A GEOMETRY-AWARE ALGORITHM TO LEARN HIERARCHICAL EMBEDDINGS IN HYPERBOLIC SPACE

Zhangyu Wang *
Alibaba Inc.
Hangzhou, China 310000
zhangyu.wzy@alibaba-inc.com

Lantian Xu
Carnegie Mellon University
Pittsburgh, PA, USA 15213
lxu2@andrew.cmu.edu

Zhifeng Kong
University of California San Diego
La Jolla, CA, USA 92092
z4kong@eng.ucsd.edu

Weilong Wang
Purdue University
West Lafayette, IN, USA, 47906
wang4167@purdue.edu

Xuyu Peng, Enyang Zheng
Alibaba Inc.
Hangzhou, China 310000
{xijiu.pxy, enyang.zhengey}@alibaba-inc.com

ABSTRACT

Hyperbolic embeddings are a class of representation learning methods that offer competitive performances when data can be abstracted as a tree-like graph. However, in practice, learning hyperbolic embeddings of hierarchical data is difficult due to the different geometry between hyperbolic space and the Euclidean space. To address such difficulties, we first categorize three kinds of illness that harm the performance of the embeddings. Then, we develop a geometry-aware algorithm using a dilation operation and a transitive closure regularization to tackle these illnesses. We empirically validate these techniques and present a theoretical analysis of the mechanism behind the dilation operation. Experiments on synthetic and real-world datasets reveal superior performances of our algorithm.

1 INTRODUCTION

Learning data representation is important in machine learning as it provides a metric space that reveals or preserves inherent data structure (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017; Hoff et al., 2002; Grover & Leskovec, 2016; Perozzi et al., 2014; Nickel et al., 2011; Bordes et al., 2013; Riedel et al., 2013). Hyperbolic embeddings, a class of hierarchy representation methods, have shown competitive performances when data can be abstracted as a graph (Chamberlain et al., 2017; Davidson et al., 2018; Ganea et al., 2018a; Gu et al., 2018; Tifrea et al., 2018).

*The author's corresponding email is zhangyuwang@ucsb.edu. The work is done during an internship at Alibaba Inc.

In this work, we focus on the following embedding task. Let $\mathcal{D}$ be a dataset incorporated with a set of hierarchical relations represented as edges in a tree-like graph $\mathcal{G}$. The goal is to learn an embedding Θ in the *hyperbolic* space by drawing positive and negative samples of edges from the graph such that Θ preserves the edge relationships, which are reflected by the order of similarity between data pairs. The formal problem statement is presented in Section 2.

Theoretically, hyperbolic space, such as the Poincaré Ball model, benefit from high representational power due to their negative curvatures (Nickel & Kiela, 2017; Sala et al., 2018). This observation has motivated research on solving real-world problems in hyperbolic space. For datasets with an *observed* structure, hyperbolic space can embed the data and preserve the structure with arbitrarily low distortion (Nickel & Kiela, 2017; Chamberlain et al., 2017; Nickel & Kiela, 2018; Ganea et al., 2018b; Chami et al., 2019c). For datasets with a *latent* structure, especially those obeying the power-law, hyperbolic space can provide a natural metric such that finer concepts are embedded into areas allowing more subtlety (Tifrea et al., 2018; Leimeister & Wilson, 2018; Le et al., 2019).

Despite the theoretical advantages of hyperbolic embeddings, learning such representation in practice is difficult. Specifically, the following fundamental difficulties have not been well-studied in the literature. (1) Many properties of the Euclidean space do not transfer to hyperbolic space. For example, the latter generally do not have the scale or shift-invariance in the sense of preserving similarity orders. (2) Many nice properties exclusive to hyperbolic space may improve learning. However, it is unclear how to design algorithms to effectively incorporate these properties. (3) Optimization in hyperbolic space is (i) expensive due to a more sophisticated distance measure and (ii) unstable because gradient descent is performed on hyperbolic manifolds.

In this paper, we analyze these difficulties and provide a set of solutions to them. First, we define bad cases as improper relationship between nodes and edges. We then categorize them into *capacity illness*, *inter-subtree illness*, and *intra-subtree illness*. Formal definitions and intuitive visualizations are presented in Section 3. We present a theoretical analysis of local capacity, capacity illness, and their relationship in Section 4. We then develop an algorithm that reduces these illness in Section 5. The algorithm involves a *dilation* operation during the learning process, adding transitive closure edges of data to positive samples, and a re-weighting strategy. We conduct experiments on synthetic and real world datasets in Appendix 6. The results show that our algorithm achieves superior performances under various evaluation metrics.

