcal{G}^t$ are said to be anchor community iff at least τ proportion of the users in them are anchor users connecting $\mathcal{C}_p^s$ and $\mathcal{C}_q^t$.*

For instance, between $\mathcal{C}_p^s$ and $\mathcal{C}_q^t$, the anchor links existing among the users in them can be denoted as $\mathcal{A}_{p,q} \subset \mathcal{A}$. Then, we have $\tau = 2|\mathcal{A}_{p,q}|/(|\mathcal{C}_p^s| + |\mathcal{C}_q^t|)$. Now, we formally define the problem of joint social network alignment as follows:

The Problem of Joint Social Network Alignment. *Given a pair of social networks $\mathcal{G}^s$ and $\mathcal{G}^t$ with the anchor user set $\mathcal{A}$, the aim of joint social network alignment is to identify:*

(1) all anchor users $\{(v_i^s, v_k^t)\}$ (i.e., user alignment) and

(2) all anchor communities $\{(\mathcal{C}_p^s, \mathcal{C}_q^t)\}$ (i.e., community alignment) simultaneously between the pair of social networks.

III. Perfect: Model

To address this problem, we propose a novel hyperbolic embedding approach, Perfect, with a unified optimization. We first elaborate on why we work with hyperbolic space, and then embed both users and communities in this embedding space. Finally, we introduce the formulation of the unified optimization so that community embeddings and user embeddings are mutually refined for the joint social network alignment.

A. Why hyperbolic embedding?

Here, we explain why hyperbolic space is a promising embedding space. Recall the example in Fig. 2. Compared against the Euclidean space, embeddings in hyperbolic space encode the latent hierarchy among the users, *i.e.*, users of higher centrality tend to reside closer to the origin. Such *hierarchical characteristic* is of significance.

For real-world graphs, the study [9] elaborates on the formation of their hierarchical characteristic in general. More importantly, the hierarchical characteristic has shown to be crucial for user alignment [2] and benefits community discovery [10]. Additionally, we demonstrate the latent hierarchy on several real-world graphs. In order to measure the hierarchical characteristic quantitatively, we introduce the Gromov $\delta-$*hyperbolicity* [11], [8], a metric from geometric group theory. Note that, a lower value of δ indicates a better hierarchical structure. Specifically, we use the following datasets:

- *Zachary karate dataset*: The example network in Fig. 2.
- *Twitter-Quora dataset*: We collect two friendship networks, Twitter and Quora, linked by common users. We use users' registered affiliations as community labels.
- *DBLP-AMiner dataset*: DBLP and AMiner [2] are two coauthor networks linked by common authors. Authors' research areas denote their community labels.

The statistics are given in Table 1 and the corresponding $\delta-$*hyperbolicity* shows their latent hierarchy.

TABLE I
Statistics and $\delta-$hyperbolicity of the datasets

Network	#(Node)	#(Comm.)	#(Link)	#(Anchor)	δ
Zachary	34	2	78	-	1
Twitter	$19,438$	60	$201,063$	$10,232$	3.5
Quora	$10,638$	60	$46,969$		4
DBLP	$13,211$	12	$46,278$	$12,213$	2.5
AMiner	$13,213$	12	$46,189$		3

Can we incorporate such latent hierarchy? Fortunately, we find that *hyperbolic space* is well suited to embed graphs with latent hierarchical structure [12], [13], [14]. Let's take an extreme example, the tree. We give the fact that the $\delta-$hyperbolicity of a tree is 0. The $\delta-$hyperbolicity of the hyperbolic space (Poincaré ball model in Section III-B) is $\log(1+\sqrt{2})$, while $\delta=\infty$ for Euclidean spaces [11]. It is obvious that hyperbolic space better matches the δ of a tree than the Euclid. Indeed, any tree can be embedded in a 2-dimensional Poincaré ball with arbitrary low reconstruction error [15], while this is not true for Euclidean spaces even when an unbounded dimension is allowed. All these facts motivate us to leverage hyperbolic space as the representation space. Furthermore, we will examine the effects of $\delta-$hyperbolicity with experimental results in Section V-A and V-B.

B. The Poincaré ball Model of Hyperbolic Space

Now, we introduce the preliminaries of hyperbolic space for our work. The hyperbolic space is a kind of isotropic space with constant negative curvature, and there are several models proposed for reasoning in hyperbolic space [16]. We prefer to work with the widely used Poincaré ball model owing to its conformality (angle-preserving *w.r.t.* Euclidean space) and convenient parameterization [12], [13], [14], [17].

The Poincaré ball model of dimension d is formally defined as a smooth manifold $\mathcal{B}^d=\{\boldsymbol{x}\in\mathbb{R}^d|\ ||\boldsymbol{x}||<1\}$ endowed with a Riemannian metric:

$$\mathfrak{g}^{\mathcal{B}}(\boldsymbol{x})=\left(\frac{2}{1-\|\boldsymbol{x}\|^2}\right)^2\mathfrak{g}^E=(\lambda_{\boldsymbol{x}})^2\,\mathfrak{g}^E,\qquad(1)$$

which is a collection of inner product in the tangent space $T_x\mathcal{B}^d$ of $\boldsymbol{x}\in\mathcal{B}^d$. $\lambda_{\boldsymbol{x}}=\frac{2}{1-\|\boldsymbol{x}\|^2}$ is the conformal factor, and $\mathfrak{g}^E$ is the Euclidean metric tensor under usual Cartesian coordinates of $\mathbb{R}^d$. The Poincaré ball is conformal but wrapped (distance curving) *w.r.t.* the Euclidean space. The distance between two points [14], *e.g.*, $\boldsymbol{x},\boldsymbol{y}\in\mathcal{B}^d$, is given as follows:

$$d(\boldsymbol{x},\boldsymbol{y})=\cosh^{-1}\left(1+\frac{2||\boldsymbol{x}-\boldsymbol{y}||^2}{(1-||\boldsymbol{x}||^2)(1-||\boldsymbol{y}||^2)}\right).\qquad(2)$$

We derive its partial derivatives *w.r.t.* $\boldsymbol{y}$ as follows:

$$\frac{\partial d(\boldsymbol{x},\boldsymbol{y})}{\partial\boldsymbol{y}}=\frac{4}{\beta\sqrt{\gamma^2-1}}\left(\frac{\|\boldsymbol{y}\|^2-2\langle\boldsymbol{y},\boldsymbol{x}\rangle+1}{\alpha^2}\boldsymbol{y}-\frac{\boldsymbol{x}}{\alpha}\right),\quad(3)$$

where $\alpha=1-\|\boldsymbol{y}\|^2$, $\beta=1-\|\boldsymbol{x}\|^2$ and $\gamma=1+\frac{2}{\alpha\beta}\|\boldsymbol{y}-\boldsymbol{x}\|^2$.

C. Hyperbolic User Embedding

We learn user embeddings $\boldsymbol{\theta}_i^x$ in hyperbolic space, a Poincaré ball $\mathcal{B}^d$. The basic idea is that, for each graph $\mathcal{G}^x$, $x\in\{s,t\}$, the proximity between users is preserved in the hyperbolic distance between user embeddings.

First, we conduct random walks to extract the proximity between users. In a random walk, the neighborhood $\mathcal{N}_i^x$ of node v_i^x is named as its "context". Intuitively, two nodes sharing more contexts are of higher proximity, and thus have similar embeddings [18]. In this case, each node is treated as a node for itself and a context for some other nodes. Hence, to differentiate the user embedding $\boldsymbol{\theta}_i^x$ for itself, we introduce a context embedding $\boldsymbol{\theta}_i^{x\prime}\in\mathcal{B}^d$ for each node as well.

Then, we leverage the hyperbolic distance to preserve the proximity in $\mathcal{B}^d$. Specifically, we define the probability of having v_j^x as a context of a given v_i^x via the hyperbolic distance in Eq. (2) as follows:

$$Pr(v_j^x|v_i^x)=\sigma[-d(\boldsymbol{\theta}_j^{x\prime},\boldsymbol{\theta}_i^x)],\qquad(4)$$

where $\sigma(x)=\frac{1}{\exp(-x)+1}$ is the sigmoid function. For $\mathcal{G}^s$ and $\mathcal{G}^t$, we minimize the negative log-likelihood as follows:

$$\mathcal{O}_{user}=-\sum_{x\in\{s,t\}}\sum_{v_i^x\in\mathcal{V}^x}\sum_{v_j^x\in\mathcal{N}_i^x}\log Pr(v_j^x|v_i^x),\qquad(5)$$

i.e., we aim to construct the neighborhood of random walks via the hyperbolic distance between user embeddings.

