
Nested Subspace Arrangement for Representation of Relational Data

Nozomi Hata^{*1} Shizuo Kaji^{*2} Akihiro Yoshida¹ Katsuki Fujisawa²

Abstract

Studies on acquiring appropriate continuous representations of discrete objects, such as graphs and knowledge base data, have been conducted by many researchers in the field of machine learning. In this study, we introduce Nested SubSpace (NSS) arrangement, a comprehensive framework for representation learning. We show that existing embedding techniques can be regarded as special cases of the NSS arrangement. Based on the concept of the NSS arrangement, we implement a Disk-ANChor ARrangement (DANCAR), a representation learning method specialized to reproducing general graphs. Numerical experiments have shown that DANCAR has successfully embedded WordNet in \mathbb{R}^{20} with an F1 score of 0.993 in the reconstruction task. DANCAR is also suitable for visualization in understanding the characteristics of graphs.

1. Introduction

Studies on acquiring the appropriate continuous representation of discrete objects have been closely associated with machine learning. These studies aim to obtain the low-dimensional vector representations of the objects by preserving their characteristics. Recent algorithms of representation learning have broad applications, such as preprocessing in machine learning or visualization. As discrete objects, graphs, knowledge bases, and social networks are the primary research targets in these fields.

By representation, we mean the following: let \mathcal{V} be a set of objects with a certain discrete structure. An *embedding* or a *representation* of \mathcal{V} is a mapping $\Psi : \mathcal{V} \rightarrow \mathcal{X}$, where \mathcal{X} is a space parametrized by real vectors. Through Ψ , the set \mathcal{V} can be equipped with various structures of \mathcal{X} such as addi-

tion, multiplication, differentiation, metric, and topology so that various operations, analysis, optimization techniques become available to deal with the elements of \mathcal{V} . Representation has served as a fundamental building block for various algorithms such as classification, clustering, information retrieval, link prediction, and visualization.

Below, we review some existing works according to the type of \mathcal{V} and \mathcal{X} , categorized in Table 1. The most classical and fundamental case is when \mathcal{V} is the set of (non-directed, simple) graphs, and \mathcal{X} is point clouds in a Euclidean space \mathbb{R}^k , where vertices of a graph are mapped as points in \mathbb{R}^k . Existing works in this direction include matrix factorization models (Chung, 1997; Cao et al., 2015; Ou et al., 2016; Singh & Gordon, 2008; Cox & Cox, 2000; Balasubramanian & Schwartz, 2002), random walk based models (Grover & Leskovec, 2016; Perozzi et al., 2014; Dong et al., 2017; Pan et al., 2016; Yanardag & Vishwanathan, 2015), and others (Kipf & Welling, 2016; Wang et al., 2016; Chami et al., 2019; Khasanova & Frossard, 2017; Li et al., 2016; Duvenaud et al., 2015; Monti et al., 2017; Wang et al., 2018).

A common problem with these methods is the determination of the embedding dimension k . If it is too small, the embedding does not preserve structures of \mathcal{V} . To achieve a high fidelity embedding, using point clouds in the Poincaré disk as the target space \mathcal{X} was proposed (Nickel & Kiela, 2017). In particular, they succeeded in obtaining a low-dimensional representation of tree-like graphs.

In search of representations of more general graphs, the Disk Embedding (Suzuki et al., 2019) was proposed to deal with directed acyclic graphs (DAGs). Their idea is to use as \mathcal{X} the set of disks (balls) in a metric space such as a Euclidean space, a spherical space, or a hyperbolic space. This approach generalizes existing works for embedding DAGs such as the Order Embedding (Vendrov et al., 2016) and the Hyperbolic Entailment Cones (Ganea et al., 2018). The key idea of the Disk Embedding is to represent an edge of a graph by an inclusion relation between two disks, which leads to a successful embedding for DAGs.

Moreover, some embedding algorithms for hypergraphs have been recently proposed (Feng et al., 2019; Tu et al., 2018; Yang et al., 2019).

Other than graphs, knowledge bases have also been inten-

^{*}Equal contribution ¹Graduate School of Mathematics, Kyushu University, Fukuoka, Japan ²Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan. Correspondence to: Nozomi Hata <n.hata@kyudai.jp>.