2 PRELIMINARIES

A hyperbolic space H^d is a d-dimensional Riemannian manifold with a constant negative sectional curvature. In this paper, we focus on the Poincaré ball model. Let $\mathcal{B} = \mathcal{B}^d$ denote the d-dimensional Poincare ball. The distance between any two points $B_1, B_2 \in \mathcal{B}^d$ is defined as

$$d(B_1, B_2) = \operatorname{arcosh}\left(1 + 2\frac{\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}\right), \tag{1}$$

where u and v are the Euclidean vectors of B_1 and B_2. In the rest of the paper, we denote the Poincaré distance by $d(\cdot, \cdot)$ and the Euclidean distance by $\|\cdot\|$. Given a set of points $\mathcal{V} = \{x_i\}_{i=1}^n$ and the relation set $\mathcal{E} \subset [n]^2$, the goal is to learn an embedding $f : \mathcal{V} \to \mathcal{B}$ that preserves the inherent structure. To achieve this goal, we define and minimize the following loss function $\mathcal{L}$. For $(i, j) \in \mathcal{E}$, define $\mathcal{N}(x_i, x_j)$ as the set of negative samples of (i, j). Let $\Theta = \{\theta_i\}_{i=1}^n$, where each $\theta_i \in \mathcal{B}^d$ is the embedding of x_i. Define $d(x_i, x_j) = d(\theta_i, \theta_j)$. Then, the loss function is defined as

$$\mathcal{L}(\Theta) = -\sum_{(i,j)\in\mathcal{E}} \log \frac{e^{-d(x_i,x_j)}}{\sum_{x' \in \mathcal{N}(x_i,x_j)\cup\{x_j\}} e^{-d(x_i,x')}} = -\sum_{(i,j)\in\mathcal{E}} \mathcal{L}_{i,j}(\Theta). \tag{2}$$

This objective can be optimized via Riemannian gradient descent (Nickel & Kiela, 2017).

3 ILLNESS

In this section, we formally define illness that harms the performance of hyperbolic embeddings and is hard to optimize. Let $\overrightarrow{AB}$ be a ground-truth edge in G, and $\overrightarrow{AB'}$ be the inferred edge from the

Algorithm 1 Geometry-Aware Algorithm

for $i = 1$ **to** N_{epoch} **do**
  Compute local capacity according to equation 5
  if local capacity is not sufficient **then** perform the dilation operation in equation 6
  end if
  if $i \leq N_{\text{tc}}$ **then** loss $\leftarrow \mathcal{L}_{\text{tc}}(\Theta)$ according to equation 7
  else loss $\leftarrow \mathcal{L}(\Theta)$ according to equation 2
  end if
  Apply Riemannian gradient descent over loss
end for

hyperbolic embeddings, where $B' \neq B$. We call this situation *illness with respect to* A. Let C be the nearest common ancestor of B and B'. We categorize three kinds of illness according to the pairwise relationships among B, B', and C. Formally, we define capacity illness, intra-subtree illness, and inter-subtree illness in **Definition** 1.

Definition 1 (Categories of Illness)**.** *We define the illness to be capacity illness if B is the parent of B'. We define the illness to be an intra-subtree illness if B is the ancestor but not the parent of B'. We define the illness to be an inter-subtree illness if $C \neq B$.*

It is straightforward to see that the union of capacity illness and intra-subtree illness are exactly situations where $C = B$. Therefore, the above three kinds of illness are a partition of all illness. We visualize these three kinds of illness in Figure 3 in the appendix.

4 Local Capacity

We define local capacity below and theoretically relate it to capacity illness.

Definition 2 (Local Capacity)**.** *Given a geodesic space $(\mathcal{X}, d)$ and a geodesic ball $\mathcal{S}_r$ centered at $A \in \mathcal{X}$ with radius r. The local capacity of (A, r) is defined as*

$$\max\left\{|\mathcal{C}| : \mathcal{C} \in \mathcal{S}_r; \forall C_1, C_2 \in \mathcal{C}, C_1 \neq C_2, d(C_1, C_2) > d(C_1, A) \vee d(C_2, A)\right\}. \tag{3}$$

Given a geodesic space $(\mathcal{X}, d)$ and a geodesic ball $\mathcal{S}_r$ centered at $A \in \mathcal{X}$ with radius r. The local capacity of (A, r) is defined as

$$\max\left\{|\mathcal{C}| : \mathcal{C} \in \mathcal{S}_r; \forall C_1, C_2 \in \mathcal{C}, C_1 \neq C_2, d(C_1, C_2) > d(C_1, A) \vee d(C_2, A)\right\}. \tag{4}$$

For not very small r and large d we have the following bounds:

$$2^d e^{\frac{dr}{2}} \gtrapprox \mathcal{A}(d, \theta_r) \gtrapprox \sqrt{2\pi} \log \frac{2}{\sqrt{3}} \cdot d^{\frac{3}{2}} \cdot 2^{1-d} e^{\frac{d-1}{2} r}, \tag{5}$$

where $\theta_r = \arcsin\left(1/(2\cosh(r/2))\right)$. Full derivations are in Appendix B.

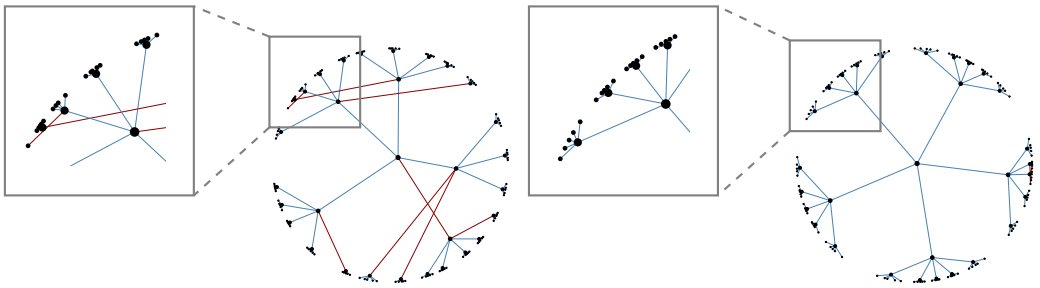
5 The Algorithm

In this section, we build a geometry-aware algorithm (Algorithm 1) targeting the three categories of illness by proposing the dilation operation and the transitive closure regularization.