D. Hyperbolic Community Embedding

With hyperbolic user embeddings above, we further learn community embeddings $\boldsymbol{\mu}_p^x$ in this Poincaré ball $\mathcal{B}^d$, where $p \in [1, C^x]$ and C^x is the number of communities in $\mathcal{G}^x$.

Inspired by model-based clustering, we consider that user embeddings $\{\boldsymbol{\theta}_\cdot^x\}$ are drawn from the mixture of multivariate distribution $\{Pr(\cdot|\boldsymbol{\psi}_p^x)\}_{p=1}^{C^x}$ in hyperbolic space, where $\boldsymbol{\psi}_p^x$ is the distribution parameter to be described in details later. Each distribution $Pr(\cdot|\boldsymbol{\psi}_p^x)$ corresponds to a community $\mathcal{C}^p$, and the community embedding $\boldsymbol{\mu}_p^x$ is given as the location of $Pr(\cdot|\boldsymbol{\psi}_p^x)$ in the hyperbolic space. Then, for all user embeddings $\{\boldsymbol{\theta}_\cdot^x\}$ in $\mathcal{G}^x$, we have the likelihood as follows:

$$\prod_{i=1}^{N^x} \sum_{p=1}^{C^x} \boldsymbol{Z}_{ip} Pr(\boldsymbol{\theta}_i^x|\boldsymbol{\psi}_p^x), \tag{6}$$

where $\boldsymbol{Z}_{ip}$ is the probability of user v_i belonging to community $\mathcal{C}^p$, and thus we have $\boldsymbol{Z}_{ip} \in [0,1]$ and $\sum_{p=1}^{C^x} \boldsymbol{Z}_{ip} = 1$.

We leverage (generalized) hyperbolic distribution to model communities in hyperbolic space whose PDF is given as

$$\begin{aligned} Pr_{\mathcal{H}}(\boldsymbol{\theta}; \boldsymbol{\mu}, \Delta, \boldsymbol{\beta}, r, \omega) =& \frac{e^{-\boldsymbol{\beta}^T \Delta^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})}}{(2\pi)^{\frac{d}{2}} |\Delta|^{\frac{1}{2}}} \left(\frac{\omega + \delta_{\boldsymbol{\theta}}}{\omega + \boldsymbol{\beta}^T \Delta^{-1} \boldsymbol{\beta}} \right)^{\frac{r-d/2}{2}} \\ & \frac{K_{r-\frac{d}{2}}\left(\sqrt{(\omega + \boldsymbol{\beta}^T \Delta^{-1} \boldsymbol{\beta})(\omega + \delta_{\boldsymbol{\theta}})} \right)}{K_r(\omega)}, \end{aligned} \tag{7}$$

where $\delta_{\boldsymbol{\theta}} = (\boldsymbol{\theta} - \boldsymbol{\mu})^T \Delta^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu})$. $\boldsymbol{\beta} \in \mathcal{B}^d$ and $\boldsymbol{\mu} \in \mathcal{B}^d$ are skewness and location vectors, respectively. ω is the concentration factor. $\Delta \in \mathbb{R}^{d \times d}$ is the positive definite scatter matrix capturing Riemannian metric, and $|\Delta|$ is its determinant. $K_r(\cdot)$ is the modified Bessel function of $(\cdot)$ with order r. Then, we have $\boldsymbol{\psi}_p^x = (\boldsymbol{\mu}_p^x, \Delta_p^x, \boldsymbol{\beta}_p^x, r_p^x, \omega_p^x)$ for each $x \in \{s, t\}$, and $\boldsymbol{\theta}_i^x \sim \sum_{p=1}^{C^x} \boldsymbol{Z}_{ip}^x Pr_{\mathcal{H}}(\boldsymbol{\theta}_i^x|\boldsymbol{\psi}_p^x)$. Thus, with user embeddings $\{\boldsymbol{\theta}_i^x\}$, we minimize the negative log-likelihood as follows:

$$\mathcal{O}_{community} = - \sum_{x \in \{s,t\}} \sum_{v_i^x \in \mathcal{V}^x} \log \sum_{p=1}^{C^x} \boldsymbol{Z}_{ip}^x Pr_{\mathcal{H}}(\boldsymbol{\theta}_i^x; \boldsymbol{\psi}_p^x), \tag{8}$$

so that we learn the community embedding and users' community membership simultaneously.

E. Hyperbolic Common Subspace

We embed both users and communities for each network. Then, across $\mathcal{G}^s$ and $\mathcal{G}^t$, we construct a common hyperbolic subspace where we can jointly align users and communities.

Specifically, the common hyperbolic subspace is constructed via aligning embedding spaces of $\mathcal{G}^s$ and $\mathcal{G}^t$ on anchor users, so that the user embedding of v_k^t is transfered via anchor link (v_i^s, v_k^t) to predict embeddings $\boldsymbol{\theta}_\cdot^{s\prime}$ in the neighborhood of its counterpart v_i^s. We minimize the negative log-likelihood:

$$\mathcal{O}_{align} = - \sum_{(v_i^s, v_k^t) \in \mathcal{A}} \left(\sum_{v_j^s \in \mathcal{N}_i^s} \log Pr(v_j^s|v_k^t) + \sum_{v_j^t \in \mathcal{N}_k^t} \log Pr(v_j^t|v_i^s) \right). \tag{9}$$

For each anchor user $(v_i^s, v_k^t) \in \mathcal{A}$, the first term is to predict embeddings in the neighborhood $\mathcal{N}_i^s$ of v_i^s in $\mathcal{G}^s$ using $\boldsymbol{\theta}_k^t$ of $\mathcal{G}^t$ and the second term is to predict embeddings in $\mathcal{N}_k^t$ using $\boldsymbol{\theta}_i^s$. In this common subspace, communities as well as users are to be aligned via the hyperbolic distance.

F. Objective Function

Finally, we formulate the unified optimization of PERFECT jointly aligning users and communities in the common hyperbolic subspace. We adopt the negative sampling [19] to define the optimization objective, *i.e.*, we replace the log term in $\mathcal{O}_{user}$ and $\mathcal{O}_{align}$ with the right hand side of the following equation and have $\mathcal{O}_{user}^{\text{NS}}$ and $\mathcal{O}_{align}^{\text{NS}}$.

$$\begin{aligned} \log Pr(v_j^t|v_i^s) \ \propto & \ \log \sigma[-d(\boldsymbol{\theta}_j^{t\prime}, \boldsymbol{\theta}_i^s)] + \\ & \sum_{v_n \in \text{NS}_i^K} \mathbb{E}_{v_n} \left[\log \sigma[d(\boldsymbol{\theta}_n^{t\,\prime}, \boldsymbol{\theta}_i^s)] \right], \end{aligned} \tag{10}$$

where the probability of negative sample v_n being selected is as proposed in the study [19], and NS_i^K is defined as a set of K negative samples being selected randomly, $(v_n, v_i) \notin \mathcal{E}$. Eq. (10) holds for identical superscripts of v_i and v_j as well. Incorporating Eqs. (5), (8) and (9), we finally obtain the objective function of PERFECT:

$$\begin{aligned} \min_{\boldsymbol{\theta}, \boldsymbol{\theta}', \boldsymbol{\psi}, \boldsymbol{Z}} & \mathcal{J}_0 = \mathcal{O}_{user}^{\text{NS}} + \alpha_1 \mathcal{O}_{community} + \alpha_2 \mathcal{O}_{align}^{\text{NS}} \\ \text{s.t.} \quad & \Delta_p^x \succeq 0, p = 1, 2, ..., C^x; \ \sum_{p=1}^{C^x} \boldsymbol{Z}_{ip}^x = 1, \forall x \in \{s, t\}, \end{aligned} \tag{11}$$

where α_1 and α_2 are two nonnegative weight parameters.

In the common subspace induced by optimizing Eq. (11), user embeddings will learn from community embeddings in both networks and, in turn, community embeddings learn from all user embeddings for *the joint social network alignment*.