Table 1. The categorization of representation learning with respect to its input and representation space.

Method	\mathcal{V}	\mathcal{X}	Structure
Most existing method TransE	undirected graph multi relational data	points in \mathbb{R}^k points in \mathbb{R}^k	metric or (dis)similarity metric, addition
Poincaré Embedding Disk Embedding	hierarchical data directed acyclic graph	points in the Poincaré disk disks in a metric space	metric metric, inclusion
DANCAR (Proposed)	directed graph	anchored disks in a metric space	metric, inclusion

sively studied by many researchers. A knowledge base consists of various relationships among entities represented as triples (head entity, relation, tail entity), e.g., (Vienna, IsCapitalOf, Austria). They are essential resources for many applications such as question answering, content tagging, fact-checking, and knowledge inference. TransE (Bordes et al., 2013) is the first translation-based method, which embeds entities and relations in a Euclidean space with the latter represented by the differences in the former. Many extended versions of TransE have been proposed, such as TransH (Wang et al., 2014), STransE (Nguyen et al., 2016), Riemannian TransE (Suzuki et al., 2018) and TorusE (Ebisu & Ichise, 2018).

In this study, we propose the Nested SubSpace arrangement (NSS arrangement), a comprehensive framework for representation learning (§2). In the NSS arrangement, a node is represented by a nested subspace of a metric space. Our NSS arrangement generalizes existing embedding techniques. As a special case of the NSS arrangement, we also propose the Disk-ANChor ARrangement (DANCAR), which is an embedding method for directed graphs possibly with cycles (§3). The DANCAR maps a node to a pair of a disk and a point contained in the disk. A directed edge is considered to be present when a disk contains the point of another pair. This containment relation is not symmetric nor transitive, which vests the DANCAR a great representational capacity.

We demonstrated the DANCAR with two numerical experiments. First, we visualized a part of the Twitter network, where both the cluster structure and the hierarchical structure of the graph were successfully captured (§4). Second, we conducted the reconstruction task using the WordNet to see the representation capacity of the DANCAR (§5). The experiment showed that the DANCAR successfully represented the graph, where an F1 score of 0.993 for the edge reconstruction task was achieved by the embedding in a 20-dimensional Euclidean space.

Our contributions in this study are summarized as follows:

- We propose the NSS arrangement, a general framework to represent relational data in a continuous space.
- As a special case of the NSS arrangement, we propose the DANCAR to represent directed graphs.
- We show that the DANCAR can be used to visualize

a large-scale network to reveal cluster structure and hierarchical structure.

- We show that the DANCAR can be used to represent a directed graph accurately in terms of the edge reconstruction task.

2. Nested SubSpace Arrangement

In this section, we introduce the NSS arrangement to represent discrete entities and their relationships in a continuous space. The NSS arrangement generalizes many existing methods for representation learning.

Let V and L be discrete sets. A *relational structure* on V with labels in L is a sequence of maps $\phi_* := \{\phi_i : V^i \rightarrow L \mid i \in \mathbb{N}\}$. For example, a directed (non-simple) graph is expressed by $L = \{0, 1, 2, \dots\}$ and $\phi_i \equiv 0$ ($i \neq 2$), where $\phi_2(u, v)$ is the number of directed edges from $u \in V$ to $v \in V$. Another example for multi-labeled network is shown in Figure 1. Denote by $\Phi_m(L)$ the set of all the relational structures on a discrete set of cardinality m with labels in L .

We call a triple (V, L, ϕ_*) *relational data*. Our purpose is to give a continuous representation of relational data. We are particularly interested in *binary* relational data in which $\phi_i \equiv 0$ ($i > 2$).