Dilation. We define a mapping $g : \mathcal{B} \to \mathcal{B}$ as a k-dilation if for any $A \in \mathcal{B}$:

$$d(O, g(A)) = k \cdot d(O, A). \tag{6}$$

Notably, g can be computed explicitly. For instance, a 2-dilation can be formulated as $g(A) = \frac{2}{1+\|A\|^2} A$. The dilation operation rescales the embedded structure so that each point is pushed to a location with sufficient local capacity. Given $A \in \mathcal{B}$ with degree k, this operation helps increase the distance between A and its k-nearest neighbor(r_A), thus increase the local capacity of (A, r_A).



(a) The baseline Poincaré embedding algorithm (b) Our algorithm

Figure 1: Visualizations of two-dimensional embeddings of a synthetic balance tree (156 nodes, 155 edges) learned by the baseline Poincaré embedding algorithm in Nickel & Kiela (2017) and our geometry-aware algorithm, respectively. Both algorithms are trained for 3000 epochs. The lines refer to ground-truth edges and the points refer to the learned hyperbolic embeddings. The red lines indicate bad cases where the embeddings fail to reconstruct these ground-truth edges.

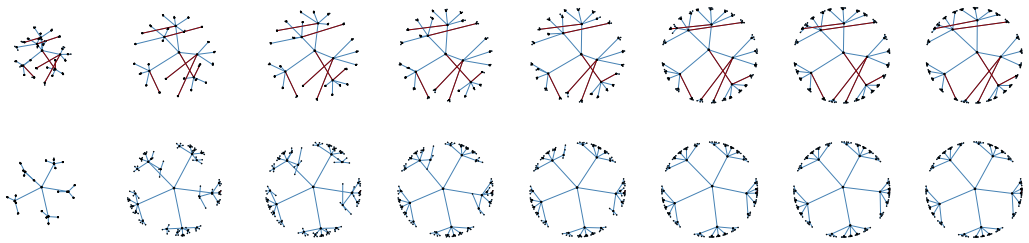


Figure 2: Visualization of the learning process of two-dimensional Poincaré embeddings with (1) the baseline algorithm in the upper row, and (2) the geometry-aware algorithm in the lower row. The dataset, baseline model, and plotting settings are identical as in Figure 1. We plot intra-subtree and inter-subtree illness consistently existing throughout the entire 3000 epochs.

Transitive closure regularization. It contains the following two operations.

Adding transitive closure edges. The transitive closure edges $\mathcal{E}_{\mathrm{tc}}$ are edges between nodes and their non-parent ancestors. These edges are also considered as positive samples in addition to $\mathcal{E}$ in the objective in equation 2. The purpose of adding these auxiliary edges is to push the subtrees apart so they are less likely to overlap in the Poincaré ball.

Re-weighting. We modify the weights of transitive closure edges to prevent overfitting in early (the first N_{tc}) epochs. Let η_{tc} be a real number between 0 and 1. Then, the objective becomes

$$\mathcal{L}_{\mathrm{tc}}(\Theta) = \mathcal{L}(\Theta) + \eta_{\mathrm{tc}} \sum_{(i,j)\in\mathcal{E}_{\mathrm{tc}}} \mathcal{L}_{i,j}(\Theta). \tag{7}$$

It is noteworthy that these operations are not admissible in the Euclidean space, where the local capacity of any $(A, r) \in \mathbb{R}^d \times \mathbb{R}$ is a constant with respect to d.

6 Experiments

We compare our algorithm to the baseline model (Nickel & Kiela, 2017) on the synthetic dataset in Figure 1 and Figure 2. Our method not only achieves perfect MAP (**0.998**) but also yields better reconstructed geometry. We do extensive experiments on multiple real-world datasets of various scales and characteristics in Appendix D. Results show our algorithm consistently outperform the baseline algorithms (Nickel & Kiela, 2017; 2018), especially on extremely bushy datasets.

7 CONCLUSION

In this paper, we analyze three categories of illness and develop a geometry-aware algorithm that targets at reducing these illnesses and improving performance. Our algorithm shows superior performance over baseline models on both synthetic and real world datasets.

REFERENCES

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

Bijan Afsari, Roberto Tron, and René Vidal. On the convergence of gradient descent for finding the riemannian center of mass. *SIAM Journal on Control and Optimization*, 51(3):2230–2260, 2013.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

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

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, pp. 1–9, 2013.

Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*, 2017.

Ines Chami, Adva Wolf, Frederic Sala, and Christopher Ré. Low-dimensional knowledge graph embeddings via hyperbolic rotations. In *Graph Representation Learning NeurIPS 2019 Workshop*, 2019a.

Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32:4869, 2019b.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in neural information processing systems*, pp. 4868–4879, 2019c.

Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. Large-margin classification in hyperbolic space. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1832–1840. PMLR, 2019.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*, pp. 1646–1655. PMLR, 2018a.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. *arXiv preprint arXiv:1804.01882*, 2018b.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *arXiv preprint arXiv:1805.09112*, 2018c.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2018.

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.

Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

Matthew Jenssen, Felix Joos, and Will Perkins. On kissing numbers and spherical codes in high dimensions. *Advances in Mathematics*, 2018.

Grigorii Anatol'evich Kabatiansky and Vladimir Iosifovich Levenshtein. On bounds for packings on a sphere and in space. *Problemy Peredachi Informatsii*, 14(1):3–25, 1978.

Matt Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. Inferring concept hierarchies from text corpora via hyperbolic embeddings. *arXiv preprint arXiv:1902.00913*, 2019.

Matthias Leimeister and Benjamin J Wilson. Skip-gram word embeddings in hyperbolic space. *arXiv preprint arXiv:1809.01498*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, 2011.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pp. 6338–6347, 2017.

Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pp. 3779–3788. PMLR, 2018.

Julien Paupert. Introduction to hyperbolic geometry. *https://math.la.asu.edu/ paupert/HyperbolicGeometryNotes.pdf*, 2016.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 74–84, 2013.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pp. 4460–4469. PMLR, 2018.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pp. 355–366. Springer, 2011.

C. E. Shannon. Probability of error for optimal codes in a gaussian channel. *Bell System Tech. J.*, 1959.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincare glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

Ivan Vulić and Nikola Mrkšić. Specialising word vectors for lexical entailment. *arXiv preprint arXiv:1710.06371*, 2017.

Melanie Weber, Manzil Zaheer, Ankit Singh Rawat, Aditya Menon, and Sanjiv Kumar. Robust large-margin learning in hyperbolic space. *arXiv preprint arXiv:2004.05465*, 2020.

Benjamin Wilson and Matthias Leimeister. Gradient descent in hyperbolic space. *arXiv preprint arXiv:1805.08207*, 2018.

A Related Work

Learning in hyperbolic space is initially proposed by Nickel & Kiela (2017). It is the most related work to our paper. Their method outperforms the Euclidean counterpart in low dimensions as to the task of learning embeddings for edge reconstruction. However, since their algorithm is directly adapted from the Euclidean space, it does not naturally leverage potentially useful geometrical properties of hyperbolic space (see Section 2). As a consequence, there remain many bad cases even after convergence (see Figure 1a).

A series of work directly incorporate properties of hyperbolic space via optimization (Wilson & Leimeister, 2018; Bonnabel, 2013; Absil et al., 2009; Afsari et al., 2013). Specifically, Nickel & Kiela (2018) conduct training in the Lorentz space with a closed-form expression of the geodesics on the hyperbolic manifold. However, since the learning objective equation 2 is highly non-convex, obtaining more accurate gradients does not completely solve the problem.

Another group of work either implement hyperbolic versions of commonly used neural network modules (Ganea et al., 2018c; Gulcehre et al., 2018; Chami et al., 2019b), or design models specifically tailored for hyperbolic space (Vulić & Mrkšić, 2017; Le et al., 2019; Cho et al., 2019; Leimeister & Wilson, 2018; Weber et al., 2020; Chami et al., 2019a). These methods are task-specific and thus expensive to deploy in downstream applications.

Apart from the above learning approaches, Sala et al. (2018) presents a combinatorial algorithm that achieves better performance than Nickel & Kiela (2017; 2018) with even lower dimensions. The core idea is to extend the 2-dimensional results of Sarkar (2011) to arbitrary dimensions. However, this algorithm suffers from three vital weaknesses: (1) it requires complete information of the graph; (2) it is sensitive to addition/removal of data; and (3) most critically, it involves discrete operations and thus does not have gradients. Therefore, in scenarios where complete information is unavailable, the graph dynamically changes, or joint learning is needed, this approach does not suffice.

In this paper, we endorse the importance of leveraging geometrical properties in learning unsupervised hyperbolic embeddings. Based on this intention, we develop a geometry-aware algorithm that improves embedding performances, which, to the best of our knowledge, is original.

B Illness

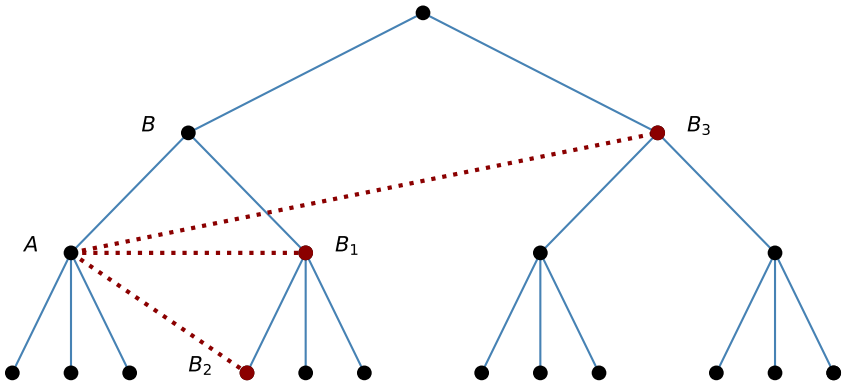


Figure 3: Illustration of the three categories of illness. A is the source node and B is the ground-truth target node. It is called (1) *capacity illness* if A connects to B_1, (2) *intra-subtree illness* if A connects to B_2, and (3) *inter-subtree illness* if A connects to B_3.

