ALIGNMENT AND OUTER SHELL ISOTROPY FOR HYPERBOLIC GRAPH CONTRASTIVE LEARNING –APPENDICES–

Anonymous authors
Paper under double-blind review

A NOTATIONS

Notations. In this paper, a graph with node features is denoted as $G = (\mathbf{X}, \mathbf{A})$ and $\mathbf{X} \in \mathbb{R}^{N\times d_x}$ is the feature matrix (*i.e.*, the i-th row of $\mathbf{X}$ is the feature vector $\boldsymbol{x}_i$ of node v_i) and $\mathbf{A} \in \{0,1\}^{n\times n}$ denotes the adjacency matrix of G, *i.e.*, the (i,j)-th entry in $\mathbf{A}$ is 1 if there is an edge between nodes i and j. The degree of node i, denoted as d_i, is the number of edges incident with i. The degree matrix $\mathbf{D}$ is a diagonal matrix and its i-th diagonal entry is d_i. For a d-dimensional vector $\boldsymbol{x} \in \mathbb{R}^d$, $||\boldsymbol{x}||_2$ is the Euclidean norm of $\boldsymbol{x}$. We use x_i to denote the i th entry of $\boldsymbol{x}$, and x_{ij} for the (i,j)-th entry of $\mathbf{X}$. $\operatorname{diag}(\boldsymbol{x}) \in \mathbb{R}^{d\times d}$ is a diagonal matrix such that the i-th diagonal entry is x_i. We use $\boldsymbol{x}_i$ denote the row vector of $\mathbf{X}$. The trace of a square matrix $\mathbf{X}$ is denoted by $\operatorname{tr}(\mathbf{X})$, which is the sum along the diagonal of $\mathbf{X}$.

B PROJECTION INTO THE POINCARÉ BALL

Assume the output space of the graph neural network $f_\Theta(\cdot)$ is in the Poincaré ball $\mathbb{D}^d_c$, we project the all the node embedding to the $\mathbb{D}^d_c$ as

$$\boldsymbol{z} := \begin{cases} \boldsymbol{z} & \text{if } ||\boldsymbol{z}|| \le \frac{1}{c} \\ (1-\epsilon)\frac{\boldsymbol{z}}{c||\boldsymbol{z}||} & \text{else} \end{cases} \tag{13}$$

C PROOF OF THEOREM 2

It directly follow from the transformation of random variables. Specifically, $p_Z(\mathbf{z}) = p_{\mathcal{N}}(f^{-1}(\mathbf{z}))\cdot \det\left(\mathbf{J}(f^{-1}(\mathbf{z}))\right)$. Notice that for $f(\boldsymbol{v}) = \exp^c_{\mathbf{0}}(\boldsymbol{v})$ the inverse is logarithmic map $f^{-1}(\boldsymbol{z}) = \log^c_{\mathbf{0}}(\boldsymbol{z}) = \frac{1}{\sqrt{c}||\mathbf{z}||_2}\tanh^{-1}(\sqrt{c}||\mathbf{z}||_2)\frac{\boldsymbol{z}}{\sqrt{c}||\boldsymbol{z}||_2}$. The main difficulty lies with computing the Jacobian $\mathbf{J}(f^{-1}(\mathbf{z}))$ and its determinant $\det\left(\mathbf{J}(f^{-1}(\mathbf{z}))\right)$, which (after crunching some maths) turns out to enjoy a simple analytical form $0.5\,\lambda^c_{\mathbf{z}}\,g^{d-1}(\mathbf{z})$.

D PROOF OF LEMMA 3

$$D(\boldsymbol{\Sigma}, \boldsymbol{\mu}) = \operatorname{tr}(\boldsymbol{\Sigma}) - \log\det(\boldsymbol{\Sigma}) - d = \sum_{i=1}^{d}\left(\lambda_i - \log\lambda_i - 1\right). \tag{14}$$

We usually centralize the embedding $\{log^c_{\mathbf{0}}(\boldsymbol{z}_i)\}$, therefore we ignore the $\boldsymbol{u}$ for brevity in Eq. (14). We know that $x - \log x \ge 1$ with equality at $x = 1$. and $x - \log x \ge \log x + 1 - \log 4$ with equality at $x = 2$. Given $\lambda_1 \ge \lambda_2 \cdots \ge \lambda_d > 0$, we have:

$$\begin{aligned} &D(\boldsymbol{\Sigma}, \boldsymbol{\mu}) \ge (\log\lambda_1 + 1 - \log 4) - (\log\lambda_d + 1) \\ &D(\boldsymbol{\Sigma}, \boldsymbol{\mu}) \ge (\log\lambda_1 - \log\lambda_d) + const \\ &D(\boldsymbol{\Sigma}, \boldsymbol{\mu}) \ge \left(\log\frac{\lambda_1}{\lambda_d}\right) + const = \log\frac{\lambda_1}{\lambda_2} + \log\frac{\lambda_2}{\lambda_3}\cdots\log\frac{\lambda_{d-1}}{\lambda_d} + \log\frac{\lambda_d}{\lambda_d} + const. \end{aligned} \tag{15}$$