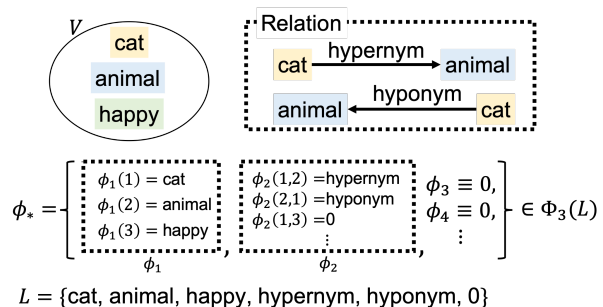


Figure 1. An example of the relational structure on a set V of cardinality 3. ϕ_1 describes the name of each element in V , and ϕ_2 describes the relation of each ordered pair of the elements.

Definition 1 Let X be a metric space. A sequence of spaces $A_1 \subset A_2 \subset \dots \subset A_n \subset X$ is called a Nested SubSpace

(NSS) with depth n in X . Denote by $\mathcal{S}_n(X)$ the set of all NSSs with depth n in X . An ordered collection of NSSs is called an NSS arrangement.

Definition 2 Fix n, X, L , and $V = \{v_1, \dots, v_m\}$. An *embedding* of (V, L, ϕ_*) is a map $f : V \rightarrow \mathcal{S}_n(X)$. A *reconstruction* is a map $g : (\mathcal{S}_n(X))^m \rightarrow \Phi_m(L)$.

This means that each node in V is represented by an NSS in X and the relational data by an NSS arrangement in X . The reconstruction map g has to be defined in a rule-based manner according to the type of relational structure.

The *reconstruction task* for a fixed reconstruction g is to find an embedding of (V, L, ϕ_*) such that $g(f(v_1), f(v_2), \dots, f(v_m))$ is close to ϕ_* .

Two or more NSSs can be related in a various manner by containment of their members. Our idea is to utilize this rich combinatorial structure among NSSs to represent binary (or possibly higher) relational data. We illustrate the generality of the NSS arrangement by showing that the majority of existing methods can be regarded as special cases of the NSS arrangement (see Figure 2).

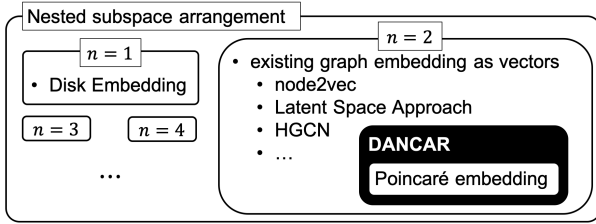


Figure 2. Hierarchy of NSS arrangement.

Hereinafter, let $D(x, r)$ be the closed ball of radius $r > 0$ centered at $x \in X$ in a metric space (X, d) .

A. **distance based** (e.g., Poincaré Embedding (Nickel & Kiela, 2017)) A basic idea for graph embedding is to represent nodes as points, and edges are drawn between two points within a specified distance threshold $\epsilon > 0$. This is a special case of the NSS arrangement with

- X is any metric space.
- $n = 2$.
- *embedding*: A node $v \in V$ is mapped to an NSS $(\{x_v\} \subset D(x_v, \epsilon))$.
- *reconstruction*: an undirected edge $u-v$ is present when $x_v \in D(x_u, \epsilon)$ (or equivalently, $x_u \in D(x_v, \epsilon)$).

For example, the Poincaré embedding utilizes the metric structure on the Poincaré disk to represent undirected graphs.

B. **inner product based** (e.g., (Chung, 1997)) A slight variation of the previous example is to use inner product for reconstruction, where edges are drawn between points whose inner product is larger than a specified threshold $\tau > 0$. This is also a special case of the NSS arrangement with

- $X = \mathbb{R}^k$ (or S^{k-1}).
- $n = 2$.
- *embedding*: A node $v \in V$ is mapped to an NSS $(\{x_v\} \subset H(x_v, \tau) \cup \{x_v\})$, where $H(x_v, \tau) = \{y \in X \mid \langle x_v, y \rangle > \tau\}$.
- *reconstruction*: an undirected edge $u-v$ is present when $x_u \in H(x_v, \tau)$.