C Local Capacity

According to **Definition** 2, for $A \in \mathcal{V}$ and a radius r, if $|\{C : C \text{ is a child of } A, d(A, C) \leq r\}|$ exceeds the local capacity of (A, r), then capacity illness must exist. To obtain bounds on local

capacity, we first bound the r-packing number of a geodesic sphere centered at point $A \in \mathcal{B}$ with radius r.

First, the packing number of the geodesic sphere centered at the origin O is the same as that of A. To see why we use a result from Paupert (2016):

$$\mathbf{Isom}(\mathcal{B}) = \mathbf{M\ddot{o}b}(\mathcal{B}). \tag{8}$$

That is, the group of isometries from $\mathcal{B}$ to itself coincide with the group of all Möbius transformations preserving $\mathcal{B}$. In the Poincaré ball model, $\mathbf{M\ddot{o}b}(\mathcal{B})$ is generated by inversions in generalized spheres $\mathcal{S}'$ such that $\mathcal{S}' \perp \partial\mathcal{B}$, therefore once we extend $\overrightarrow{OA}$ to C with $\|OC\|^2 - 1 = \|OC\| \cdot \|CA\|$, then by taking the restriction in $\mathcal{B}$ of the inversion in generalized sphere $\mathcal{S}'$ centered at C we get an isometry from $\mathcal{B}$ to itself which maps A to O. Specifically, it is an isometry between any geodesic sphere centered at A and the geodesic sphere centered at the origin O (with the same radius r).

Then, we compute the r-packing number of the geodesic sphere S_r centered at O with Poincaré radius r. For $B_1, B_2 \in S_r$, let $u = \overrightarrow{OB_1}, v = \overrightarrow{OB_2}$ and r be the Poincaré norm of u. Then, as long as the angle θ between u and v satisfies

$$\theta \geq \theta_r = 2 \arcsin\left(\frac{1}{2\cosh(r/2)}\right), \tag{9}$$

we have $d(B_1, B_2) \geq d(B_i, O)$, $i = 1, 2$. For not very small r, $\theta_r \approx 2e^{-r/2}$. Then, the r-packing problem is equivalent to evaluating the size of the largest spherical code of angle θ_r in dimension d, defined as $\mathcal{A}(d, \theta_r)$. According to Jenssen et al. (2018), we have

$$\mathcal{A}(d, \theta) \geq (1 + o(1))\frac{c_\theta \cdot d}{s_d(\theta)}, \tag{10}$$

where $c(\theta) = \log \frac{\sin^2(\theta)}{\sqrt{(1-\cos\theta)^2(1+2\cos\theta)}} \approx \log \frac{2}{\sqrt{3}}$ for small θ, $s_d(\theta) = (1 + o(1))\frac{\sin^{d-1}\theta}{\sqrt{2\pi d}\cdot\cos\theta}$. Specifically, when $d \leq 16$ with small θ, Shannon (1959) provides a better bound:

$$\mathcal{A}(d, \theta) \geq \frac{1}{s_d(\theta)} = (1 + o(1))\frac{\sqrt{2\pi d} \cdot \cos\theta}{\sin^{d-1}\theta}. \tag{11}$$

To sum up, for not very small r we have the following lower bounds under different dimensions:

$$\mathcal{A}(d, \theta_r) \gtrapprox \begin{cases} \pi e^{\frac{r}{2}} & d = 2 \\ \sqrt{2\pi d} \cdot 2^{1-d} e^{\frac{d-1}{2}r} & 3 \leq d \leq 16 \\ \sqrt{2\pi}\log\frac{2}{\sqrt{3}} \cdot d^{\frac{3}{2}} \cdot 2^{1-d} e^{\frac{d-1}{2}r} & d \geq 17 \end{cases}. \tag{12}$$

As for the upper bound, according to Kabatiansky & Levenshtein (1978),

$$\mathcal{A}(d, \theta) \leq e^{\phi(\theta)d(1+o(1))}, \tag{13}$$

where $\phi(\theta) > -\log\sin\theta$ is a certain function. Therefore,

$$\mathcal{A}(d, \theta_r) \lessapprox \begin{cases} \pi e^{\frac{r}{2}} & d = 2 \\ 2^d e^{\frac{dr}{2}} & d \geq 3 \end{cases}. \tag{14}$$

Note that by considering the extension of radius, one can define local capacity on geodesic balls instead of spheres(4), which leads to the same conclusion.