Algorithm 1: Alternating Riemannian Optimization

```
Input: graph pair $(\mathcal{G}^s, \mathcal{G}^t, \mathcal{A})$, #(community) $C^s$, $C^t$,
       embedding dimension $d$, #(negative sample) $K$
Output: user embeddings $\boldsymbol{\theta}^x$, community embeddings
        $\boldsymbol{\mu}^x$ and membership $\boldsymbol{Z}^x$ for $x \in \{s, t\}$
Conduct random walks for each social network;
Initialize $\boldsymbol{\theta}^x$ and $\boldsymbol{\theta}^{x\prime}$ via $\min \mathcal{O}_{user}^{\text{NS}}$ for each
  $x \in \{s, t\}$;
while not converging do
    forall social network do
        forall community $\mathcal{C}_p^x$ in each $\mathcal{G}^x$ do
            Calculate the expectations in Eq. (12);
            Obtain community membership $\boldsymbol{Z}^x$;
            Update $\{\boldsymbol{\psi}_p^x\}_{p=1}^{C^x}$ via Eq. (13);
        forall node $v_i^x$ in each $\mathcal{G}^x$ do
            Update $\boldsymbol{\theta}_i^x$ via Eqs. (19)-(21) ;
            Update $\boldsymbol{\theta}_i^{x\prime}$ via Eqs. (22)-(23);
```

IV. PERFECT: OPTIMIZATION

To address the optimization in Eq. (11) , we first decompose this optimization problem into two subproblems, *i.e.*, community embedding subproblem *w.r.t.* $(\boldsymbol{\psi}, \boldsymbol{Z})$ and user embedding subproblem *w.r.t.* $(\boldsymbol{\theta}, \boldsymbol{\theta}')$. Then, we propose an alternating Riemannian optimization algorithm. The main idea is that we alternatively optimize one subproblem while fixing the parameters of the other. The overall process is summarized in Algorithm 1. In each iteration (Line 4-10), community embeddings and user embeddings are mutually refined in the common hyperbolic subspace for the joint alignment. Every time we obtain the embeddings, the alignment is naturally revealed in the hyperbolic distance. We finally output the joint alignment results together until they cannot refine each other.

A. Community Embedding Subproblem

Given user embeddings, we update community embeddings for community alignment. To facilitate the optimization, we first introduce auxiliary random variables of inverse Gaussian $W \sim \mathcal{I}(\omega, 1, r)$, $W \in \mathbb{R}$, and Gaussian $\boldsymbol{g} \sim \mathcal{N}(\mathbf{0}, \Delta)$, $\boldsymbol{g} \in \mathbb{R}^d$. Then, a random variable $\boldsymbol{\theta}$ of the hyperbolic distribution with parameter $\boldsymbol{\psi} = (\boldsymbol{\mu}, \Delta, \boldsymbol{\beta}, r, \omega)$ is equal to the combination $\boldsymbol{\mu} + W\boldsymbol{\beta} + \sqrt{W}\boldsymbol{g}$ [20]. Then, under the reformulation of this combination, we optimize $(\boldsymbol{Z}^x, \{\boldsymbol{\psi}_p^x\})$ via expectation-maximization algorithm. Take $\mathcal{G}^s$ for instance.

In the expectation-step, we require following expectations:

$$\begin{aligned}
z_{ip}^s &= \mathbb{E}\left[\boldsymbol{Z}_{ip}^s | \boldsymbol{\theta}_i^s\right] = \frac{\boldsymbol{Z}_p^s Pr_{\mathcal{H}}\left(\boldsymbol{\theta}_i^s | \boldsymbol{\psi}_p^s\right)}{\sum_{p=1}^{C^s} \boldsymbol{Z}_p^s Pr_{\mathcal{H}}\left(\boldsymbol{\theta}_i^s | \boldsymbol{\psi}_p^s\right)}, \\
a_{ip}^s &= \mathbb{E}\left[W_{ip}^s | \boldsymbol{\theta}_i^s, \boldsymbol{Z}_{ip}^s = 1\right] = \frac{K_{r_p^s+1}(\omega_p^s)}{K_{r_p^s}(\omega_p^s)} - \frac{2r_p^s}{\omega_p^s}, \\
b_{ip}^s &= \mathbb{E}\left[1/W_{ip}^s | \boldsymbol{\theta}_i^s, \boldsymbol{Z}_{ip}^s = 1\right] = \frac{K_{r_p^s+1}(\omega_p^s)}{K_{r_p^s}(\omega_p^s)}, \\
c_{ip}^s &= \mathbb{E}\left[\log(W_{ip}^s) | \boldsymbol{\theta}_i^s, \boldsymbol{Z}_{ip}^s = 1\right] = \frac{1}{K_{r_p^s}(\omega_p^s)} \frac{\partial K_{r_p^s}(\omega_p^s)}{\partial r_p^s},
\end{aligned} \tag{12}$$

where $\boldsymbol{Z}_{ip}^s$ is the membership of i^{th} user to p^{th} community. Then, we define: $n_p^s = \sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s$, $\overline{a}_p^s = \frac{1}{C^s} \sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s a_{ip}^s$, $\overline{b}_p^s = \frac{1}{C^s} \sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s b_{ip}^s$ and $\overline{c}_p^s = \frac{1}{C^s} \sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s c_{ip}^s$.

In the maximization-step, we derive the updating rules of hyperbolic distribution parameters as follows:

$$\begin{aligned}
\boldsymbol{\mu}_p^s &= \frac{\sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s \boldsymbol{\theta}_i^s \left(\overline{a}_p^s b_{ip}^s - 1\right)}{\sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s \left(\overline{a}_p^s b_{ip}^s - 1\right)}, \\
\boldsymbol{\beta}_p^s &= \frac{\sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s \boldsymbol{\theta}_i^s \left(\overline{b}_p^s - b_{ip}^s\right)}{\sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s \left(\overline{a}_p^s b_{ip}^s - 1\right)}, \\
\Delta_p^s &= -\boldsymbol{\beta}_p^s \left(\overline{\boldsymbol{\theta}}_p^s - \boldsymbol{\mu}_p^s\right)^T - \left(\overline{\boldsymbol{\theta}}_p^s - \boldsymbol{\mu}_p^s\right) \boldsymbol{\beta}_p^{sT} + \overline{a}_p^s \boldsymbol{\beta}_p^s \boldsymbol{\beta}_p^{sT} \\
&\quad + \frac{1}{C^s} \sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s b_{ip}^s \left(\boldsymbol{\theta}_i^s - \boldsymbol{\mu}_p^s\right) \left(\boldsymbol{\theta}_i^s - \boldsymbol{\mu}_p^s\right)^T,
\end{aligned} \tag{13}$$

where $\overline{\boldsymbol{\theta}}_p^s = \frac{1}{C^s} \sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s \boldsymbol{\theta}_i^s$. We employ the strategy in the study [20] to update (r^s, ω^s) and omit them due to the limit of space. The updating rules in network $\mathcal{G}^t$ are obtained by replacing the superscript s with t.

Correctness: Note that, we estimate the hyperbolic distribution parameter $\boldsymbol{\psi} = (\boldsymbol{\mu}, \Delta, \boldsymbol{\beta}, r, \omega)$ without considering the positive definite constraint of scatter matrices Δ. Next, we prove that the positive definiteness is naturally guaranteed, *i.e.*, *Theorem 1 (Positive Definiteness)*, and thus show the correctness of the given updating rules.

Theorem 1 (Positive Definiteness). *All scatter matrices of* Δ_p^s *and* Δ_p^t *under updating rules in Eq. (13) are positive definite.*

Proof. The proofs of Δ_p^s and Δ_p^t are the same, thus we use Δ_p^s for the elaboration. To prove the positive definiteness of Δ_p^s, we first introduce an auxiliary matrix $\tilde{\Delta}_p^s$ defined as follows:

$$\tilde{\Delta}_p^s = \frac{1}{n_p} \sum_{i=1}^{N^s} z_{ip}^s b_{ip}^s \left(\boldsymbol{\theta}_i^s - \boldsymbol{\mu}_p^s - \frac{\boldsymbol{\beta}_p^s}{b_{ip}^s}\right) \left(\boldsymbol{\theta}_i^s - \boldsymbol{\mu}_p^s - \frac{\boldsymbol{\beta}_p^s}{c_{ip}^s}\right)^T . \tag{14}$$

Note that, $a_{ip}^s = \mathbb{E}\left[W_{ip}^s | \boldsymbol{\theta}_i^s, \boldsymbol{Z}_{ip}^s = 1\right]$ and $b_{ip}^s = \mathbb{E}[1/W_{ip}^s | \boldsymbol{\theta}_i^s, \boldsymbol{Z}_{ip}^s = 1]$. Based on Jensen's inequality, we have $1/\mathbb{E}\left[W_{ip}^s\right] \leq \mathbb{E}\left[1/W_{ip}^s\right]$ for all i, *i.e.*, $1/a_{ip}^s \leq b_{ip}^s$, and thus

$$\overline{a}_p^s = \frac{1}{\boldsymbol{Z}_p^s} \sum_{i=1}^{N^s} \boldsymbol{Z}_{ip}^s a_{ip}^s \geq \frac{1}{\boldsymbol{Z}_p^s} \sum_{i=1}^{N^s} \frac{\boldsymbol{Z}_{ip}^s}{b_{ip}^s}. \tag{15}$$

Finally, we have the following hold:

$$\Delta_p^s = \tilde{\Delta}_p^s + \left(\overline{a}_p^s - \frac{1}{\boldsymbol{Z}_p^s} \sum_{i=1}^{N^s} \frac{\boldsymbol{Z}_{ip}^s}{b_{ip}^s}\right) \boldsymbol{\beta}_p^s {\boldsymbol{\beta}_p^s}^T . \tag{16}$$

With the fact that $\boldsymbol{x}\boldsymbol{x}^T$ is positive definite for any $\boldsymbol{x} \in \mathbb{R}^d$, we ensure the positive definiteness of Δ_p^s as well as Δ_p^t. □

B. User Embedding Subproblem

With community embeddings fixed, we focus on updating user embeddings for user alignment. These user embeddings live in the common Poincaré ball $\mathcal{B}^d$, a smooth unit manifold with Riemannian metric $\mathfrak{g}^{\mathcal{B}}(\boldsymbol{\theta})$. In this manifold, the back-propagated gradient is Riemannian gradient and usual (Euclidean) gradient makes no sense as the operator of addition is not completely defined [21].