C. **TransE** TransE (Bordes et al., 2013) is the first translation-based model for representing multi-relational data, that is, $|L| > 2$. TransE maps a node v to $x_v \in \mathbb{R}^k$ and a relation $l \in \{1, 2, \dots, s\}$ to $y_l \in \mathbb{R}^k$ to conform $x_u + y_l \approx x_v$ for a triple (u, l, v) . This is another example of the NSS arrangement with a threshold $\epsilon > 0$ with

- $X = \mathbb{R}^k$.
- $n = s + 1$.
- *embedding*: A node $v \in V$ is mapped to an NSS $(A_1^v = \{x_v + y_1\} \subset A_2^v = \{x_v + y_1, x_v + y_2\} \subset \dots \subset A_{s+1}^v = \{x_v + y_1, x_v + y_2, \dots, x_v + y_s\} \cup D(x_v, \epsilon))$.
- *reconstruction*: a relation (u, l, v) is present when $A_l^u \setminus A_{l-1}^u \subset D(x_v, \epsilon)$, where we regard $A_0^u = \emptyset$.

D. **Disk Embedding** Disk Embedding (Suzuki et al., 2019) can be regarded as an example of the NSS arrangement.

- $n = 1$.
- *embedding*: A node $v \in V$ is mapped to an NSS $D(x_v, r_v)$.
- *reconstruction*: A directed edge (u, v) is reconstructed when $D(x_u, r_u) \subset D(x_v, r_v)$.

Note that the Order Embedding (Vendrov et al., 2016) and the Hyperbolic Entailment Cones (Ganea et al., 2018) are special cases of the Disk Embedding, thus are also examples of the NSS arrangement.

E. **Multi-graphs** The NSS arrangement can also represent a multi-graph (non-simple directed graph); k -fold edges from u to v are represented by the relationship $A_1^u \subset A_{n-k+1}^v$ (see Figure 3).

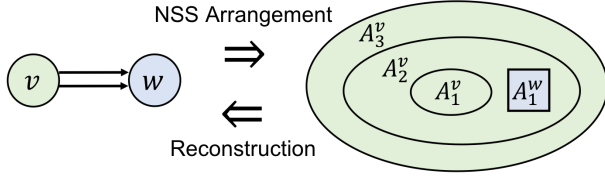


Figure 3. An example to represent multi-edge with NSS arrangement.

To close this section, we explain the rich combinatorial structure of the NSS by a simple example. First, note that Disk Embedding can only represent transitive relations: $A \rightarrow B$ and $B \rightarrow C$ automatically imply $A \rightarrow C$. In other words, Disk Embedding can represent only partially ordered sets. On the other hand, the NSS can distinguish the different relations among three objects depicted in Figure 4 by assigning, for example, the following containment conditions:

- (left) $B_1 \subset A_2, C_1 \subset B_2, C_1 \not\subset A_2, C_2 \not\subset A_2$
- (center) $B_1 \subset A_2, C_1 \subset B_2, C_1 \subset A_2, C_2 \not\subset A_2$
- (right) $B_1 \subset A_2, C_1 \subset B_2, C_1 \subset A_2, C_2 \subset A_2$.

The third one can be interpreted as a kind of directed ternary hyper-edge from A to (B, C) . The NSS can represent mathematical objects which are more general than partially ordered sets (see also Figure 6).

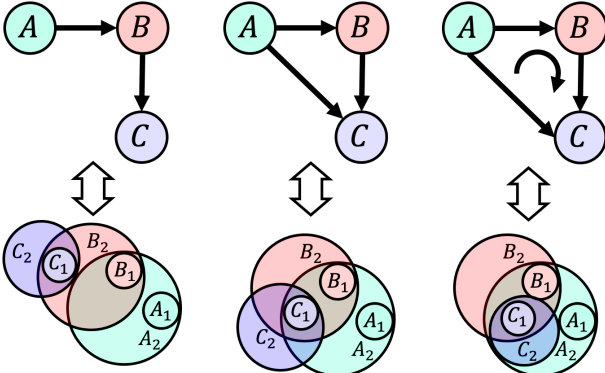


Figure 4. NSS can be used to capture higher relational data. Three types of relations among three objects above can be modelled by different containment relations of the corresponding NSSs.

3. DANCAR : Disk-ANCHOR Arrangement

We give an efficient implementation for the embedding of directed graphs by the NSS arrangement of depth 2 in the Euclidean space \mathbb{R}^k , which we call the DANCAR.

