Generalization Error Bounds for Graph Embedding Using Negative Sampling: Linear vs Hyperbolic

Atsushi Suzuki
University of Greenwich
London, United Kingdom
atsushi.suzuki.rd@gmail.com

Atsushi Nitanda
Kyushu Institute of Technology
Fukuoka, Japan
nitanda@ai.kyutech.ac.jp

Jing Wang*
University of Greenwich
London, United Kingdom
jing.wang@greenwich.ac.uk

Linchuan Xu
The Hong Kong Polytechnic University
Hong Kong SAR, China
linch.xu@polyu.edu.hk

Kenji Yamanishi
The University of Tokyo
Tokyo, Japan
yamanishi@g.ecc.u-tokyo.ac.jp

Marc Cavazza
National Institute of Informatics
Tokyo, Japan
marc.cavazza@gmail.com

Abstract

Graph embedding, which represents real-world entities in a mathematical space, has enabled numerous applications such as analyzing natural languages, social networks, biochemical networks, and knowledge bases. It has been experimentally shown that graph embedding in hyperbolic space can represent hierarchical tree-like data more effectively than embedding in linear space, owing to hyperbolic space's exponential growth property. However, since the theoretical comparison has been limited to ideal noiseless settings, the potential for the hyperbolic space's property to worsen the generalization error for practical data has not been analyzed. In this paper, we provide a generalization error bound applicable for graph embedding both in linear and hyperbolic spaces under various negative sampling settings that appear in graph embedding. Our bound states that error is polynomial and exponential with respect to the embedding space's radius in linear and hyperbolic spaces, respectively, which implies that hyperbolic space's exponential growth property worsens the error. Using our bound, we clarify the data size condition on which graph embedding in hyperbolic space can represent a tree better than in Euclidean space by discussing the bias-variance trade-off. Our bound also shows that imbalanced data distribution, which often appears in graph embedding, can worsen the error.

1 Introduction

Graphs are a fundamental formulation of real-world entities and their relations, such as words in natural languages, people in social network, and objects in knowledge bases. Here, the vertices and edges of a graph correspond to the entities and the relations among them, respectively. Based on the formulation, graph embedding has enabled numerous applications for those data, such as machine translation and sentiment analysis for natural language [1, 2, 3, 4], and community detection and

*Corresponding author

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

link prediction for social network data [5, 6, 7, 8, 9], pathway prediction of biochemical network [10, 11], and link prediction and triplet classification for knowledge base [12, 13, 14, 15, 16, 17]. Graph embedding produces representations of a graph's vertices in a space equipped with a function that defines the dissimilarity between two points. In this paper, we call a function that defines the dissimilarity a *dissimilarity function* and a space equipped with a dissimilarity function a *dissimilarity space*. For example, we can consider the squared distance as a dissimilarity function. Graph embedding aims to obtain representations such that the dissimilarity function reflects the relations defined by the edges. Specifically, we expect that the dissimilarity function returns a small value for the representations of a *positive pair*, a pair of vertices connected by an edge, and a large value for a *negative pair*, a pair not connected. Here, to reduce the computational cost, obtaining training data by the *negative sampling* strategy has been known to be effective [1]. In this strategy, we sample a positive pair for each iteration, followed by sampling negative pairs around the positive pair.

As a dissimilarity space, many graph embedding methods have used linear space equipped with an inner product function [1, 2, 3, 5, 6, 7, 8, 9], which we call *linear graph embedding (LGE)*. However, linear space has limitations in representing data with a hierarchical tree-like structure [18, 19, 20, 21]. These limitations are due to linear space's polynomial growth property, which means that the volume or surface of a ball in linear space grows polynomially with respect to its radius. This linear space's growth speed is significantly slower than embedding hierarchical data such as an r-ary tree ($r \geq 2$) requires, which is exponential [21]. To overcome this limitation, graph embedding in hyperbolic space has recently attracted much attention [20, 22, 23, 24, 25, 26, 4, 27], which we call *hyperbolic graph embedding (HGE)* in this paper. In contrast to linear space's polynomial growth property, hyperbolic space has the exponential growth property, that is, the volume of any ball in hyperbolic space grows exponentially with respect to its radius [18, 19, 20, 21]. As a result, hyperbolic space is almost tree-like in that it can be well approximated by a tree [28], and we can embed any tree in hyperbolic space with arbitrarily low distortion [29]. Existing HGE papers have experimentally shown HGE's ability to effectively represent hierarchical tree-like data such as taxonomies and social networks. However, the theoretical guarantee of HGE's performance is limited to ideal noiseless settings [28, 29, 23], and the comparison between LGE and HGE's generalization performance in noisy settings has not been discussed, although HGE could have a much worse generalization error than LGE in compensation for hyperbolic space's exponential growth property and cause overfitting for real data, which are often noisy.

In this paper, we derive a generalization bound for graph embedding using the negative sampling strategy under noisy settings. To the best of our knowledge, this is the first work that derives a complete generalization error bound for both LGE and HGE. As discussed in [21], since the generalization error of a learning model reflects the volume of its hypothesis space, we can conjecture that the generalization errors of LGE and HGE are polynomial and exponential with respect to the embedding space's radius, reflecting inner product space and hyperbolic space's polynomial and exponential growth property. Also, an imbalanced data distribution, which often appears in graph embedding reflecting the graph structure's imbalance, may worsen the error. In this paper, we formally prove that the above conjectures are true, as well as clarify the dependency of the error bound on the number of entities and the size of training data. Based on the derived generalization error bounds, we also clarify the data size condition on which HGE outperforms LGE in embedding a tree, by discussing the bias-variance trade-off.

To derive a generalization error bound in embedding problem, existing papers [30, 21] deriving ordinal embedding's bounds have converted the problem into a linear prediction problem to calculate its Rademacher complexity [31, 32, 33]. Also, for hyperbolic embedding, the decomposition of the Lorentz Gramian matrix [34] has been combined with the above technique [21]. For graph embedding using negative sampling, however, we cannot straightforwardly apply these techniques, which have been effective for deriving ordinal embedding's generalization error bound. Since the data distribution depends on the graph structure and positive sampling affects the distribution of negative sampling in the negative sampling structure, we cannot apply the i.i.d.-uniform-distribution-based discussions in [30, 21] to graph embedding using the negative sampling strategy. Although a recent unpublished paper [35] has attempted to derive a generalization error bound only for LGE, which is incomplete in that the bound still has an unevaluated part and cannot be applied for noisy settings, this dependency between the distributions has been ignored. We solve this problem by decomposing the loss function into functions of each edge sampled by the negative sampling strategy. We achieved this decomposition by our novel multivariable version of the Ledoux-Talagrand contraction lemma

[36]. By our approach, we can upper-bound the Rademacher complexity of a graph embedding loss function by the sum of the Rademacher complexities [31, 32, 33] of linear prediction models, which have been calculated in [30, 21]. As a result, our generalization bound is valid for various negative sampling settings where the distribution of positive and negative edges are dependent.

Our contributions are threefold:

- We have derived the generalization error bound for negative-sampling-based graph embedding. Our theorem is applicable to various settings regarding the embedding space, data distribution and loss function, such as LGE and HGE, the dependency between the positive pairs and negative pairs' distribution, and sigmoid loss functions. Our upper bound shows that LGE and HGE cause a polynomial and exponential error with respect to the embedding space's radius, respectively. Our bound also shows that imbalanced data distribution can worsen the error.
- We have derived specific error bounds for practical negative sampling strategies.
- We have derived an explicit training data size condition on which HGE can represent a tree better than LGE.