We optimize $\{\boldsymbol{\theta}_i^x\}$ and $\{\boldsymbol{\theta}_i^{x\prime}\}$ via exponential map with Riemannian gradient. Take $\mathcal{G}^s$ for instance. We first compute Riemannian gradient $\nabla_{\theta_i^s}^R \mathcal{J}$ of user $\boldsymbol{\theta}_i^s$ to identify optimizing direction in $T_{\boldsymbol{\theta}_i^s}\mathcal{B}^d$, and then leverage exponential map $\boldsymbol{\theta}_i^s \leftarrow \exp_{\theta_i^s}\left(-\rho \nabla_{\theta_i^s}^R \mathcal{J}\right)$ to move $\boldsymbol{\theta}_i^s$ along the mapped geodesic in the common Poincaré ball $\mathcal{B}^d$ with a step size ρ [21]. Fortunately, owing to the conformality of $\mathcal{B}^d$, Riemannian gradient $\nabla_\theta^R \mathcal{J}$ is obtained by rescaling the Euclidean gradient $\nabla_\theta^E \mathcal{J}$, *i.e.*, $\nabla_\theta^R \mathcal{J} = \left(\frac{1}{\lambda_\theta}\right)^2 \nabla_\theta^E \mathcal{J}$. The exponential map $\exp_\theta(\boldsymbol{a})$ at $\boldsymbol{\theta} \in \mathcal{B}^d$ is defined in the following fraction [14]:

$$\frac{\lambda_\theta \theta \left(\cosh\left(\lambda_\theta \|\boldsymbol{a}\|\right) + \left\langle \theta, \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} \right\rangle \sinh\left(\lambda_\theta \|\boldsymbol{a}\|\right)\right) + \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} \sinh\left(\lambda_\theta \|\boldsymbol{a}\|\right)}{1 + (\lambda_\theta - 1)\cosh\left(\lambda_\theta \|\boldsymbol{a}\|\right) + \lambda_\theta \left\langle \boldsymbol{\theta}, \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|} \right\rangle \sinh\left(\lambda_\theta \|\boldsymbol{a}\|\right)}. \tag{17}$$

The remaining challenge lies in the challenge of obtaining Euclidean gradients owing to summation within logarithm. To address this challenge, we optimize an upper bound objective function $\mathcal{J}_1 = \mathcal{O}_{user}^{\text{NS}} + \alpha_1 \mathcal{O}_{community}^{\text{UP}} + \alpha_2 \mathcal{O}_{align}^{\text{NS}}$ instead of $\mathcal{J}_0$. Specifically, we replace $\mathcal{O}_{community}$ with its upper bound $\mathcal{O}_{community}^{\text{UP}}$ defined as follows:

$$\mathcal{O}_{community}^{\text{UP}} = -\sum_{x \in \{s,t\}} \sum_{v_i^x \in \mathcal{V}^x} \sum_{p=1}^{C^x} \boldsymbol{Z}_{ip}^x \log Pr_{\mathcal{H}}(\boldsymbol{\theta}_p^x; \boldsymbol{\psi}_p^x). \tag{18}$$

It is easy to be verified via the log-concavity:

$$\log \sum\nolimits_{p=1}^{C^x} \boldsymbol{Z}_{ip} Pr_{\mathcal{H}}(\boldsymbol{\theta}_p^x; \boldsymbol{\psi}_p^x) \geq \sum\nolimits_{p=1}^{C^x} \boldsymbol{Z}_{ip} \log Pr_{\mathcal{H}}(\boldsymbol{\theta}_p^x; \boldsymbol{\psi}_p^x).$$

Thus, we have the partial derivative of $\mathcal{J}_1$ *w.r.t.* $\boldsymbol{\theta}_i^s$, *i.e.*, $\frac{\partial}{\partial \boldsymbol{\theta}_i^s} \mathcal{J}_1 = \frac{\partial \mathcal{O}_{user}^{\text{NS}}}{\partial \boldsymbol{\theta}_i^s} + \alpha_1 \frac{\partial \mathcal{O}_{community}^{\text{UP}}}{\partial \boldsymbol{\theta}_i^s} + \alpha_2 \mathbb{I}_{(v_i^s, v_k^t) \in \mathcal{A}} \frac{\partial \mathcal{O}_{align}^{\text{NS}}}{\partial \boldsymbol{\theta}_i^s}$,

$$\frac{\partial \mathcal{O}_{user}^{\text{NS}}}{\partial \boldsymbol{\theta}_i^s} = - \sum_{v_j^s \in \mathcal{N}_i^s} \left(\sigma[d(\boldsymbol{\theta}_j^{s\prime}, \boldsymbol{\theta}_i^s)] \frac{\partial d(\boldsymbol{\theta}_j^{s\prime}, \boldsymbol{\theta}_i^s)}{\partial \boldsymbol{\theta}_i^s} - \sum_{v_n \in \text{NS}_i^K} \mathbb{E}_{v_n} \left[\sigma[-d(\boldsymbol{\theta}_n^{s\prime}, \boldsymbol{\theta}_i^s)] \frac{\partial d(\boldsymbol{\theta}_n^{s\prime}, \boldsymbol{\theta}_i^s)}{\partial \boldsymbol{\theta}_i^s} \right] \right), \tag{19}$$

$$\frac{\partial \mathcal{O}_{community}^{\text{UP}}}{\partial \boldsymbol{\theta}_i^s} = - \sum_{p=1}^{C^s} \boldsymbol{\pi}_p^s \left((\Delta_p^s)^{-1} (\boldsymbol{\beta}_p^s + \frac{\zeta_p^s \tilde{\boldsymbol{\theta}}_i^s}{\tilde{\delta}_{\theta_i^s}}) + \frac{\partial \log K_{\zeta_p^s}(\sqrt{\nu_p^s \tilde{\delta}_{\theta_i^s}})}{\partial \boldsymbol{\theta}_i^s} \right), \tag{20}$$

$$\frac{\partial \mathcal{O}_{align}^{\text{NS}}}{\partial \boldsymbol{\theta}_i^s} = - \sum_{v_j^t \in \mathcal{N}_k^t} \left(\sigma[d(\boldsymbol{\theta}_j^{t\prime}, \boldsymbol{\theta}_i^s)] \frac{\partial d(\boldsymbol{\theta}_j^{t\prime}, \boldsymbol{\theta}_i^s)}{\partial \boldsymbol{\theta}_i^s} - \sum_{v_n \in \text{NS}_i^K} \mathbb{E}_{v_n} \left[\sigma[-d(\boldsymbol{\theta}_n^{s\prime}, \boldsymbol{\theta}_i^s)] \frac{\partial d(\boldsymbol{\theta}_n^{t\prime}, \boldsymbol{\theta}_i^s)}{\partial \boldsymbol{\theta}_i^s} \right] \right), \tag{21}$$

where $\nu_p^s = \omega_p^s + \boldsymbol{\beta}_p^{sT} (\Delta_p^s)^{-1} \boldsymbol{\beta}_p^s$, $\zeta_p^s = r_p^s - d/2$, $\tilde{\boldsymbol{\theta}}_i^s = \boldsymbol{\theta}_i^s - \boldsymbol{\mu}_p^s$, $\tilde{\delta}_{\theta_i^s} = \delta_{\theta_i^s} + \omega_p^s$ and $\sigma(x)$ is the standard sigmoid function. $\mathbb{I}_{(\cdot)}$ returns 1 iff the condition $(\cdot)$ is true; otherwise, $\mathbb{I}_{(\cdot)} = 0$. $\frac{\partial d(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}{\partial \boldsymbol{\theta}_1}$ is given in Eq. (3). Utilizing the fact $\frac{\partial}{\partial x} K_r(x) = -\frac{r}{x} K_r(x) - K_{r-1}(x)$, we give the partial derivative of $\log K_{\zeta_p^s}(\sqrt{\nu_p^s \tilde{\delta}_{\theta_i^s}})$ *w.r.t.* $\boldsymbol{\theta}_i^s$ as follows:

$$\frac{\partial \log K_{\zeta_p^s}(\sqrt{\nu_p^s \tilde{\delta}_{\theta_i^s}})}{\partial \boldsymbol{\theta}_i^s} = - \left(\frac{\zeta_p^s}{\tilde{\delta}_{\theta_i^s}} + \sqrt{\frac{\nu_p^s}{\tilde{\delta}_{\theta_i^s}}} \frac{K_{\zeta_p}(\sqrt{\nu_p^s \tilde{\delta}_{\theta_i^s}})}{K_{\zeta_p - 1}(\sqrt{\nu_p^s \tilde{\delta}_{\theta_i^s}})} \right) (\Delta_p^s)^{-1} \tilde{\boldsymbol{\theta}}_i^s.$$