3.1. The DANCAR model

The DANCAR is a special case of the NSS arrangement:

- $X = \mathbb{R}^k$.
- $n = 2$.
- *embedding*: $V \ni v \mapsto A_1^v = \{x_v\} \subset A_2^v = D(c_v, r_v) \subset X$, where $A_1^v = \{x_v\}$ is called the *anchor* of the disk $A_2^v = D(c_v, r_v)$.
- *reconstruction*: a directed edge (v, w) is present when $x_w \in D(c_v, r_v)$ as illustrated in Figure 5.

We optimize $x_v, c_v \in \mathbb{R}^k$ and $r_v > 0$ to find an embedding with a good reconstruction (see §3.3).

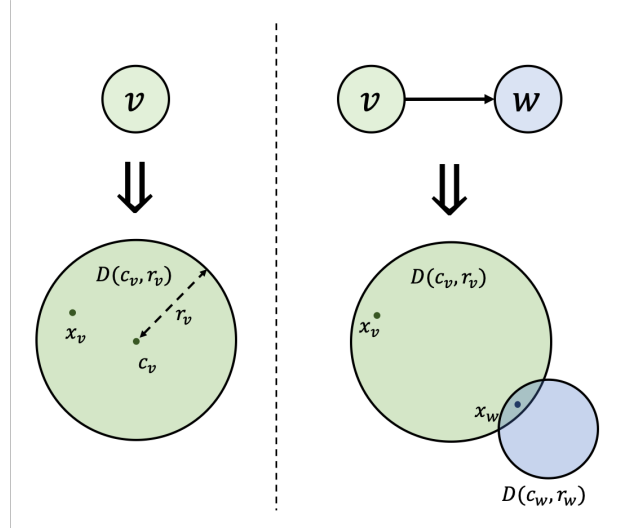


Figure 5. DANCAR represents a node by a pair of a disk and an anchor, and an edge by their membership relation.

Introduction of the anchor enables the DANCAR to represent non-symmetric and cyclic relations by containment (see Figure 6).

3.2. Representational capacity of DANCAR

At a glance, DANCAR is not so different from Disk Embedding, adding just anchor points. However, this simple trick to add the anchor points provides a great representational power to DANCAR. DANCAR can represent directed cycles (Figure 6) and “emulate” the hyperbolic metric with the Euclidean metric (Proposition 2).

First, it is easy to see the following proposition.

Proposition 1 Any directed tree can be embedded into \mathbb{R}^2 using the DANCAR.

Figure 7 graphically depicts how to embed a tree; we can choose the radius of a node shrinking exponentially with respect to the distance from the root. A concrete choice for the radius and the center position can be easily computed (see Algorithm 1).

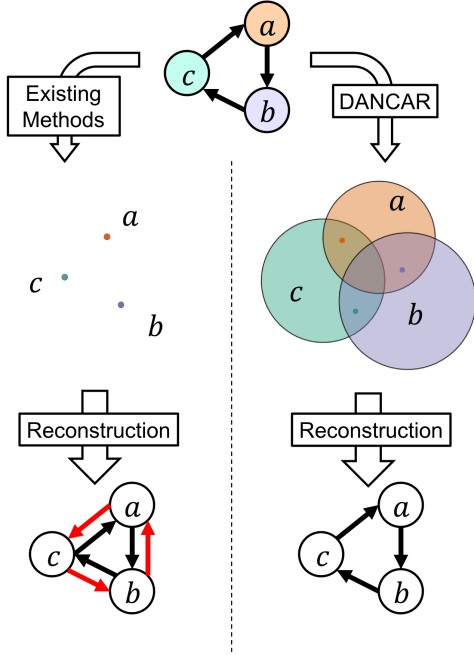


Figure 6. Directed cycles cannot be faithfully represented by a point cloud based embedding (left). Red arrows represent the edges which only exist in the reconstructed graph. DANCAR (right), on the other hand, can faithfully embed directed cycles.