2 Preliminary

Notation In this paper, the symbol $:=$ is used to state that its left hand side is defined by its right hand side. We denote by $\mathbb{Z}, \mathbb{Z}_{>0}, \mathbb{R}, \mathbb{R}_{\geq 0}$ the set of integers, the set of positive integers, the set of real numbers, and the set of non-negative real numbers, respectively. Suppose that $D, V \in \mathbb{Z}_{>0}$. We denote by , $\mathbb{R}^D$ the set of D-dimensional real vectors. For a matrix $\boldsymbol{A} \in \mathbb{R}^{D,V}$, we denote by $[\boldsymbol{A}]_{d,v}$ the element in the d-row and the v-th column. For a vector $\boldsymbol{x} \in \mathbb{R}^D$, we denote by $\|\boldsymbol{x}\|_2$ the 2-norm of $\boldsymbol{x}$, defined by $\|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$, and for a matrix $\boldsymbol{A} \in \mathbb{R}^{D,V}$, we denote by $\|\boldsymbol{A}\|_{\mathrm{op},2}$, the operator norm of $\boldsymbol{A}$ with respect to the 2-norm, defined by $\|\boldsymbol{A}\|_{\mathrm{op},2} := \sup_{\boldsymbol{x} \in V, \boldsymbol{x} \neq \boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}$. By $(a_n)_{n=1}^N$, we denote a sequence $(a_1, a_2, \ldots, a_N)$.

2.1 Graph Embedding

In this section, we first formulate the general embedding problem, before we specialize it into a graph embedding. Consider an entity set $\mathcal{V}$ and the *true dissimilarity function* $\mathcal{V} \times \mathcal{V} \to \mathbb{R}$ defined on the entity set $\mathcal{V}$. For $i, j \in \mathcal{V}$, we call $\delta^*(i, j)$ the *true dissimilarity* between i and j, and a small $\delta^*(i, j)$ value implies that entity i and j are similar or closely related to each other, and large $\delta^*(i, j)$ value implies its converse. In this paper, we identify $\mathcal{V}$ with the integer set $\{1, 2, \ldots, |\mathcal{V}|\}$. Embedding aims to get representations $z_1, z_2, \ldots, z_V$ of the entity set $\mathcal{V}$ in a space $(\mathcal{Z}, \delta_{\mathcal{Z}})$ equipped with a dissimilarity function $\delta_{\mathcal{Z}} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ of $\mathcal{Z}$ so that the representations are consistent to the true dissimilarity among the entities in that if $\delta^*(i, j)$ is small, then $\delta(z_i, z_j)$ is also small, and vice versa. We call $\mathcal{Z}$ the *embedding space*. Specifically, we aim to satisfy

$$\delta^*(i, j) \lessgtr \theta^* \Leftrightarrow \delta_{\mathcal{Z}}(z_i, z_j) \lessgtr \theta_{\mathcal{Z}}, \tag{1}$$

as frequent as possible with respect to some distribution regarding (i, j), which we discuss in the next subsection. Here, $\theta^*, \theta_{\mathcal{Z}} \in \mathbb{R}_{\mathcal{Z}}$ are fixed thresholds regarding the true dissimilarity δ^* and the dissimilarity $\delta_{\mathcal{Z}}$ in the embedding space.

As a dissimilarity function of the embedding space $\mathcal{Z}$, we mainly consider the square distance $[\Delta_{\mathcal{Z}}(z_i, z_j)]^2$ if $\mathcal{Z}$ is equipped with a distance function $\Delta_{\mathcal{Z}} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$, and the negative inner product $-\langle z_i, z_j \rangle$ if $\mathcal{Z}$ is equipped with an inner product $\langle \cdot, \cdot \rangle : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$. In this paper, we deal with the following four dissimilarity spaces.

Definition 1. (Dissimilarity spaces)

(a) The D-dimensional Euclidean space $\left(\mathbb{R}^D, \Delta_{\mathbb{R}^D}\right)$ consists of the set of D-dimensional real vectors and the distance function $\Delta_{\mathbb{R}^D} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ defined by $\Delta_{\mathbb{R}^D}(\boldsymbol{z}, \boldsymbol{z}') := \|\boldsymbol{z} - \boldsymbol{z}'\|_2^2$. The dissimilarity function is given by $\delta_{\mathbb{R}^D}(\boldsymbol{z}, \boldsymbol{z}') := [\Delta_{\mathbb{R}^D}(\boldsymbol{z}, \boldsymbol{z}')]^2$.

(b) The D-dimensional hyperbolic space $\left(\mathbb{L}^D, \Delta_{\mathbb{L}^D}\right)$ consists of the D-dimensional hyperboloid $\mathbb{L}^D \subset \mathbb{R}^{D+1}$ and the distance function $\Delta_{\mathbb{L}^D}$ defined by

$$\mathbb{L}^D \coloneqq \left\{\boldsymbol{x} \in \mathbb{R}^{D+1} \;\middle|\; \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathrm{M}} = -1\right\}, \qquad \Delta_{\mathbb{L}^D}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \operatorname{arcosh}(-\langle \boldsymbol{x}, \boldsymbol{x}' \rangle_{\mathrm{M}}), \tag{2}$$

where $\langle \cdot, \cdot \rangle_{\mathrm{M}} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is the Minkowski inner prodct defined by $\left\langle \begin{bmatrix} z_0 & z_1 & \cdots & z_D \end{bmatrix}^\top, \begin{bmatrix} {z'}_0 & {z'}_1 & \cdots & {z'}_D \end{bmatrix}^\top \right\rangle_{\mathrm{M}} \coloneqq -z^0 {z'}^0 + \sum_{d=1}^D z_d {z'}_d$. The dissimilarity function is given by $\delta_{\mathbb{L}^D}(\boldsymbol{z}, \boldsymbol{z}') \coloneqq \left[\Delta_{\mathbb{L}^D}(\boldsymbol{z}, \boldsymbol{z}')\right]^2$.

(c) The D-dimensional sphere $\left(\mathbb{S}^D, \Delta_{\mathbb{S}^D}\right)$ consists of the subset $\mathbb{S}^D \subset \mathbb{R}^{D+1}$ and the distance function $\Delta_{\mathbb{L}^D}$ defined by

$$\mathbb{S}^D \coloneqq \left\{\boldsymbol{x} \in \mathbb{R}^{D+1} \;\middle|\; \boldsymbol{x}^\top \boldsymbol{x} = 1\right\}, \qquad \Delta_{\mathbb{S}^D}(\boldsymbol{x}, \boldsymbol{x}') \coloneqq \arccos\left(\boldsymbol{x}^\top \boldsymbol{x}'\right). \tag{3}$$

The dissimilarity function is given by $\delta_{\mathbb{S}^D}(\boldsymbol{z}, \boldsymbol{z}') \coloneqq \left[\Delta_{\mathbb{S}^D}(\boldsymbol{z}, \boldsymbol{z}')\right]^2$.

(d) The canonical D-dimensional inner product space $\left(\mathbb{I}^D, \delta_{\mathbb{I}^D}\right)$ as a dissimilarity space consists of the set $\mathbb{I}^D = \mathbb{R}^D$ of D-dimensional real vectors and the dissimilarity function $\delta_{\mathbb{I}^D} : \mathbb{I}^D \times \mathbb{I}^D \to \mathbb{R}$ defined by the negative canonical inner product $\delta_{\mathbb{I}^D}(\boldsymbol{z}, \boldsymbol{z}') \coloneqq -\boldsymbol{z}^\top \boldsymbol{z}'$.

Our main focus in this paper is $\mathbb{R}^D$ and $\mathbb{L}^D$, although our bound is also applicable to $\mathbb{S}^D$ and $\mathbb{I}^D$.