Similarly, for $\boldsymbol{\theta}_j^{s\prime}$, we have $\frac{\partial \mathcal{J}_1}{\partial \boldsymbol{\theta}_j^{s\prime}} = \frac{\partial \mathcal{O}_{user}^{\text{NS}}}{\partial \boldsymbol{\theta}_j^{s\prime}} + \alpha_1 \frac{\partial \mathcal{O}_{align}^{\text{NS}}}{\partial \boldsymbol{\theta}_j^{s\prime}}$,

$$\frac{\partial \mathcal{O}_{user}^{\text{NS}}}{\partial \boldsymbol{\theta}_j^{s\prime}} = - \sum_{v_i^s \in \mathcal{V}^s} \left(\mathbb{I}_{v_j^s \in \mathcal{N}_i^s} \sigma[d(\boldsymbol{\theta}_j^{s\prime}, \boldsymbol{\theta}_i^s)] \frac{\partial d(\boldsymbol{\theta}_j^{s\prime}, \boldsymbol{\theta}_i^s)}{\partial \boldsymbol{\theta}_j^{s\prime}} - \sum_{v_n \in \text{NS}_i^K} \mathbb{E}_{v_n} \left[\mathbb{I}_{v_j^s = v_n^s} \sigma[-d(\boldsymbol{\theta}_n^{s\prime}, \boldsymbol{\theta}_i^s)] \frac{\partial d(\boldsymbol{\theta}_n^{s\prime}, \boldsymbol{\theta}_i^s)}{\partial \boldsymbol{\theta}_n^{s\prime}} \right] \right), \tag{22}$$

$$\frac{\partial \mathcal{O}_{align}^{\text{NS}}}{\partial \boldsymbol{\theta}_j^{s\prime}} = - \sum_{(v_i^s, v_k^t) \in \mathcal{A}} \left(\mathbb{I}_{v_j^s \in \mathcal{N}_i^s} \sigma[d(\boldsymbol{\theta}_j^{s\prime}, \boldsymbol{\theta}_k^t)] \frac{\partial d(\boldsymbol{\theta}_j^{s\prime}, \boldsymbol{\theta}_k^t)}{\partial \boldsymbol{\theta}_j^{s\prime}} - \sum_{v_n \in \text{NS}_i^K} \mathbb{E}_{v_n} \left[\mathbb{I}_{v_j^s = v_n^s} \sigma[-d(\boldsymbol{\theta}_n^{s\prime}, \boldsymbol{\theta}_k^t)] \frac{\partial d(\boldsymbol{\theta}_n^{s\prime}, \boldsymbol{\theta}_k^t)}{\partial \boldsymbol{\theta}_n^{s\prime}} \right] \right). \tag{23}$$

Note that, $\mathbb{I}_{(\cdot)}$ is defined above. The updating rules in network $\mathcal{G}^t$ are obtained by swapping the superscript s and t.

Remarks: In the common Poincaré ball, Eq. (20) encourages users to join in their corresponding communities, the alignment of which is inferred by user embeddings within the community. Hence, user embeddings and community embeddings are mutually refined for the joint social network alignment.

Computational Complexity: We analyze the computational complexity of Algorithm 1. First, we generate random walks (Line 1), costing $O(Nhl)$, $N := \max\{N^s, N^t\}$, where h is the number of walks per node and l is the walk length. Then, we solve two subproblems alternatively, which tends to be converged in a few iterations. Specifically, updating the parameters of community embedding subproblem (Line 5-7) costs $O(C^2 d^2)$, where $C := \max\{C^s, C^t\}$, and $O(Nhl(\epsilon K d + C d^2))$ for user embedding subproblem (Line 8-10), where ϵ is the size of the node neighborhood. To sum up, the overall complexity of PERFECT is linear to the number of users N.

V. EXPERIMENTS

In PERFECT, we jointly align communities and users in the hyperbolic space. Thus, we evaluate the performance of PERFECT with baseline methods on both community alignment and user alignment in the experiments, whose results are reported in Sections V-A and V-B, respectively. Datasets are introduced in Section III-A. We repeat each experiment 10 times and report the mean with 95% confidence interval.

A. Experiments on Community Alignment

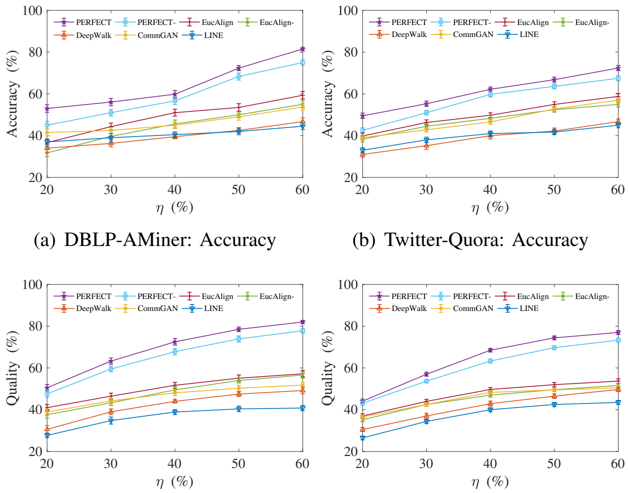
In this part, we provide the performance of PERFECT compared against comparison methods on community alignment.

1) Comparison Methods: To the best of our knowledge, the only method [22] considering community alignment is designed for attributed networks specifically, which will not work for normal networks without node attributes and thus is not comparable in our experiments. Specifically, the baseline methods compared here include both classic network embedding methods, *e.g.*, DeepWalk and LINE and the latest ones with considerations of community structures, *i.e.*, CommGAN:

- *CommGAN* [23]: It is a recent method using a minimax game to update node embeddings for community discovery. For each network, we perform CommGAN and calculate mean embedding in each community. Community alignment is obtained by calculating Euclidean distance.
- *DeepWalk* [18]: For each network, we perform DeepWalk to obtain node embeddings and then employ K-Means to discover communities. We align communities by Euclidean distance between mean embeddings.
- *LINE* [24]: We embed each network via LINE and use K-Means to discover communities, which will be aligned by Euclidean distance between mean embeddings.

To further evaluate PERFECT, we design several variants of it:

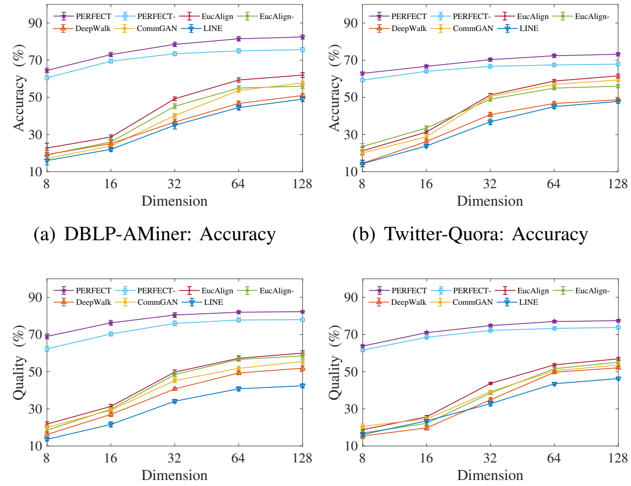
- PERFECT−: To demonstrate the superiority of unified optimization, we design a two-step method, *i.e.*, we learn the common subspace and community embeddings separately. Specifically, we first learn the common subspace via optimizing $\mathcal{O}_{user}^{\text{NS}} + \mathcal{O}_{align}^{\text{NS}}$ and then obtain community embeddings via optimizing $\mathcal{O}_{community}^{\text{UP}}$.
- *EucAlign*: To demonstrate the superiority of hyperbolic space, we consider the corresponding Euclidean version of PERFECT. We only replace the hyperbolic distance $d(\boldsymbol{x}, \boldsymbol{y})$ in PERFECT with Euclidean distance.
- *EucAlign−*: It is the Euclidean version of PERFECT− where the Euclidean distance is used.