D EXPERIMENTAL SETTINGS AND OVERVIEW

We apply our algorithm to both a synthetic dataset and real-world datasets on the graph reconstruction task. We evaluate the performance by mean average precision (MAP) and Mean Rank (MR) defined below. For $A \in \mathcal{V}$ with degree $\deg(A)$ and neighborhood $\mathcal{N}_A = \left\{B_1, ...B_{\deg(A)}\right\}$, let R_{A,B_i} be the smallest subset of $\mathcal{V}$ containing B_i and all points closer to A than B_i. Then, the MAP is defined as

$$\text{MAP}(f) = \frac{1}{|\mathcal{V}|} \sum_{A \in \mathcal{V}} \frac{1}{\deg(A)} \sum_{i=1}^{|\mathcal{N}_A|} \frac{|\mathcal{N}_A \cap R_{A,B_i}|}{|R_{A,B_i}|}. \tag{15}$$

and the MR is defined as

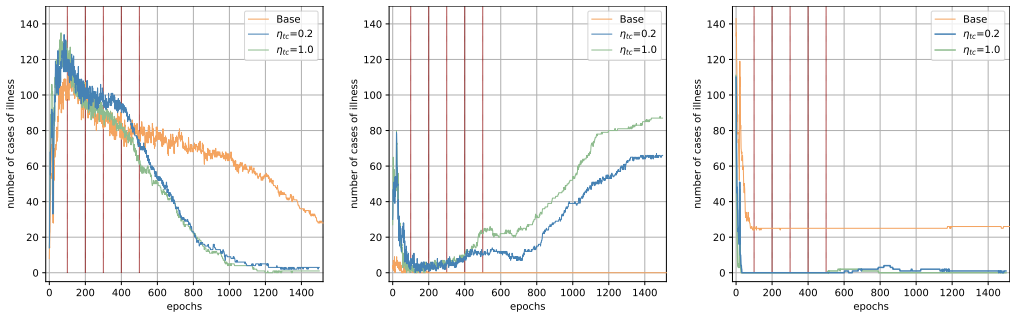
$$\text{MR}(f) = \frac{1}{|\mathcal{V}|} \sum_{A \in \mathcal{V}} \sum_{i=1}^{|\mathcal{N}_A|} \left(|R_{A,B_i}| - i\right). \tag{16}$$

In addition, we report the number of three kinds of illness defined in **Definition** 1 after the algorithm converges.

We run baseline algorithms (Nickel & Kiela, 2017; 2018) and our algorithm on a synthetic dataset, Yelp Challenge (Tree) (see Table 2), WordNet Verbs (Tree) (see Table 3), WordNet Nouns (Tree) (see Table 4), Commodity Catalog (Tree) (see Table 5) and WordNet Nouns (Closure) (see Table 6). We report the reconstruction Mean Rank, MAP, number of capacity errors, number of intra-subtree errors and number of inter-subtree errors respectively.

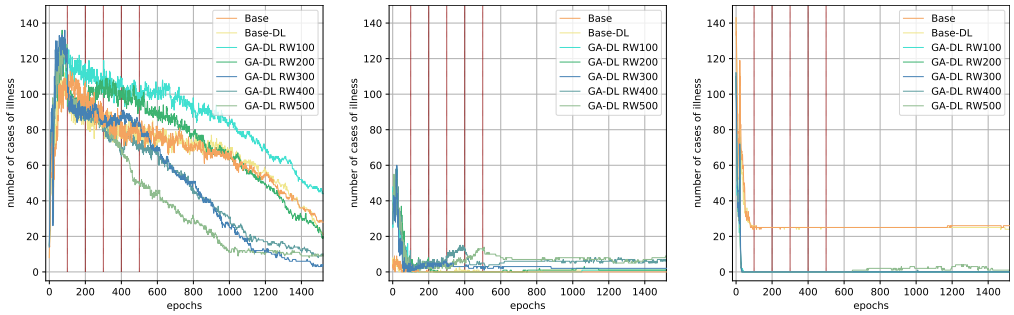
A key point to notice is that we focus on tree datasets instead of general DAG or transitive closures of trees. In terms of the objective (reducing MAP and MR), learning a tree structure is much harder because the size of the neighborhood set is as few as one. Experiments validate this statement: we apply the baseline algorithm (Nickel & Kiela, 2017) to the same WordNet Noun Hierarchy dataset (Nickel & Kiela, 2017) where the transitive closure edges are removed. The performances in terms of MAP and MR significantly drop compared to the numbers reported in (Nickel & Kiela, 2017) (See Table 4).

E SYNTHETIC DATASET EXPERIMENTS

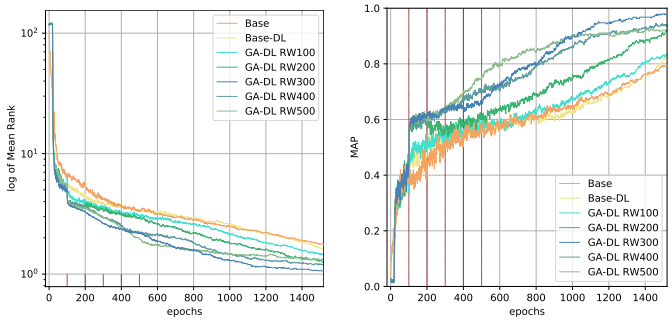


(a) Capacity illness (b) Intra-subtree illness (c) Inter-subtree illness

Figure 4: Number of different kinds of illness under different η_{tc}. Base denotes the baseline algorithm.