Remark 1. There are multiple models to represent the above spaces. For example, we can use the Poincaré ball model, upper half space model, Klein ball model to represent hyperbolic space, other than the hyperboloid model used in Definition 1. While these are isometric to each other, we used the models in Definition 1 because we can formulate the dissimilarity function as a simple function of a linear combination of inner products $\boldsymbol{z}^\top \boldsymbol{z}$, $\boldsymbol{z}^\top \boldsymbol{z}'$, and ${\boldsymbol{z}'}^\top \boldsymbol{z}'$, or $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle_{\mathrm{M}}$, which makes it easy to apply techniques in [30, 21].

Graph Embedding Consider a graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the edge set. Here, we only consider undirected graphs, and thus assume that if $(i, j) \in \mathcal{E}$ then $(j, i) \in \mathcal{E}$ holds. Graph embedding for a graph $(\mathcal{V}, \mathcal{E})$ is a special case of embedding problem defined above, where the vertices $\mathcal{V}$ are the entities, the true dissimilarity between two entities i, j is given by the graph distance defined by

$$\delta^*(i, j) \coloneqq \min\left\{K \;\middle|\; \exists (v_1, v_2, \ldots, v_{K-1}) \in \mathcal{V}^{K-1}, (i, v_1), (v_1, v_2), \ldots, (v_{K-1}, j) \in \mathcal{E}.\right\}, \tag{4}$$

and $\theta^* = 1$. Then, the objective defined by (1) is equivalent to

$$(i, j) \in \mathcal{E} \Leftrightarrow \delta_{\mathcal{Z}}(z_i, z_j) \leq \theta_z. \tag{5}$$

We say that a pair (i, j) is *truly positive* if $(i, j) \in \mathcal{E}$ and *truly negative* otherwise.

In the following, we denote by $\deg(i)$ the degree of $i \in \mathcal{V}$, defined by $\deg(i) \coloneqq \{j | (i, j) \in \mathcal{E}\}$.

2.2 Data distribution

In this paper, we consider negative-sampling-based training data and loss functions. Training data consist of M positive-negative pair sequences. The m-th positive-negative pair sequence $s_m = (s_m^+, s_m^-)$ consists of K^+ positive pairs $s_m^+ \coloneqq \left(\left(i_{1,m}^+, j_{1,m}^+\right), \left(i_{2,m}^+, j_{2,m}^+\right), \ldots, \left(i_{K^+,m}^+, j_{K^+,m}^+\right)\right) \in \left(\mathcal{V} \times \mathcal{V}\right)^{K^+}$ and K^- negative pairs $s_m^- \coloneqq \left(\left(i_{1,m}^-, j_{1,m}^-\right), \left(i_{2,m}^-, j_{2,m}^-\right), \ldots, \left(i_{K^-,m}^-, j_{K^-,m}^-\right)\right) \in \left(\mathcal{V} \times \mathcal{V}\right)^{K^-}$. Here, for $m = 1, 2, \ldots, M$, we expect that $i_{k,m}^+$ and $j_{k,m}^+$ are similar to each other, that is, $\delta^*\left(i_{k,m}^+, j_{k,m}^+\right) \leq \theta^*$ is valid for $k^+ = 1, 2, \ldots, K^+$, and conversely, $i_{k,m}^-$ and $j_{k,m}^-$ are dissimilar to each other, that is, $\delta^*\left(i_{k,m}^-, j_{k,m}^-\right) > \theta^*$ is valid for $k^- = 1, 2, \ldots, K^-$, although these rules do not always hold owing to noise or the sampling strategy aiming to save the computational cost. To derive a meaningful generalization error bound, an assumption on the distribution of the training data is needed. We consider the following weak assumption.

Assumption 1. $s_1, s_2, \ldots, s_M$ are independently and identically distributed.

Remark 2. Assumption 1 does NOT imply the independence and identity of the distribution of pairs in s_m^+ and s_m^-. For example, the distribution of $\left(i_{k',m}^-, j_{k',m}^-\right)$ may depend on $\left(i_{k,m}^-, j_{k,m}^-\right)$. This weakness assumption allows us to discuss practical negative sampling settings where negative pairs' distributions depend on positive pairs.

We give some training data distribution examples, which may be noisy. In the following, $\mathbb{P}\Big[\left(i_{k,m}^+, j_{k,m}^+\right)\Big]$ and $\mathbb{P}\Big[\left(i_{k,m}^-, j_{k,m}^-\right)\Big]$ denote the probability of edge $\left(i_{k,m}^+, j_{k,m}^+\right)$ and $\left(i_{k,m}^-, j_{k,m}^-\right)$ being generated as the k-th positive and negative pairs, respectively, and $\mathbb{P}\Big[\left(i_{k,m}^-, j_{k,m}^-\right)\Big|\left(i_{k',m}^+, j_{k',m}^+\right)\Big]$ denotes the probability of edge $\left(i_{k,m}^-, j_{k,m}^-\right)$ being generated as the k-th negative pair given $\left(i_{k',m}^+, j_{k',m}^+\right)$ being generated as the k'-th positive pair.

Example 1. (Data Distribution)

(a) (Simple positive-negative sampling) First, we consider the simplest case. Regarding the positive pair generation, suppose that all the pairs (i, j) in the edge set $\mathcal{E}$ have the same probability of being generated as a positive pair, as a simple case. Also, suppose that all the pairs (i, j) not in the edge set have a possibly non-zero probability that is lower than that for the positive pairs, which means the existence of noise. Here, we assume that each pair not in the edge set have the same probability, that is, the following holds:

$$\mathbb{P}\Big[\left(i_{k,m}^+, j_{1,m}^+\right)\Big] = \begin{cases} p^+ & \text{if } \left(i_{k,m}^+, j_{k,m}^+\right) \in \mathcal{E}, \\ r^+ p^+ & \text{if } \left(i_{k,m}^+, j_{k,m}^+\right) \notin \mathcal{E} \text{ and } i_{k,m}^+ \neq j_{k,m}^+, \\ 0 & \text{if } i_{k,m}^+ = j_{k,m}^+, \end{cases} \tag{6}$$

where $r^+ \in [0, 1]$ indicates the noise intensity. If $r = 0$, then only truly positive pairs appear, and if $r = 1$, then all edges appears in the same probability. Here, since $\sum_{(i_{1,m}^+, j_{1,m}^+) \in \mathcal{V} \times \mathcal{V}} \mathbb{P}\big[(i_{1,m}^+, j_{1,m}^+)\big] = 1$, we have that $p^+ = \frac{1}{(1-r^+)|\mathcal{E}| + r^+ |\mathcal{V}|(|\mathcal{V}|-1)}$. For negative pair sampling, we consider the following simple distribution.

$$\mathbb{P}\Big[\left(i_{k,m}^-, j_{1,m}^-\right)\Big] = \begin{cases} p^- & \text{if } \left(i_{k,m}^-, j_{k,m}^-\right) \notin \mathcal{E} \text{ and } i_{k,m}^- \neq j_{k,m}^-, \\ r^- p^- & \text{if } \left(i_{k,m}^-, j_{k,m}^-\right) \in \mathcal{E}, \\ 0 & \text{if } i_{k,m}^- = j_{k,m}^-, \end{cases} \tag{7}$$

where $r^- \in [0, 1]$ indicates the noise intensity. Also, p^- is given by $p^- = \frac{1}{(1-r^-)|\mathcal{V}|(|\mathcal{V}|-1) + r^- |\mathcal{E}|}$.