(a) DBLP-AMiner: Accuracy (b) Twitter-Quora: Accuracy

(c) DBLP-AMiner: Quality (d) Twitter-Quora: Quality

Fig. 3. Community alignment under different overlaps



(a) DBLP-AMiner: Accuracy (b) Twitter-Quora: Accuracy

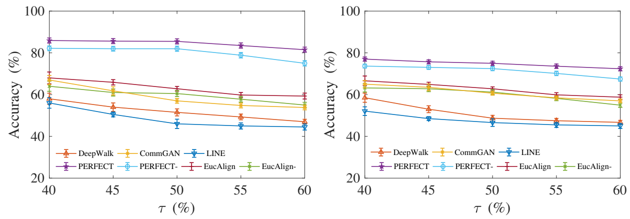
(c) DBLP-AMiner: Quality (d) Twitter-Quora: Quality

Fig. 4. Community alignment under different dimensions

2) Evaluation Metric:

- *Accuracy* $= \frac{1}{N_C}\sum_{i=1}^{N_C} success_\tau(\mathcal{C}_p^s, \mathcal{C}_q^t)$, where N_C is the number of groundtruth anchor communities. $success_\tau(\mathcal{C}_p^s, \mathcal{C}_q^t)$ returns 1 iff groundtruth anchor community $(\mathcal{C}_p^s, \mathcal{C}_q^t)$ is successfully identified *w.r.t.* the threshold τ; otherwise, it returns 0.
- *Quality* $= 2\mathbb{E}_i\{\sigma[-dist(\boldsymbol{\mu}_p^s, \boldsymbol{\mu}_q^t)]\}$, where the sigmoid $\sigma(\cdot)$ is for normalization. $\boldsymbol{\mu}_p^s$ and $\boldsymbol{\mu}_q^t$ are community embeddings of groundtruth anchor community $\mathcal{C}_p^s$ and $\mathcal{C}_q^t$, respectively. We use *Quality* to evaluate community alignment in the embedding space, and $dist(\cdot, \cdot)$ is the distance function of corresponding comparison methods.

3) Experimental Results and Discussions: First, we evaluate the performance under different overlap rates η. The overlap rate η is defined as $2|\mathcal{A}|/(N^s + N^t)$, where $|\mathcal{A}|$, N^s and N^t are the number of anchor users, source network users and target network users, respectively. An $\eta-$overlap dataset is generated by randomly deleting users according to the overlap rate from the dataset. Fixing embedding dimension $d = 64$, we report results under $\eta = \{20, 30, 40, 50, 60\}(\%)$ in terms of *Accuracy* and *Quality* on both datasets in Fig. 3. Second, we discuss the effect of embedding dimension d. Specifically, embedding dimension d takes different values in $[8, 16, 32, 64, 128]$ and we report results under $\eta = 60\%$ in terms of *Accuracy* and *Quality* in Fig. 4. Note that, the alignment threshold τ of *Accuracy* is set to 60% in the experiments above. Next, we study the parameter sensitivity of τ and report results under $d = 64$ and $\eta = 60\%$ in Fig. 5.



(a) DBLP-AMiner: Accuracy (b) Twitter-Quora: Accuracy

Fig. 5. Parameter sensitivity of the threshold τ

Finally, we summarize our findings and discuss the reasons:

- PERFECT *consistently outperforms its competitors.* The reason is that PERFECT enjoys the strengths of hyperbolic space and the unified optimization closing the loop of community alignment and user alignment.
- *The models of hyperbolic space consistently beat the Euclidean ones as shown in both Figs. 3 and 4.* Moreover, we obtain better performance on the dataset of higher hyperbolicity. (Refer to Section III-A and Table I.) The reason is that hyperbolic space better matches the inherent hyperbolicity of these datasets than the Euclid, and hyperbolic space benefits community alignment.
- *The models of unified optimization perform better in general as shown in both Figs. 3 and 4.* The reason is that, in a unified optimization, community alignment and user alignment benefit each other, while community alignment neglects the effect of user alignment and vice versa in the two-step methods.
- *The proposed hyperbolic model,* PERFECT, *with low-dimensional embeddings (e.g., 16) outperforms Euclidean ones with high-dimensional embeddings (e.g., 128) as shown in Fig. 4.* The reason is that, well suited for networks with latent hierarchy, PERFECT generates faithful embeddings with a few dimensions in hyperbolic space while it is not true for Euclidean ones.
- PERFECT *shows better robustness regarding threshold* τ *as shown in Fig. 5.* The reason is that, in PERFECT, there tends to be more anchor users among aligned communities, *i.e.*, better results regardless of τ.

B. Experiments on User Alignment

In this part, we will illustrate the learning performance of PERFECT compared against the comparison methods on user alignment.

TABLE II
THE PRECISION OF USER ALIGNMENT ON DBLP-AMINER AND TWITTER-QUORA DATASETS (%)

Dataset	Method	k=10	k=15	k=20	k=25	k=30
DBLP & AMiner	PALE	24.57 ± 2.23	31.36 ± 1.31	46.22 ± 0.98	52.37 ± 1.33	57.81 ± 1.51
	IONE	29.05 ± 1.13	35.21 ± 1.45	48.63 ± 1.58	53.45 ± 1.13	59.59 ± 1.41
	SNNA	32.46 ± 1.78	37.53 ± 1.85	43.26 ± 2.17	57.42 ± 2.04	63.09 ± 1.82
	MOANA	26.82 ± 1.37	35.46 ± 1.13	44.53 ± 1.33	55.81 ± 1.49	64.86 ± 1.35
	DeepLink	34.61 ± 1.69	54.25 ± 1.59	59.12 ± 1.29	62.68 ± 1.61	66.78 ± 1.12
	PERFECT	$\mathbf{44.53 \pm 1.23}$	$\mathbf{61.28 \pm 0.93}$	$\mathbf{67.85 \pm 1.11}$	$\mathbf{70.76 \pm 0.68}$	$\mathbf{72.33 \pm 0.73}$
Twitter & Quora	PALE	25.36 ± 1.74	29.43 ± 1.01	44.72 ± 0.75	51.03 ± 1.05	56.80 ± 1.14
	IONE	27.71 ± 0.87	33.57 ± 1.14	45.39 ± 1.23	50.92 ± 0.93	56.37 ± 1.06
	SNNA	31.05 ± 1.36	34.20 ± 1.45	39.55 ± 1.75	52.86 ± 1.66	59.49 ± 1.47
	MOANA	25.85 ± 1.04	34.59 ± 0.94	43.12 ± 1.23	54.07 ± 1.13	62.48 ± 1.05
	DeepLink	36.21 ± 1.33	53.67 ± 1.29	57.93 ± 1.01	61.52 ± 1.21	66.23 ± 0.86
	PERFECT	$\mathbf{42.12 \pm 0.93}$	$\mathbf{58.35 \pm 0.70}$	$\mathbf{66.67 \pm 0.91}$	$\mathbf{69.24 \pm 0.53}$	$\mathbf{71.02 \pm 0.55}$

TABLE III
THE MAP OF USER ALIGNMENT ON DBLP-AMINER AND TWITTER-QUORA DATASETS (%)