Let $q_i = \frac{\lambda_i}{\sum_i^d \lambda_i}$ and $0 < q_i \leq 1$, then:

$$
\begin{aligned}
D(\mathbf{\Sigma}, \boldsymbol{\mu}) &= \log \frac{q_1}{q_2} + \log \frac{q_2}{q_3} + \log \frac{q_{d-1}}{q_d} + \log \frac{q_d}{q_d} + const \\
&\geq \log q_1 + \log q_2 \cdots \log q_{d-1} + \log q_d + const \\
&\geq \sum_{q=1}^{d} q_i \log q_i + const \\
-D(\mathbf{\Sigma}, \boldsymbol{\mu}) &\leq -\sum_{q=1}^{d} q_i \log q_i + const \\
\exp(-D(\mathbf{\Sigma}, \boldsymbol{\mu})) &\leq \exp(-\sum_{q=1}^{n-1} q_i \log q_i) + const \\
D(\mathbf{\Sigma}, \boldsymbol{\mu}) &\geq -\log \operatorname{Erank}(\mathbf{\Sigma}) + const.
\end{aligned}
\tag{16}
$$

Thus, our loss yields an upper bound on the Effective Rank.

E Setting of the Collaborative Filtering

Datasets. We use three publicly available datasets Amazon-Book, Amazon-CD, and Yelp2020, which are also employed in the HRCF. The statistics are summarized in Table 5 in the appendix.

Baselines. Compared methods. To verify the effectiveness of our proposed method, the compared methods include both well-known or leading hyperbolic models and Euclidean baselines. For hyperbolic models, the HGCF (Sun et al., 2021), HVAE and HAE (Liang et al., 2018) and are compared. HAE (HVAE) combines a (variational) autoencoder with hyperbolic geometry. Besides, we include strong Euclidean baselines, *i.e.*, LGCN (He et al., 2020) and NGCF (Wang et al., 2019).

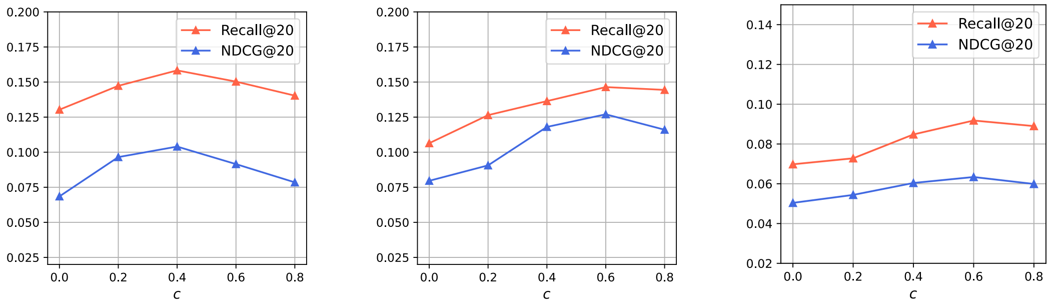
Setting. To show that hyperbolic uniformity is crucial for learning the hierarchical representation, we combine the proposed uniformity metric with the existing SOTA method (*i.e.*, HRCF (Yang et al., 2022)) by adding $\mathcal{L}_U^{\mathbb{D}_c^d}$ as an auxiliary loss. We test the model using the relevancy-based metric Recall@20 and the ranking-aware metric NDCG@20. In order to maintain a fair comparison and reduce the workload of our experiments, we closely adhere to the settings of HRCF (Yang et al., 2022). Specifically, we set the embedding size to 50 and fix the total training epochs at 500. The range of λ values in the loss function is $\{10, 15, 20, 25, 30\}$, while the aggregation order is searched in range from 2 to 10. When it comes to the margin, we explore values within the range of $\{0.1, 0.15, 0.2\}$. To train the network parameters, we employ Riemannian SGD (Bonnabel, 2013) with weight decay, using values from the range 1e-4, 5e-4, 1e-3, along with learning rates of $\{0.001, 0.0015, 0.002\}$. RSGD is a technique that emulates stochastic gradient descent optimization while accounting for the geometry of the hyperbolic manifold (Bonnabel, 2013). For the baseline settings of HAE, HAVE and HGCF, we refer the reader to (Sun et al., 2021).

Table 5: Statistics of the experimental datasets.

Dataset	#User	#Item	#Interactions	Density
Amazon-CD	22,947	18,395	422,301	0.00100
Amazon-Book	52,406	41,264	1,861,118	0.00086
Yelp2020	71,135	45,063	1,940,014	0.00047

F Impact of Curvature c.

Since the curvature parameter c controls the depth of hierarchy (height of the tree embedding), we analyze its effect on results. The notion of height-level uniformity is related to the value of c: the larger c is, the more concentration of the distribution towards the tree root. Figure 8 shows results w.r.t. varying c. The result shows HyperGCL achieves the best result for different c meaning the the height-level uniformity is data dependent and related to sparsity of the datasets (sparsity is indicated in caption brackets of Figure 8), *e.g.*, graphs with relatively larger density require smaller c.



(a) Amazon-CD (0.01)

(b) Amazon-Book (0.008)

(c) Yelp2020 (0.004)

Figure 8: Performance w.r.t. the value of curvature c. In caption brackets, we indicate the dataset sparsity.

G Broader impact and limitations

Our method enjoys impact and limitations similar to those in graph contrastive learning. Typical GCL models cannot guarantee they can utilize the feature space efficiently due to the mode collapse phenomenon. As we utilize the feature space more efficiently due to the Hyperbolic geometry and the penalty preventing collapse, our model works better, delivering better prediction on graphs for the similar computational cost. Our idea can be universally applied to other scenarios where the mode collapse is an issue. Of course, in this work we do not study fairness or biases but we believe that poorer utilization of the feature space in other methods can exacerbate such issues.