(b) (Skipgram [1] type negative sampling) In some applications, it is not easy to sample truly negative pairs effectively. In this case, more effective but inaccurate methods are often used. In the following, we explain a negative sampling strategy in [1], a representative on in such methods. Let $K^+ = 1$, and consider the positive pair sampling strategy again (6). Based on this simple setting, we consider the negative sampling strategy in [1]. Here, one vertex of a negative pair $i_{k,m}^-$ is always the same as that of the positive pair $i_{1,m}^+$, and the other entity of the negative pair is generated according to the distribution whose probability mass function is proportional to a value $\pi\big(U\big(j_{1,m}^-\big)\big)$, where $\pi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a function and $U\big(j_{1,m}^-\big)$ is the frequency of the vertex $j_{1,m}^-$ appearing in a positive pair. In summary, the conditional distribution is given by

$$\mathbb{P}\Big[\left(i_{k,m}^-, j_{k,m}^-\right)\Big|\left(i_{1,m}^+, j_{1,m}^+\right)\Big] = \begin{cases} q\pi\big(U\big(j_{1,m}^-\big)\big) & \text{if } i_{k,m}^- = i_{1,m}^+, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Here, since $\sum_{j_{k,m}^- \in \mathcal{V}} \mathbb{P}\Big[\left(i_{k,m}^-, j_{k,m}^-\right)\Big|\left(i_{1,m}^+, j_{1,m}^+\right)\Big] = 1$, we have $q = \frac{1}{\sum_{j_{k,m}^- \in \mathcal{V}} \pi\left(U\left(j_{1,m}^-\right)\right)}$. If $\mathbb{P}\big[(i_{1,m}^+, j_{1,m}^+)\big]$ is given by (6), $U\big(j_{1,m}^-\big)$ is given by $U\big(j_{1,m}^-\big) = \frac{\left(1-r^+\right)\deg\left(j_{1,m}^-\right) + r^+(|\mathcal{V}|-1)}{(1-r^+)|\mathcal{E}| + r^+|\mathcal{V}|(|\mathcal{V}|-1)}$. In the following, we set $\pi(x) = x$ for simplicity. Then we have $\mathbb{P}\Big[\left(i_{k,m}^-, j_{k,m}^-\right)\Big|\left(i_{1,m}^+, j_{1,m}^+\right)\Big] = U\big(j_{1,m}^-\big)$. The above setting is an example of the negative pair's distribution depending on the positive pair. Note that the above setting does not guarantee that truly negative pairs appear as a negative pair more frequently than truly positive pairs.

We remark that this paper's discussion is not limited to the above examples, and Assumption 1 is the only assumption for our main theorem.

2.3 Loss function

To quantify the consistency of the true dissimilarities of entities defined by δ^* and those of representations defined by $\delta_{\mathcal{Z}}$, we consider loss function $l : \mathbb{R}^{K^+} \times \mathbb{R}^{K^-} \to \mathbb{R}_{\geq 0}$. The loss on a positive-negative pair sequence s_m is given by $l\big(\boldsymbol{\delta}_m^+, \boldsymbol{\delta}_m^-\big)$, where

$$\boldsymbol{\delta}_m^\bullet \coloneqq \begin{bmatrix} \delta_{i_{1,m}^\bullet, j_{1,m}^\bullet} & \delta_{i_{2,m}^\bullet, j_{2,m}^\bullet} & \cdots & \delta_{i_{K^\bullet,m}^\bullet, j_{K^\bullet,m}^\bullet} \end{bmatrix}^\top, \tag{9}$$

for $\bullet = -, +$ and $\delta_{i,j}$ is given by $\delta_{i,j} \coloneqq \delta_{\mathcal{Z}}(z_i, z_j)$ for $i, j \in \mathcal{V}$. Here, we expect that l is increasing with respect to each element of $\boldsymbol{\delta}_m^+$, the dissimilarity between a positive pair, and decreasing with respect to each element of $\boldsymbol{\delta}_m^-$, the dissimilarity between a negative pair.

Assumption 2. The loss function l is Lipschitz continuous for each variable.

We denote the Lipschitz constant of l with respect to $\delta_{i_k^+, j_k^+}$ and $\delta_{i_k^-, j_k^-}$ by L_k^+ and L_k^-, respectively. Note that the assumption regarding the loss function's Lipschitz continuity is common to derive a generalization bound using statistical learning theory (e.g., [33]). The following examples show that the above framework is general enough to include the existing applications' settings. In the following, $h : \mathbb{R} \to \mathbb{R}$ is an increasing function that converts the dissimilarity.

Example 2. (Loss functions)

(a) (Sigmoid-base loss) Define l by

$$l\big(\boldsymbol{\delta}_m^+, \boldsymbol{\delta}_m^-\big) \coloneqq \sum_{k^+=1}^{K^+} \ln \sigma\left(-\left[h\left(\delta_{i_{k^+,m}^+, j_{k^+,m}^+}\right) - h(\theta)\right]\right) + \sum_{k^-=1}^{K^-} \ln \sigma\left(h\left(\delta_{i_{k^-,m}^-, j_{k^-,m}^-}\right) - h(\theta)\right), \tag{10}$$

where σ is a sigmoid shape function. For example, we can use the standard sigmoid function defined by $\sigma_{\mathrm{std}}(x) = \frac{1}{1+\exp(-x)}$, the hinge loss function $\sigma_{\mathrm{hinge}}(x) = \max\{0, x+1\}$ or the ramp loss function $\sigma_{\mathrm{ramp}}(x) = \min\{1, \sigma_{\mathrm{hinge}}(x)\}$. h is L_h-Lipschitz. Then $L_{k^+}^+ = L_{k^-}^- = \frac{L_h}{4}$ if $\sigma = \sigma_{\mathrm{std}}$ and $L_{k^+}^+ = L_{k^-}^- = L_h$ if $\sigma = \sigma_{\mathrm{hinge}}$ or $\sigma = \sigma_{\mathrm{ramp}}$, for $k^+ = 1, 2, \ldots, K^+$ and $k^- = 1, 2, \ldots, K^-$. The above loss function (10) with the standard sigmoid function σ corresponds to the negative-sampling-based loss function in [1] ($h(x) = x$ and $\theta = 0$) and the loss function in [20] for network embedding ($h(x) = \frac{x}{t}$ and $\theta = r$, where r, t are fixed constants defined in the paper).

(b) (Softmax-like loss) Let $K^+ = 1$ and define l by

$$l\big(\boldsymbol{\delta}_m^+, \boldsymbol{\delta}_m^-\big) \coloneqq -h\left(\delta_{i_{1,m}^+, j_{1,m}^+}\right) + \ln \sum_{k^-=1}^{K^-} \exp\left(-h\left(\delta_{i_{k^-,m}^-, j_{k^-,m}^-}\right)\right). \tag{11}$$

Suppose that h is L_h-Lipschitz. Then $L_{k^+}^+ = L_{k^-}^- = L_h$, for $k^+ = 1, 2, \ldots, K^+$ and $k^- = 1, 2, \ldots, K^-$. The loss function (11) corresponds to the loss function for embedding taxonomy in [20] with $h(x) = \sqrt{x}$, although this h gives a non-Lipschitz-continuous loss function. In this paper, since a non-smooth loss function often loses a generalization guarantee, we only consider the case where h and l are Lipschitz continuous, as in [33].