(a) Capacity illness (b) Intra-subtree illness (c) Inter-subtree illness



(d) log MR (e) MAP

Figure 5: Performances of our algorithm under different hyperparameter settings on the synthetic tree. Base denotes the baseline algorithm, GA is our algorithm where DL is the dilation operation, RW is the re-weighting strategy followed by the threshold epoch number N_{tc}.

We synthesize a balanced tree T of 5 layers including the root node. Each non-leaf node in T has 5 children. The edges are directed, pointing from children to parents. We illustrate our algorithm on this synthetic dataset.

We compare our algorithm (Algorithm 1) to the baseline Poincaré embedding algorithm (Nickel & Kiela, 2017). In both algorithms, we set the dimension to be 2, learning rate to be 0.5, batch size to be 50, and the number of negative samples m to be 50.

The comparison between learning procedures is presented in Figure 2. As demonstrated, our algorithm learns visually more balanced embeddings with less illness. Quantitative results of the comparison, including the number of illness, MAP and MR, are presented in Figure 4 and Figure 5. We compare

different N_{tc} in Figure 5 and different η_{tc} in Figure 4. Our algorithm produces less illness than the baseline algorithm and achieves the highest MAP and MR.

Figure 4 shows our explorations on how to use transitive closure edges.They tend to increase the overall effective gradient magnitude and draw vertices of a same subtree tightly together. Therefore, this could help reduce capacity illness quickly (See Figure 4 (a)) and eliminate inter-subtree illness (See Figure 4 (c)). However, it might also confuse the ground-truth tree edges with the added ones, thus increasing intra-subtree illness (See Figure 4 (b)). When we assign weights to the transitive closure edges, this side effect is mitigated: $\eta_{tc} = 0.2$ yields the best results.

Figure 5 shows our explorations on how to use dilation and reweighting. These two operations should be applied after certain epochs of training so that 1) the subtrees are pushed relatively far from each other [1] to ensure the dilation operation will push vertices in the appropriate directions, and 2) before the vertices are already pushed to places with sufficient capacity. Empirically we find this threshold epoch number $N_{tc} = 300$ yields the best results.

[1] A similar idea is shown in the burn-in stage of (Nickel & Kiela, 2017).

F REAL-WORLD DATASET EXPERIMENTS

F.1 DATASET STATISTICS

Table 1 displays statistics of several datasets used in our experiments (including the Yelp Challenge Dataset [2], the Commodity Catalog Dataset, and the WordNet dataset (Miller, 1998)). We report the number of nodes $|\mathcal{V}|$, the number of edges $|\mathcal{E}|$, maximum degree, variance of all degrees, and tree depths. We make subjective remarks to their size and characteristics.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Max deg	Std of deg	Depth	Size	Characteristics
Yelp Challenge (T)	1587	1586	194	7.9	4	small	shallow
Commodity Catalog (T)	134812	134811	726	26.4	6	large	shallow; bushy
WordNet Verbs (T)	13542	13541	361	9.1	12	medium	deep
WordNet Nouns (T)	82115	82114	666	6.6	15	large	deep
WordNet Nouns (C)	82115	769130	82114	913.6	15	large	deep; dense

Table 1: Dataset statistics (T: Tree; C: Closure)

F.2 YELP CHALLENGE DATASET (TREE) RESULTS

The results for Yelp Challenge (Tree) are illustrated in Table 2. In all algorithms, we set the learning rate to be 1.0, batch size to be 10, and the number of negative samples m to be 50.

		Dim	MR	MAP	Capacity	Intra	Inter
YELP CHALLENGE **Reconstruction**	**Euclidean**	2	27.612	0.107	1440	82	**28**
	Nickel & Kiela (2017)		1.681	0.855	292	**45**	64
	Nickel & Kiela (2018)		1.520	0.894	163	47	74
	GA-DL (Ours)		1.351	**0.926**	**86**	65	47
	GA-DL-RW (Ours)		**1.202**	0.914	162	54	37
	Euclidean	5	11.320	0.326	1236	9	44
	Nickel & Kiela (2017)		1.063	0.988	**1**	**0**	28
	Nickel & Kiela (2018)		1.101	0.984	**1**	11	26
	GA-DL (Ours)		1.062	0.987	3	9	20
	GA-DL-RW (Ours)		**1.030**	**0.989**	3	9	**18**
	Euclidean	10	1.528	0.910	183	1	14
	Nickel & Kiela (2017)		1.042	0.990	**0**	**0**	25
	Nickel & Kiela (2018)		1.064	0.987	**0**	9	23
	GA-DL (Ours)		1.051	0.989	1	5	21
	GA-DL-RW (Ours)		**1.015**	**0.993**	5	6	**9**

Table 2: Yelp Challenge Dataset

[2] https://www.yelp.com/dataset/documentation/main

F.3 WORDNET VERBS DATASET (TREE) RESULTS

The results for WordNet Verbs (Tree) are illustrated in Table 3. In all algorithms, we set the learning rate to be 1.0, batch size to be 10, and the number of negative samples m to be 50.