Dataset	Method	k=10	k=15	k=20	k=25	k=30
DBLP & AMiner	PALE	11.13 ± 0.65	11.75 ± 0.38	12.67 ± 0.29	12.96 ± 0.46	13.16 ± 0.50
	IONE	12.38 ± 0.37	12.93 ± 0.56	13.76 ± 0.62	13.99 ± 0.40	14.22 ± 0.48
	SNNA	13.24 ± 0.51	13.66 ± 0.69	14.00 ± 0.72	14.63 ± 0.78	14.84 ± 0.59
	MOANA	11.67 ± 0.55	12.43 ± 0.43	12.99 ± 0.50	13.52 ± 0.44	13.86 ± 0.48
	DeepLink	16.45 ± 0.60	18.13 ± 0.44	18.41 ± 0.40	18.58 ± 0.47	18.73 ± 0.43
	PERFECT	$\mathbf{18.86 \pm 0.47}$	$\mathbf{20.38 \pm 0.37}$	$\mathbf{20.79 \pm 0.36}$	$\mathbf{20.93 \pm 0.21}$	$\mathbf{20.99 \pm 0.22}$
Twitter & Quora	PALE	10.87 ± 0.61	11.24 ± 0.35	12.19 ± 0.27	12.49 ± 0.37	12.71 ± 0.39
	IONE	12.05 ± 0.33	12.56 ± 0.40	13.28 ± 0.51	13.53 ± 0.31	13.74 ± 0.42
	SNNA	12.93 ± 0.52	13.23 ± 0.59	13.57 ± 0.63	14.22 ± 0.62	14.48 ± 0.57
	MOANA	13.68 ± 0.39	14.47 ± 0.33	15.01 ± 0.37	15.52 ± 0.47	15.84 ± 0.37
	DeepLink	16.26 ± 0.55	17.83 ± 0.48	18.09 ± 0.35	18.26 ± 0.50	18.43 ± 0.33
	PERFECT	$\mathbf{17.55 \pm 0.38}$	$\mathbf{18.98 \pm 0.25}$	$\mathbf{19.51 \pm 0.31}$	$\mathbf{19.64 \pm 0.18}$	$\mathbf{19.70 \pm 0.20}$

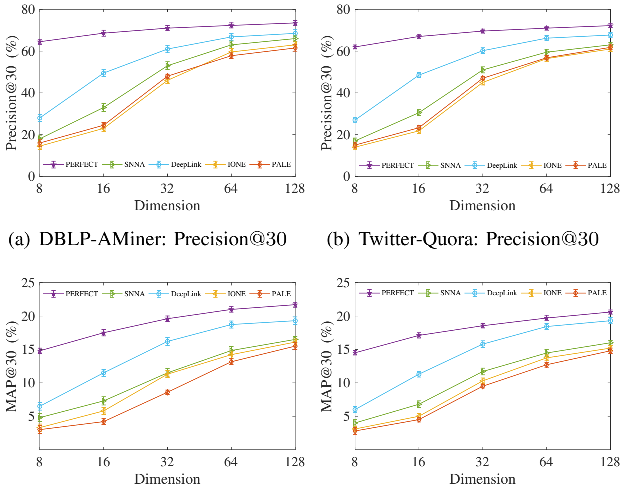
1) Comparison Methods: We chose several state-of-the-art methods on user alignment as follows:

- *IONE* [25]: It embeds social networks together with anchor links in the Euclidean subspace to align users.
- *PALE* [6]: It first embeds each network and then matches users via the Euclidean metric.
- *DeepLink* [7]: It leverages dual learning to refine the Euclidean subspace where network alignment is performed.
- *SNNA* [26]: It proposes a weakly-supervised adversarial learning method for alignment from the distribution level.
- *MOANA* [27]: It introduces a coarsen-align-interpolate method via matrix analysis to find node correspondence.

2) Evaluation Metric: We employ the widely used $Precision@K$ and $MAP@K$ as the evaluation metric.

3) Experimental Results and Discussions: First, we evaluate the performance of user alignment in terms of $Precision@k$ and $MAP@k$ on both datasets, whose experimental results are shown in Table II and Table III, respectively. Specifically, k takes different values in $[10, 15, ..., 30]$ and embedding dimension is set to 64. Our findings are two-fold: **(1)** PERFECT *consistently outperforms its comparison methods.* The reasons lie in that, besides the structural information encoded in typical embedding methods, *e.g.*, DeepLink, IONE, PALE, user embeddings in PERFECT incorporate the community structure explicitly and latent hierarchy among users in hyperbolic space implicitly, and thus own more discriminative information for user alignment. Additionally, this demonstrates that hyperbolic space benefits the user alignment.
(2) *The performance gain of* PERFECT *is correlated to the network hyperbolicity.* The proposed approach achieves higher performance on the dataset of higher hyperbolicity (DBLP-AMiner), and beats Euclidean methods on both datasets.

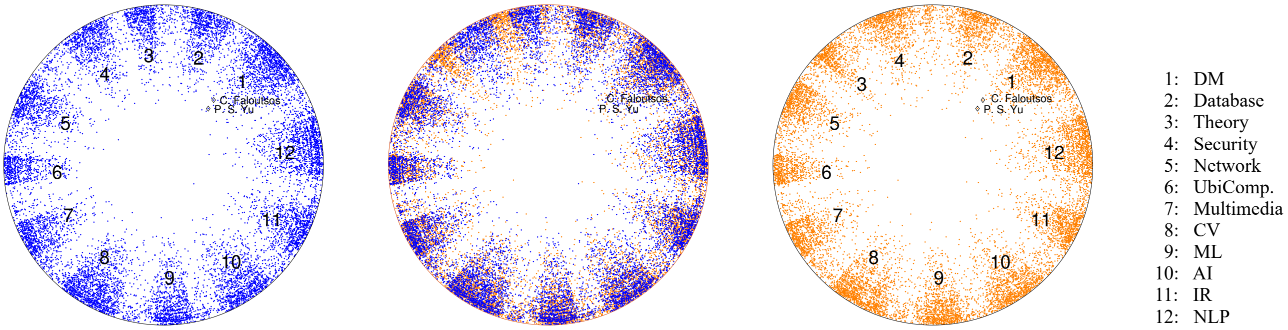
Second, we discuss the effect of embedding dimension d. We take different values in $[8, 16, 32, 64, 128]$, and report the performance of user alignment in terms of $Precision@30$ and $MAP@30$ on both datasets in Fig. 6. We omit the performance



(a) DBLP-AMiner: Precision@30 (b) Twitter-Quora: Precision@30

(c) DBLP-AMiner: MAP@30 (d) Twitter-Quora: MAP@30

Fig. 6. User alignment under different dimensions



(a) The embeddings of DBLP

(b) The common Poincaré disk

(c) The embeddings of AMiner

Fig. 7. The common Poincaré disk of DBLP-AMiner dataset

of MOANA as it is not an embedding method. We find that *as shown in Fig. 6,* PERFECT *consistently outperforms its competitors in all dimensions, and obtains dramatic performance gain with low-dimensional embeddings.*

C. Case Study

In this part, we will give a case study on DBLP-AMiner dataset to demonstrate the performance of PERFECT on both user alignment and community alignment.

We visualize the common Poincaré disk of PERFECT in Fig. 7 (b) by setting $d = 2$. We filter the embeddings of each network for clarity. The embeddings of DBLP and AMiner are plotted in Fig. 7 (a) and (c), respectively, where communities are labeled as illustrated in the legend. In this common Poincaré disk, we have three main findings:

(1) *The latent hierarchy among users is generally preserved in the two-dimensional embeddings.* We find that authors of high impact, *e.g.*, Philip S. Yu and Christos Faloutsos, are positioned closer to the origin, while those of relative low impact are pushed to the boundary of the disk.

(2) *User embeddings of the same user in different networks locate closely.* Zoom in the community of data mining. The user embedding of Philip S. Yu in DBLP, shown in Fig. 7 (a), is nearly the same as that in AMiner, Fig. 7 (c). Thus, user alignment is easy to be inferred via PERFECT.

(3) *Centripetal regions of corresponding communities are naturally aligned.* We find that user embeddings of the same community cluster into a centripetal region, which is also reported in the study [28], and it is obvious that most of centripetal regions of the same research area are aligned. That is, both users and communities are aligned in the common Poincaré disk, verifying the basic idea of our approach.

VI. RELATED WORKS

We briefly summarize the related works in following areas:

Network embedding is to map the nodes of a network into a vector space [29], [30]. Here, we roughly classify the literature by the embedding space. Most existing studies [18], [24], [31] explicitly or implicitly work with the Euclidean space. Some studies [32], [33] focus on exploiting in the static or temporal network structure, while others [34], [35] attempt to incorporate other attributes. However, is Euclidean space the appropriate embedding space? Recent advances uncover the hyperbolicity of some real-world networks, and researchers attempt to facilitate network embedding in hyperbolic space. Some studies [12], [13] suggest the superiority of hyperbolic geometry for network embedding. Recently, some neural networks on graphs [8], [36], [37] learn node embeddings underpinned by hyperbolic space. Distinguishing from these methods, PERFECT is tailored for network alignment.

User alignment, or anchor link prediction, is to align users across different social networks according to the underlying identity. To address this problem, some studies [38], [3] leverage attribute information, such as screen name pattern and user behaviors, to discover the identity consistency, while some studies [6] exploit network topology to link user identities. Moreover, there exist studies [39] considering both topology and attribute information. The study [40] gives a comprehensive survey. Recently, HGANE [41] incorporates attentive mechanism to induce the common subspace. GAlign [42] performs network alignment in an unsupervised way. Note that, similar to our prior works [43], [44], [45], [46], all of methods in the literature work with the Euclidean space, while PERFECT works with hyperbolic space. Additionally, beyond user alignment, PERFECT jointly considers community alignment in a unified approach.