2.4 Generalization error

The core discussion of this paper is the generalization error, which is the difference between the empirical risk and expected risk. We define the empirical risk function $\hat{\mathcal{R}}_{(s_m)_{m=1}^M}^{\mathcal{Z}} : (\mathcal{Z})^V \to \mathbb{R}_{\geq 0}$ on training data $s = (s)_{m=1}^M$ and the expected risk function $\mathcal{R}^{\mathcal{Z}} : (\mathcal{Z})^V \to \mathbb{R}_{\geq 0}$ as follows:

$$\hat{\mathcal{R}}_s^{\mathcal{Z}}\left((\boldsymbol{z}_v)_{v=1}^V\right) \coloneqq \frac{1}{M} \sum_{m=1}^M l\big(\boldsymbol{\delta}_m^+, \boldsymbol{\delta}_m^-\big), \quad \mathcal{R}^{\mathcal{Z}}\left((\boldsymbol{z}_v)_{v=1}^V\right) \coloneqq \mathbb{E}_{s_m}\, l\big(\boldsymbol{\delta}_m^+, \boldsymbol{\delta}_m^-\big). \tag{12}$$

Let $\mathcal{B} \subset \mathcal{Z}^{|\mathcal{V}|}$ be the space which we search for optimal representations $z_1, z_2, \ldots, z_{|\mathcal{V}|}$. As we discuss in the next section, $\mathcal{B}$ may be a bounded set rather than the whole space $\mathcal{Z}^{|\mathcal{V}|}$. We define the empirical risk minimizer $(\hat{\boldsymbol{z}}_v)_{v=1}^{|\mathcal{V}|}$ and expected risk minimizer $(\boldsymbol{z}_v^*)_{v=1}^{|\mathcal{V}|}$ by

$$(\hat{\boldsymbol{z}}_v)_{v=1}^{|\mathcal{V}|} \coloneqq \operatorname*{argmin}_{(\boldsymbol{z}_v)_{v=1}^{|\mathcal{V}|} \in \mathcal{B}} \hat{\mathcal{R}}_s^{\mathcal{Z}}\Big((\boldsymbol{z}_v)_{v=1}^{|\mathcal{V}|}\Big), \quad (\boldsymbol{z}_v^*)_{v=1}^{|\mathcal{V}|} \coloneqq \operatorname*{argmin}_{(\boldsymbol{z}_v)_{v=1}^{|\mathcal{V}|} \in \mathcal{B}} \mathcal{R}^{\mathcal{Z}}\Big((\boldsymbol{z}_v)_{v=1}^{|\mathcal{V}|}\Big). \tag{13}$$

Our interest is the *excess risk* given by $\mathcal{R}^{\mathcal{Z}}\Big((\hat{\boldsymbol{z}}_v)_{v=1}^{|\mathcal{V}|}\Big) - \mathcal{R}^{\mathcal{Z}}\Big((\boldsymbol{z}_v^*)_{v=1}^{|\mathcal{V}|}\Big)$, which indicates the generalization error of embedding. We derive the upper bound of the excess risk in the next section.

Remark 3. In the remainder of this paper, we compare the generalization error among multiple dissimilarity spaces. Since the loss itself depends on the embedding space's dissimilarity function $\mathcal{Z}$, the comparison is not always completely fair. Nevertheless, the comparison can be meaningful. For example, consider the 0-1 loss defined by (10) with $\sigma(x) = \sigma_{0-1}(x) \coloneqq \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases}$. Regardless of the choice of embedding space $\mathcal{Z}$, the risk function on the 0-1 loss indicates the error rate of classifying a pair (i, j) into "similar ($\delta^*(i, j) \leq \theta^*$)" or "dissimilar ($\delta^*(i, j) > \theta^*$)." Thus, it is fair to compare this 0-1-loss-based risk between different embedding spaces. Although the 0-1 loss itself does not satisfy Assumption 2, since the hinge loss and ramp loss dominates the 0-1 loss in that $\sigma_{0-1}(x) \leq \sigma_{\mathrm{ramp}}(x) \leq \sigma_{\mathrm{hinge}}(x)$, deriving and comparing bounds for these loss indirectly enables comparison between the 0-1-loss-based risks of different embedding spaces. In this sense, comparing the risks of different embedding spaces based on a Lipschitz continuous loss is of interest.

3 Generalization Bounds for Graph Embedding

3.1 Assumptions on Embedding Space's Radius

As also discussed in [21] for ordinal embedding, to derive a finite generalization bound, in general, it is necessary to restrict parameters (in embedding cases, representations) to a bounded domain (e.g., linear prediction models [37, 38], neural networks [37, 39]). In this section, we discuss our restriction on embedding space. For the derived generalization bound to be practical, the restriction should be simple and geometrically intuitive. Following [21], we put the following simple restrictions on embedding space's radius. Specifically, we discuss the case where we search for representations in $\mathcal{B} = \mathcal{B}_R$ defined by

$$\mathcal{B}_R \coloneqq \Big\{(z_v)_{v=1}^{|\mathcal{V}|} \Big| \forall v \in \mathcal{V} : \Delta_{\mathcal{Z}}(z_0, z_v) \leq R\Big\}, \tag{14}$$

where z_0 is $[0 \quad 0 \quad \ldots 0] \in \mathbb{R}^D$ for $\mathcal{Z} = \mathbb{R}^D, \mathbb{I}^D$ and $[1 \quad 0 \quad \ldots 0] \in \mathbb{R}^{D+1}$ for $\mathcal{Z} = \mathbb{L}^D, \mathbb{S}^D$, and $\Delta_{\mathbb{I}^D}$ is defined by $\Delta_{\mathbb{I}^D}(\boldsymbol{z}, \boldsymbol{z}') = \|\boldsymbol{z}' - \boldsymbol{z}\|_2$. In the next section, we provide the generalization error bound for empirical risk minimizer in $\mathcal{B}^R$.

3.2 Main Result: Finite Sample Upper Bounds for the Generalization Error

In this section, we first give our main theorem, which gives an upper bound for the generalization error, followed by remarks about intuitive interpretation of the bound and simplified versions.

Theorem 1. *Let $\mathcal{Z} = \mathbb{R}^D, \mathbb{L}^D, \mathbb{S}^D$ or $\mathbb{I}^D$, and $(\hat{\boldsymbol{z}}_v)_{v=1}^{|\mathcal{V}|}$ and $(\boldsymbol{z}_v^*)_{v=1}^{|\mathcal{V}|}$ be empirical and expected risk minimizers defined by* (13)*. Under Assumptions 1 and 2, the following inequality holds with probability $1 - \mathfrak{d}$:*

$$\mathcal{R}^{\mathcal{Z}}\Big((\hat{\boldsymbol{z}}_v)_{v=1}^{|\mathcal{V}|}\Big) - \mathcal{R}^{\mathcal{Z}}\Big((\boldsymbol{z}_v^*)_{v=1}^{|\mathcal{V}|}\Big) \leq \frac{2\omega_{\mathcal{Z}}(R)}{M} \sum_{\bullet=+,-} \sum_{k=1}^{K^\bullet} L_k^\bullet \Big(\sqrt{2M\nu_{k\mathcal{Z}}^\bullet \ln|\mathcal{V}|} + \frac{\kappa_{\mathcal{Z}}}{3}\ln|\mathcal{V}|\Big) + I_l(\mathcal{B})\sqrt{\frac{\ln\frac{2}{\mathfrak{d}}}{M}}, \tag{15}$$

where $\omega_{\mathcal{Z}}$ is defined by $\omega_{\mathbb{R}^D}(R) = \omega_{\mathbb{I}^D}(R) \coloneqq (2R)^2$, $\omega_{\mathbb{L}^D}(R) \coloneqq \cosh^2 R + \sinh^2 R$ and $\omega_{\mathbb{S}^D}(R) \coloneqq \frac{2\arccos(1-\cos 2R)}{\sqrt{-\cos 2R(\cos 2R - 2)}}$, $\kappa_{\mathbb{L}^D} = \kappa_{\mathbb{S}^D} = \kappa_{\mathbb{I}^D} = \frac{1}{2}$ and $\kappa_{\mathbb{R}^D} = 2$, and $\nu_k^\bullet \coloneqq$