		Dim	MR	MAP	Capacity	Intra	Inter
WORDNET VERBS **Reconstruction**	**Euclidean**	2	66.302	0.124	7366	3778	1879
	Nickel & Kiela (2017)		8.088	0.521	4826	2582	1683
	Nickel & Kiela (2018)		6.875	0.448	5459	**1769**	2812
	GA-DL (Ours)		8.319	0.510	5023	2361	1908
	GA-DL-RW (Ours)		**3.559**	**0.563**	**4543**	2741	**1399**
	Euclidean	5	21.245	0.295	10383	**1273**	**277**
	Nickel & Kiela (2017)		2.329	0.854	235	2321	454
	Nickel & Kiela (2018)		2.389	0.853	285	2277	468
	GA-DL (Ours)		2.195	**0.878**	**211**	1443	920
	GA-DL-RW (Ours)		**1.722**	0.841	378	1306	1893
	Euclidean	10	6.240	0.677	4205	1424	300
	Nickel & Kiela (2017)		1.950	0.856	232	2322	457
	Nickel & Kiela (2018)		1.945	0.855	257	2338	**441**
	GA-DL (Ours)		**1.654**	**0.884**	**184**	1456	828
	GA-DL-RW (Ours)		**1.654**	0.842	241	**1255**	2110

Table 3: WordNet Verbs

F.4 WORDNET VERBS NOUNS (TREE) RESULTS

The results for WordNet Nouns (Tree) are illustrated in Table 4. In all algorithms, we set the learning rate to be 1.0, batch size to be 50, and the number of negative samples m to be 50.

		Dim	MR	MAP	Capacity	Intra	Inter
WORDNET NOUNS **Reconstruction**	**Euclidean**	5	151.321	0.235	46563	21942	2993
	Nickel & Kiela (2017)		71.271	0.322	48844	**17293**	438
	Nickel & Kiela (2018)		74.625	0.230	50950	18777	311
	GA-DL (Ours)		19.313	0.481	35119	21158	444
	GA-DL-RW (Ours)		**2.869**	**0.697**	**13394**	23907	**27**
	Euclidean	10	23.113	0.278	23978	31018	15745
	Nickel & Kiela (2017)		41.014	0.324	49152	**17276**	154
	Nickel & Kiela (2018)		38.395	0.241	50986	18690	100
	GA-DL (Ours)		7.900	0.754	10647	13547	5551
	GA-DL-RW (Ours)		**2.738**	**0.722**	**9483**	24844	**32**

Table 4: WordNet Nouns

F.5 COMMODITY CATALOG (TREE) RESULTS

The results for Commodity Catalog (Tree) are illustrated in Table 5. In all algorithms, we set the learning rate to be 1.0, batch size to be 10, and the number of negative samples m to be 50.

		Dim	MR	MAP	Capacity	Intra	Inter
COMMODITY CATALOG **Reconstruction**	**Euclidean**	5	163.044	0.024	132276	**1480**	700
	Nickel & Kiela (2017)		86.759	0.063	130479	2001	174
	Nickel & Kiela (2018)		50.393	0.082	127437	1968	238
	GA-DL (Ours)		5.405	**0.745**	**49837**	1971	301
	GA-DL-RW (Ours)		**2.951**	0.683	56922	7711	**26**
	Euclidean	10	59.133	0.052	131806	**1676**	289
	Nickel & Kiela (2017)		67.283	0.071	130212	1998	147
	Nickel & Kiela (2018)		36.836	0.112	124366	1964	196
	GA-DL (Ours)		2.167	**0.978**	**1871**	1861	390
	GA-DL-RW (Ours)		**2.015**	0.881	19902	5525	**26**

Table 5: Commodity Catalog

The Commodity Catalog (Tree) dataset is generated from real-world e-commerce data. It is extremely bushy. According to Sala et al. (2018) the hyperbolic space is capable to embed even extremely bushy trees. However, we find the baseline algorithm in Nickel & Kiela (2017) can not fully exert such capability. This is because a bushy tree requires large local capacity, but the baseline algorithm takes very long time to reach such capacity and instead easily gets stuck at local optimum. Experimentally, the baseline algorithm does not perform well on such bushy dataset, while our algorithm with simple dilation yields significantly better results [3].

F.6 WORDNET NOUNS (CLOSURE) RESULTS

The results for WordNet Nouns (Closure) are illustrated in Table 6. In all algorithms, we set the learning rate to be 1.0, batch size to be 50, and the number of negative samples m to be 50. [4]

		Dim	MR	MAP	Capacity	Intra	Inter
WORDNET NOUNS CLOSURE **Reconstruction**	**Nickel & Kiela (2017)**	10	4.736	0.772	46192	151324	**9942**
	GA-DL (Ours)	10	4.788	0.781	41756	144358	11987
	GA-DL-RW (Ours)	10	**4.270**	**0.797**	**38277**	**134541**	12575

Table 6: WordNet Nouns Closure

[3] Noticeably, since the tree is bushy, adding transitive closure edges would largely increase intra-class illness, which makes MAP drop.

[4] Results of Nickel & Kiela (2017) are based on their official implementation and hyperparameters in `https://github.com/facebookresearch/poincare-embeddings`.