Communities play a fundamental role in social network analysis. To the best of our knowledge, most existing studies [10] focus on community discovery in an isolated social network. Various types of methods have been proposed, such as modularity optimization [47] and spectral algorithms [48]. Generative methods are often explored as well, such as community affiliation models [49], model-based clustering and GAN [23]. Actually, social networks are correlated and partially aligned on their common users. Some studies [50] facilitate community discovery with the information of its counterpart network. Recently, the study [22] considers the community structure to facilitate user alignment across social networks. Different from these studies, we for the first time close the loop of community alignment and user alignment so that they mutually enhanced each other.

VII. CONCLUSION

In this paper, we present a novel hyperbolic optimization framework, namely PERFECT, to jointly align users and communities in hyperbolic space. To address the optimization of PERFECT, we propose a novel alternating Riemannian optimization algorithm with solid theoretical analyses so that user alignment and community alignment benefit from each other. Extensive experiments show the superiority of PERFECT in both user alignment and community alignment.

ACKNOWLEDGMENT

This work was supported in part by: National Natural Science Foundation under Grants U1936103 and 61921003, National Key Research and Development Program of China under Grant 2018YFB1003804, Fundamental Research Funds for the Central Universities 2019XD11, Natural Science Foundation under Grants III-1526499, III-1763325, III-1909323, SaTC-1930941, IIS-1763365, and by Florida State University.

REFERENCES

[1] J. Zhang and P. S. Yu, "Pct: Partial co-alignment of social networks," in *WWW*, 2016, pp. 749–759.

[2] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "Cosnet: Connecting heterogeneous social networks with local and global consistency," in *KDD*, 2015, pp. 1485–1494.

[3] X. Mu, F. Zhu, E.-P. Lim, J. Xiao, J. Wang, and Z.-H. Zhou, "User identity linkage by latent user space modelling," in *KDD*, 2016, pp. 1775–1784.

[4] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *CIKM*, 2013, pp. 179–188.

[5] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach," in *KDD*, 2013, pp. 41–49.

[6] T. Man, H. Shen, S. Liu, X. Jin, and X. Cheng, "Predict anchor links across social networks via an embedding approach," in *IJCAI*, 2016, pp. 1823–1829.

[7] F. Zhou, L. Liu, K. Zhang, G. Trajcevski, J. Wu, and T. Zhong, "Deeplink: A deep learning approach for user identity linkage," in *INFOCOM*, 2018, pp. 1313–1321.

[8] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *NIPS*, 2019, pp. 4869–4880.

[9] E. Ravasz and A.-L. Barabási, "Hierarchical organization in complex networks," *Physical review E*, vol. 67, no. 2, p. 026112, 2003.

[10] E. Abbe, "Community detection and stochastic block models: recent developments," *JMLR*, vol. 18, no. 1, pp. 6446–6531, 2017.

[11] W. Chen, W. Fang, G. Hu, and M. W. Mahoney, "On the hyperbolicity of small-world and treelike random graphs," *Internet Mathematics*, vol. 9, no. 4, pp. 434–491, 2013.

[12] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *NIPS*, 2017, pp. 6338–6347.

[13] X. Wang, Y. Zhang, and C. Shi, "Hyperbolic heterogeneous information network embedding." AAAI, 2019.

[14] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic entailment cones for learning hierarchical embeddings," in *ICML*, 2018.

[15] A. Tifrea, G. Bécigneul, and O.-E. Ganea, "Poincaré glove: Hyperbolic word embeddings," in *ICLR*, 2019.

[16] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná, "Hyperbolic geometry of complex networks," *Physical Review E*, vol. 82, no. 3, p. 036106, 2010.

[17] 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 *WSDM*, 2020, pp. 609–617.

[18] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, pp. 701–710.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

[20] R. P. Browne and P. D. McNicholas, "A mixture of generalized hyperbolic distributions," *Canadian Journal of Statistics*, vol. 43, no. 2, pp. 176–198, 2015.

[21] S. Bonnabel, "Stochastic gradient descent on riemannian manifolds," *IEEE Trans. on Automatic Control*, vol. 58, no. 9, pp. 2217–2229, 2013.

[22] Z. Chen, X. Yu, B. Song, J. Gao, X. Hu, and W.-S. Yang, "Community-based network alignment for large attributed network," in *CIKM*, 2017, pp. 587–596.

[23] Y. Jia, Q. Zhang, W. Zhang, and X. Wang, "Communitygan: Community detection with generative adversarial nets," in *WWW*, 2019, pp. 784–794.

[24] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015, pp. 1067–1077.

[25] L. Liu, W. K. Cheung, X. Li, and L. Liao, "Aligning users across social networks using network embedding," in *IJCAI*, 2016, pp. 1774–1780.

[26] C. Li, S. Wang, Y. Wang, P. S. Yu, Y. Liang, Y. Liu, and Z. Li, "Adversarial learning for weakly-supervised social network alignment," in *AAAI*, vol. 33, 2019, pp. 996–1003.

[27] S. Zhang, H. Tong, R. Maciejewski, and T. Eliassi-Rad, "Multilevel network alignment," in *WWW*, 2019.

[28] A. Muscoloni, J. M. Thomas, S. Ciucci, G. Bianconi, and C. V. Cannistraci, "Machine learning meets complex networks via coalescent embedding in the hyperbolic space," *Nature Communications*, vol. 8, no. 1, p. 1615, 2017.

[29] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. on Knowledge and Data Engineering*, 2018.

[30] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Network representation learning: A survey," *IEEE Trans. on Big Data*, pp. 1–25, 2018.

[31] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, 2016, pp. 855–864.

[32] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep recursive network embedding with regular equivalence," in *KDD*, 2018, pp. 2357–2366.

[33] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *IJCAI*, 2018, pp. 2609–2615.

[34] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *WSDM*, 2017, pp. 731–739.

[35] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *WWW*, 2018, pp. 1063–1072.

[36] Q. Liu, M. Nickel, and D. Kiela, "Hyperbolic graph neural networks," in *NIPS*, 2019, pp. 8228–8239.

[37] 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," *ICLR*, 2019.

[38] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "Hydra: Large-scale social identity linkage via heterogeneous behavior modeling," in *SIGMOD*, 2014, pp. 51–62.

[39] Z. Zhong, Y. Cao, M. Guo, and Z. Nie, "Colink: An unsupervised framework for user identity linkage," in *AAAI*, 2018, pp. 5714–5721.

[40] K. Shu, S. Wang, J. Tang, R. Zafarani, and H. Liu, "User identity linkage across online social networks: A review," *SIGKDD Explorations*, vol. 18, no. 2, pp. 5–17, 2017.

[41] Y. Jiao, Y. Xiong, J. Zhang, and Y. Zhu, "Collective link prediction oriented network embedding with hierarchical graph attention," in *CIKM*, 2019, pp. 419–428.

[42] T. T. Huynh, V. V. Tong, T. T. Nguyen, H. Yin, M. Weidlich, and Q. V. H. Nguyen, "Adaptive network alignment with unsupervised and multi-order convolutional networks," *ICDE*, pp. 85–96, 2019.

[43] L. Sun, Z. Zhang, P. Ji, J. Wen, S. Su, and P. S. Yu, "Dna: Dynamic social network alignment," in *IEEE Intl. Conf. on Big Data*, 2019, pp. 1124–1231.

[44] S. Su, L. Sun, Z. Zhang, G. Li, and J. Qu, "Master: across multiple social networks, integrate attribute and structure embedding for reconciliation," in *IJCAI*, 2018, pp. 3863–3869.

[45] Z. Zhang, L. Sun, S. Su, J. Qu, and G. Li, "Reconciling multiple social networks effectively and efficiently: An embedding approach," in *IEEE Trans. on Knowledge and Data Engineering*.

[46] F. Ren, Z. Zhang, J. Zhang, S. Su, L. Sun, G. Zhu, and C. Guo, "Banana: when behavior analysis meets social network alignment," in *IJCAI*, 2020, pp. 1348–1444.

[47] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[48] P. Chin, A. Rao, and V. Vu, "Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery," in *Conference on Learning Theory*, 2015, pp. 391–423.

[49] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *WSDM*, 2013, pp. 587–596.

[50] J. Zhang and P. S. Yu, "Community detection for emerging networks," in *SDM*, 2015, pp. 127–135.