$\left\|\mathbb{E}_{(i^{\bullet}_{k,m},j^{\bullet}_{k,m})}\boldsymbol{E}^{\mathcal{Z}}_{i^{\bullet}_{k,m},j^{\bullet}_{k,m}}{}^{\top}\boldsymbol{E}^{\mathcal{Z}}_{i^{\bullet}_{k,m},j^{\bullet}_{k,m}}\right\|_{\mathrm{op},2}$ *for* $k=1,2,\ldots,K^{\bullet}$ *and* $\bullet=+,-$. *Here,* $\boldsymbol{E}^{\mathcal{Z}}_{i,j}$ *is defined by*

$$\left[\boldsymbol{E}^{\mathcal{Z}}_{i,j}\right]_{i',j'}\coloneqq\begin{cases}a^{\mathcal{Z}}_{diag} & \text{if } (i,j)=(i',j'),\\ a^{\mathcal{Z}}_{off} & \text{if } i'=j'=i \text{ or } i'=j'=j \text{ and } (i,j)\neq(i',j''),\\ 0 & \text{otherwise},\end{cases} \tag{16}$$

where $a^{\mathbb{R}^D}_{diag}=1$, $a^{\mathbb{R}^D}_{off}=-1$, $a^{\mathbb{L}^D}_{diag}=-\frac{1}{2}$, $a^{\mathbb{S}^D}_{diag}=a^{\mathbb{I}^D}_{diag}=\frac{1}{2}$, *and* $a^{\mathbb{L}^D}_{off}=a^{\mathbb{S}^D}_{off}=a^{\mathbb{I}^D}_{off}=0$. $I_l(\mathcal{B})$ *is the range of* l *defined by*

$$I_l(\mathcal{B})\coloneqq\max\left\{l\big(\boldsymbol{\delta}^+,\boldsymbol{\delta}^-\big)\;\middle|\;s_m\in\mathcal{S},(\boldsymbol{z}_v)_{v=1}^V\in\mathcal{B}\right\}-\min\left\{l\big(\boldsymbol{\delta}^+,\boldsymbol{\delta}^-\big)\;\middle|\;s_m\in\mathcal{S},(\boldsymbol{z}_v)_{v=1}^V\in\mathcal{B}\right\}, \tag{17}$$

where $\mathcal{S}\coloneqq(\mathcal{V}\times\mathcal{V})^{K^+}\times(\mathcal{V}\times\mathcal{V})^{K^-}$.

Remark 4. $\omega_{\mathcal{Z}}(R)$ corresponds to the embedding space's volume. Similar to ordinal embedding cases in [21], Theorem 1 argues that the larger the embedding space is, the larger the generalization error is. Since the volume of a ball is polynomial and exponential with respect to its radius in an inner product space and hyperbolic space, respectively, the generalization error also behaves correspondingly. $\nu^{\bullet}_k$ indicates the imbalance in the training data's distribution. Theorem 1 suggests that an imbalanced distribution causes large generalization error. This is intuitive because training data's imbalance makes it difficult to understand the big picture of the data. We will explore this in Example 3 and Remark 8.

Remark 5. If $\mathcal{Z}=\mathbb{R}^D$ or $\mathbb{I}^D$, multiplying a constant α to R and α^2 to Lipschitz constant $L^{\bullet}_k$ imply the same relaxation of the condition, since these two are equivalent by scaling representations. Hence, the bound in Theorem 1 for $\mathcal{Z}=\mathbb{R}^D$ or $\mathbb{I}^D$ includes a term $R^2L^{\bullet}_k$, which indicates the essential size of the embedding space. This discussion is not true of $\mathcal{Z}=\mathbb{L}^D$ and $\mathbb{S}^D$, because balls with different sizes are not similar to each other in these spaces.

Remark 6. Owing to Assumption 2, $I_l\leq(2R)^2\sum_{\bullet=+,-}\sum_{k=1}^{K^{\bullet}}L^{\bullet}_k$ always holds.

Remark 7. For any distribution on s, by Jensen's inequality, we have that $\nu^{\bullet}_k\leq\mathbb{E}_{(i^{\bullet}_{k,m},j^{\bullet}_{k,m})}\left\|\boldsymbol{E}^{\mathcal{Z}}_{i^{\bullet}_{k,m},j^{\bullet}_{k,m}}{}^{\top}\boldsymbol{E}^{\mathcal{Z}}_{i^{\bullet}_{k,m},j^{\bullet}_{k,m}}\right\|_{\mathrm{op},2}=c_{\mathcal{Z}}$ holds, where $c_{\mathcal{Z}}=4$ for $\mathcal{Z}=\mathbb{R}^D$ and $c_{\mathcal{Z}}=\frac{1}{4}$ for $\mathcal{Z}=\mathbb{L}^D,\mathbb{S}^D$, and $\mathbb{I}^D$.

The proof is based on evaluating the Rademacher complexity [31, 32, 33], but technically nontrivial due to the dependency among the data. We have overcome this difficulty by our novel decomposition of the Rademacher complexity. See the supplementary materials for the complete proof of Theorem 1. In the following, we discuss examples of specific negative sampling strategies.

Example 3 (Calculation of $\nu^{\bullet}_k$)**.** Consider the setting in Example 1. Then the following holds.

Proposition 1.

(a) *Suppose that the data distribution is given by* (6) *and* (7)*. Then we have that*

$$\nu^{+}_{k,\mathcal{Z}}=\frac{1}{2}\cdot\frac{(1-r^+)\max_{i\in\mathcal{V}}\deg(i)+r^+(|\mathcal{V}|-1)}{[(1-r^+)|\mathcal{E}|+r^+|\mathcal{V}|(|\mathcal{V}|-1)]}\leq\frac{1}{2}\cdot\frac{\max_{i\in\mathcal{V}}\deg(i)}{|\mathcal{E}|}, \tag{18}$$

$$\nu^{-}_{k,\mathcal{Z}}=\frac{1}{2}\cdot\frac{(1-r^-)[(|\mathcal{V}|-1)-\min_{i\in\mathcal{V}}\deg(i)]+r^-(|\mathcal{V}|-1)}{[(1-r^-)[|\mathcal{V}|(|\mathcal{V}|-1)-|\mathcal{E}|]+r^-|\mathcal{V}|(|\mathcal{V}|-1)]}\leq\frac{1}{2}\cdot\frac{(|\mathcal{V}|-1)-\min_{i\in\mathcal{V}}\deg(i)}{|\mathcal{V}|(|\mathcal{V}|-1)-|\mathcal{E}|}, \tag{19}$$

for $\mathcal{Z}=\mathbb{L}^D,\mathbb{I}^D,\mathbb{S}^D$*, and for* $\mathcal{Z}=\mathbb{R}^D$ *we have that*

$$\nu^{+}_{k,\mathbb{R}^D}\leq\frac{2[(1-r^+)\{2\max_{i\in\mathcal{V}}\deg(i)\}+r^+|\mathcal{V}|]}{(1-r^+)|\mathcal{E}|+r^+|\mathcal{V}|(|\mathcal{V}|-1)}\leq4\cdot\frac{\max_{i\in\mathcal{V}}\deg(i)}{|\mathcal{E}|}, \tag{20}$$

$$\nu^{-}_{k,\mathbb{R}^D}\leq4\frac{1}{|\mathcal{V}|}\frac{(|\mathcal{V}|-1)-(1-r^-)\min_{i\in\mathcal{V}}\deg(i)}{(|\mathcal{V}|-1)-(1-r^-)\frac{|\mathcal{E}|}{|\mathcal{V}|}}\leq4\cdot\frac{1}{|\mathcal{V}|}\frac{(|\mathcal{V}|-1)-\min_{i\in\mathcal{V}}\deg(i)}{(|\mathcal{V}|-1)-\frac{|\mathcal{E}|}{|\mathcal{V}|}}. \tag{21}$$

(b) *If the conditional distribution of negative pairs is given by* (8)*, then we have that*

$$\nu^{-}_{k,\mathcal{Z}}=\frac{1}{2}\cdot\frac{(1-r^-)\max_{i\in\mathcal{V}}\deg(i)+r^-(|\mathcal{V}|-1)}{[(1-r^-)|\mathcal{E}|+r^-|\mathcal{V}|(|\mathcal{V}|-1)]}\leq\frac{1}{2}\cdot\frac{\max_{i\in\mathcal{V}}\deg(i)}{|\mathcal{E}|}, \tag{22}$$

for $\mathcal{Z} = \mathbb{L}^D, \mathbb{I}^D, \mathbb{S}^D$, *and*

$$\nu^-_{k,\mathbb{R}^D} = 2\left[\frac{(1-r^-)\max_{i\in\mathcal{V}}\deg(i) + r^-(|\mathcal{V}|-1)}{(1-r^-)|\mathcal{E}| + r^-|\mathcal{V}|(|\mathcal{V}|-1)} + 1\right] \leq 2\left[\frac{\max_{i\in\mathcal{V}}\deg(i)}{|\mathcal{E}|} + 1\right]. \tag{23}$$

See the supplementary materials for the proof of Proposition 1.

Remark 8. We can interpret the evaluations of ν^+_1 and ν^-_k in Example 3 as follows. In the following, we take (18) and (22) as examples, but similar discussion holds for all settings discussed in Example 3. First, we consider $r = 0$, which implies no noise setting. In this case, $\nu^+_1 = \nu^-_k = \frac{1}{2}\cdot\frac{\max_{i\in\mathcal{V}}\deg(i)}{|\mathcal{E}|}$ is valid. Here, the right hand side indicates the edge distribution's imbalance. For example, the right hand side takes the minimum $\frac{1}{2|\mathcal{V}|}$ if the graph is regular, where the distribution of edges are completely balanced, and takes the maximum $\frac{1}{2}$ if the graph is a star, where all edges are connected to one vertex, which is most imbalanced case. Hence, the equation regarding ν^+_1 and ν^-_k implies that the more balanced the edge distribution is, the smaller generalization error it has. This relation is consistent to what we have emphasized in Remark 4. Second, we consider the relation between r and the generalization error. The bound is decreasing with respect to r, the noise intensity. This is because training data is balanced between truly positive and negative pairs if the noise is intensive. Note that it does not imply that larger noise is better, because large noise worsens the optimal expected loss, even though it has small generalization error, which may lead to a large loss of learned representations.

4 Comparison between LGE and HGE

Theorem 1 implies that HGE has a larger generalization error than LGE; in other words, HGE has a higher variance than LGE. Conversely, hyperbolic space's ability to obtain low-distortion representations for a complete noiseless tree has been shown [29, 40, 21]; in other words, HGE has a lower bias than LGE. To evaluate the performance of learning models, discussing the bias-variance trade-off is essential. In this section, by combining the above bias and variance discussions, we derive an explicit condition on which HGE outperforms LGE in obtaining representations for a tree.

For fair discussion, suppose that the loss function l is given by (10) with the ramp loss function $\sigma = \sigma_{\mathrm{ramp}}$, which dominates the 0-1 loss function as discussed in Remark 3, and the data distribution is given by the setting in Example 1 (a) with $K^+ = K^- = 1$. Consider the following conditions stronger than (5):

$$(i,j) \in \mathcal{E} \Rightarrow \delta_{\mathcal{Z}}(z_i, z_j) \leq \theta_{\mathcal{Z}} - 1, \quad (i,j) \notin \mathcal{E} \Rightarrow \delta_{\mathcal{Z}}(z_i, z_j) \geq \theta_{\mathcal{Z}} + 1, \tag{24}$$

and define $\mathcal{R}^*$ as the expected risk given by representations that satisfy the above conditions. Under the conditions, the loss is zero if truly positive pairs appear as positive pair $\left(\left(i^+_{k,m}, j^+_{k,m}\right) \in \mathcal{E}\right)$ or truly negative pairs appear as negative pair $\left(\left(i^-_{k,m}, j^-_{k,m}\right) \in \mathcal{E}\right)$. Then $\mathcal{R}^*$ is the achievable minimum expected risk.

Let $V^{\mathbb{R}^2}_{\min}$ and $V^{\mathbb{L}^2}_{\min}$ denote the minimum V attainable by LGE using $\mathbb{R}^2$ and HGE using $\mathbb{L}^2$, respectively. Assume that the true dissimilarity δ^* is the graph distance of a tree. Regarding $V^{\mathbb{R}^2}_{\min}$ and $V^{\mathbb{L}^2}_{\min}$, the following lemmata hold [29, 40, 21] (See the supplementary materials for the proofs).

Lemma 1. *Suppose that* $(\mathcal{V}, \mathcal{E})$ *is a tree and* $\delta^* : \mathcal{V}\times\mathcal{V} \to \mathbb{R}_{\geq 0}$ *is given by its graph distance. Then, there exist* $R \in \mathbb{R}_{\geq 0}$, *representations* $(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_V) \in \mathcal{B}_R$ *in* $\mathbb{L}^2$, *and threshold* $\theta_{\mathcal{Z}} \in \mathbb{R}$ *that satisfy* (24) *for all* $i, j \in \mathcal{V}$. *In particular,* $\mathcal{R}^{\mathcal{Z}}\left((\boldsymbol{z}^*_v)^{|\mathcal{V}|}_{v=1}\right) = \mathcal{R}^*$.

Lemma 2. *Let* $(\mathcal{V}, \mathcal{E})$ *be a graph and define* W *and* ρ *by* $W \coloneqq |\mathcal{V}|(|\mathcal{V}|-1)$ *and* $\mathcal{E} \coloneqq \frac{|\mathcal{E}|}{W}$. *Define* $\mu \coloneqq \min\{\mu^+, \mu^-\}$, *where* $\mu^+ \coloneqq \frac{1}{W}\frac{1}{(1-r^+)\rho+r^+} - \frac{1}{W}\frac{r^-}{(1-r^-)(1-\rho)+r^-}$ *and* $\mu^- \coloneqq \frac{1}{W}\frac{1}{(1-r^-)(1-\rho)+r^-} - \frac{1}{W}\frac{r^+}{(1-r^+)\rho+r^+}$. *In LGE, the expected risk of the expected risk minimizers satisfies* $\mathcal{R}^{\mathcal{Z}}\left((\boldsymbol{z}^*_v)^{|\mathcal{V}|}_{v=1}\right) \geq \mathcal{R}^* + V^{\mathbb{R}^2}_{\min}\mu$. *Here,* $V^{\mathbb{R}^2}_{\min}$ *is not smaller than the number of disjoint 6-star subgraphs in the graph for the 2-dimensional LGE.*

By Theorem 1, we can conclude as follows:

Proposition 2. *Suppose that $(\mathcal{V}, \mathcal{E})$ is a tree and $\delta^* : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geq 0}$ is given by its graph distance, and take R given in Lemma 1. Let $\nu_{1,\mathbb{L}^D} \coloneqq \left(\sqrt{\nu^+_{1,\mathbb{L}^D}} + \sqrt{\nu^-_{1,\mathbb{L}^D}}\right)^2$. If*

$$M > \left(\frac{3\omega_{\mathbb{L}^D}(R)}{4|\mathcal{V}|\mu V^{\mathbb{R}^2}_{\min}} \left(\sqrt{8\nu_{1,\mathbb{L}^D} \ln |\mathcal{V}|} + \sqrt{\ln \frac{2}{\mathfrak{d}}} \right) + \frac{1}{2\left(\sqrt{8\nu_{1,\mathbb{L}^D} \ln |\mathcal{V}|} + \sqrt{\ln \frac{2}{\mathfrak{d}}} \right)} \right)^2, \tag{25}$$

then HGE's $\mathcal{R}^{\mathcal{Z}}\left((\hat{\boldsymbol{z}}_v)_{v=1}^{|\mathcal{V}|}\right)$ is smaller than LGE's.

See the supplementary materials for the proof of Proposition 2.

Remark 9. Proposition 2 implies that if the true dissimilarity is given by the graph distance of a tree, then HGE is better than LGE even if the data is not complete and noisy, if M is larger than the right hand side of (25).

Example 4. We consider the complete balanced λ-ary tree with height h. Suppose $\lambda = 5$ and $h = 4$, and the positive and negative pair distributions are given by Example 1 (a) with $r^+ = r^- = 10^{-4}$. Then, HGE's $\mathcal{R}^{\mathcal{Z}}\left((\hat{\boldsymbol{z}}_v)_{v=1}^{|\mathcal{V}|}\right)$ with $R = 39.50...$ is smaller than LGE's if $M > 7.735 \times 10^{69}$ with probability 1 - $\mathfrak{d}$, where $\mathfrak{d} = 10^{-1}$.

See the supplementary materials for the proof of Example 4. Example 4 is the first specific calculation result that theoretically guarantees the superiority of HGE to LGE. Nevertheless, tightening the right hand side or deriving a necessary condition could be future work.

5 Conclusion

We have shown that LGE and HGE cause a polynomial and exponential error with respect to the embedding space's radius, respectively, and that imbalanced data distribution can worsen the error. Our bias-variance trade-off discussion implies that even though HGE has larger generalization error than LGE, HGE with sufficiently large number of data can represent hierarchical data more effectively. This discussion provides a guide for embedding space selection in real applications.

One limitation of our result is that it does not clarify the error's dependency on the dimension D. Our bound does not depend on D, which is not consistent to our intuition that using low-dimensional space should give low generalization error. This problem is essentially the same as that pointed out in [21], which partially used similar techniques to ours. Deriving a tighter bound in terms of the dimension for general ordinal embedding could be future work.

Acknowledgments and Disclosure of Funding

This work was partially supported by JST KAKENHI 191400000190, 19K20337, and JST-AIP JPMJCR19U4, Daiwa Foundation Award (Ref: 13849/14682), and JST-PRESTO.

References

[1] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 27th Conference on Neural Information Processing Systems*, pp. 3111–3119, 2013.

[2] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[4] A. Tifrea, G. Bécigneul, and O.-E. Ganea, "Poincaré GloVe: Hyperbolic word embeddings," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[5] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *Journal of the american Statistical association*, vol. 97, no. 460, pp. 1090–1098, 2002.

[6] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proceedings of The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 701–710, 2014.

[7] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," in *Proceedings of Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pp. 1067–1077, 2015.

[8] J. Tang, M. Qu, and Q. Mei, "PTE: predictive text embedding through large-scale heterogeneous text networks," in *Proceedings of Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pp. 1165–1174, 2015.

[9] A. Grover and J. 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.

[10] J. M. Dale, L. Popescu, and P. D. Karp, "Machine learning methods for metabolic pathway prediction," *BMC bioinformatics*, vol. 11, no. 1, pp. 1–14, 2010.

[11] A. R. MA Basher and S. J. Hallam, "Leveraging heterogeneous network embedding for metabolic pathway prediction," *Bioinformatics*, vol. 37, no. 6, pp. 822–829, 2021.

[12] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th International Conference on Machine Learning*, pp. 809–816, 2011.

[13] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proceedings of Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 2787–2795, 2013.

[14] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pp. 74–84, 2013.

[15] M. Nickel, L. Rosasco, and T. A. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pp. 1955–1961, 2016.

[16] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, vol. 48, pp. 2071–2080, 2016.

[17] T. Ebisu and R. Ichise, "TorusE: Knowledge graph embedding on a Lie group," in *Proceedings of Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 1819–1826, 2018.

[18] J. Lamping and R. Rao, "Laying out and visualizing large trees using a hyperbolic space," in *Proceedings of the 7th ACM Symposium on User Interface Software and Technology*, pp. 13–14, 1994.

[19] H. Ritter, "Self-organizing maps on non-euclidean spaces," in *Kohonen maps*, Elsevier, 1999, pp. 97–109.

[20] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, pp. 6338–6347, 2017.

[21] A. Suzuki, A. Nitanda, J. Wang, L. Xu, K. Yamanishi, and M. Cavazza, "Generalization error bound for hyperbolic ordinal embedding," *arXiv preprint arXiv:2105.10475*, 2021.

[22] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic entailment cones for learning hierarchical embeddings," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 1632–1641, 2018.

[23] F. Sala, C. D. Sa, A. Gu, and C. Ré, "Representation tradeoffs for hyperbolic embeddings," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 4457–4466, 2018.

[24] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," in *Proceedings of the 32nd Conference on Neural Information Processing Systems*, pp. 5350–5360, 2018.

[25] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pp. 4869–4880, 2019.

[26] Ç. Gülçehre *et al.*, "Hyperbolic attention networks," in *Proceedings of 7th International Conference on Learning Representations*, 2019.

[27] I. Balazevic, C. Allen, and T. M. Hospedales, "Multi-relational poincaré graph embeddings," in *Proceedings of the 33rd Conference on Neural Information Processing Systems*, pp. 4465–4475, 2019.

[28] M. Gromov, "Hyperbolic groups," in *Essays in group theory*, Springer, 1987, pp. 75–263.

[29] R. Sarkar, "Low distortion delaunay embedding of trees in hyperbolic plane," in *Proceedings of the 19th International Symposium on Graph Drawing*, vol. 7034, pp. 355–366, 2011.

[30] L. Jain, K. G. Jamieson, and R. D. Nowak, "Finite sample prediction and recovery bounds for ordinal embedding," in *Proceedings of the 30th Conference on Neural Information Processing Systems*, vol. 29, pp. 2703–2711, 2016.

[31] V. Koltchinskii, "Rademacher penalties and structural risk minimization," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1902–1914, 2001.

[32] V. Koltchinskii and D. Panchenko, "Rademacher processes and bounding the risk of function learning," in *High dimensional probability II*, Springer, 2000, pp. 443–457.

[33] P. L. Bartlett, S. Boucheron, and G. Lugosi, "Model selection and error estimation," *Machine Learning*, vol. 48, no. 1-3, pp. 85–113, 2002.

[34] P. Tabaghi and I. Dokmanic, "Hyperbolic distance matrices," in *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1728–1738, 2020.

[35] Y. Gao, C. Zhang, J. Peng, and A. Parameswaran, "The importance of norm regularization in linear graph embedding: Theoretical analysis and empirical demonstration," *arXiv preprint arXiv:1802.03560*, 2018.

[36] M. Ledoux and M. Talagrand, "Probability in banach spaces: Isoperimetry and processes," 1991.

[37] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2002.

[38] S. M. Kakade, K. Sridharan, and A. Tewari, "On the complexity of linear prediction: Risk bounds, margin bounds, and regularization," in *Proceedings of the 22nd Conference on Neural Information Processing Systems*, pp. 793–800, 2008.

[39] J. Schmidt-Hieber *et al.*, "Nonparametric regression using deep neural networks with relu activation function," *Annals of Statistics*, vol. 48, no. 4, pp. 1875–1897, 2020.

[40] A. Suzuki, J. Wang, F. Tian, A. Nitanda, and K. Yamanishi, "Hyperbolic ordinal embedding," in *Proceedings of the 11th Asian Conference on Machine Learning*, vol. 101, pp. 1065–1080, 2019.

[41] J. A. Tropp, "An introduction to matrix concentration inequalities," *Foundations and Trends in Machine Learning*, vol. 8, no. 1-2, pp. 1–230, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes]
(b) Did you include complete proofs of all theoretical results? [Yes] See the supplementary materials

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]