Less trivial is the fact that the DANCAR model generalizes the Poincaré embedding model. In fact, since any tree can be faithfully embedded by the 2-dimensional Poincaré embedding, the above Proposition is a corollary of the following.

Proposition 2 The Poincaré embedding is a special case of the DANCAR which satisfies

1. $c_v = x_v / (K_v + 1)$,
2. $\|x_v\| < 1$,
3. $r_v = \sqrt{\frac{K_v}{K_v+1} \left(1 - \frac{1}{K_v+1} \|x_v\|^2\right)}$,

where

$$K_v := \frac{\cosh r - 1}{2} (1 - \|x_v\|^2).$$

This follows from the fact that a sphere in a Poincaré disk is also a sphere in the Euclidean space but with a different radius and a center. We provide a formal proof in Appendix. Proposition 2 indicates that our model can replicate the result of Poincaré embedding using the standard Euclidean norm.

Remark 1 We can view the DANCAR as the combination of a graph transformation and the Disk Embedding (see

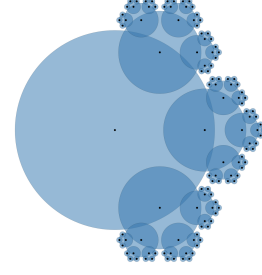


Figure 7. DANCAR embedding of the perfect ternary tree with depth = 5.

Figure 8). Given a directed graph $G = (V, E)$, let $G_2 = (V_2, E_2)$ be the directed bipartite graph defined by

$$V_2 := \{u_i \mid u \in V, i \in \{0, 1\}\}$$

$$E_2 := \{(u_0, v_1) \in V_2^2 \mid (u, v) \in E \text{ or } u = v\}.$$

The DANCAR embedding $\{(x_v, D(c_v, r_v))\}_{v \in V}$ is identified with the Disk embedding of G_2 , where the radius for v_1 nodes are fixed to zero.

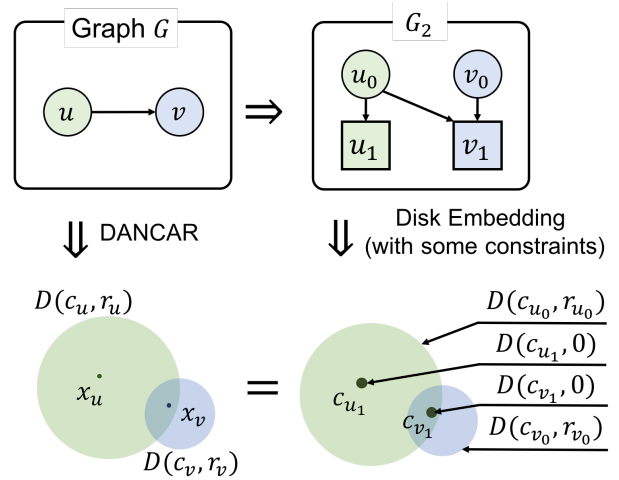


Figure 8. DANCAR as Disk Embedding of a transformed graph.

3.3. Construction of an embedding

Let (V, E) be a directed graph. We formulate the problem of finding a good embedding of (V, E) by the DANCAR as an optimization problem (see Figure 9). We introduce three loss functions with a hyperparameter, μ , called *margin*:

- **Positive Loss** : If there is a directed edge (v, w) , the anchor of the tail node w should be included by the disk of the head node v .

$$L_{\text{pos}} := \frac{1}{|E|} \sum_{(v,w) \in E} \text{ReLU}(d(c_v, x_w) - r_v + \mu). \quad (1)$$

- **Negative Loss** : If there is no directed edge (v, w) , the anchor of the tail node w should not be included by the disk of the head node v .

$$L_{\text{neg}} := \frac{1}{|E^c|} \sum_{(v,w) \in E^c} \text{ReLU}(r_v - d(c_v, x_w) + \mu), \quad (2)$$

where $E^c = \{(v, w) \in V \times V \mid (v, w) \notin E, v \neq w\}$. When sampling from E^c is computational intractable, we approximate E^c by $\{(v, w) \in V \times V \mid v \neq w\}$.

- **Anchor Loss** : the anchor should be contained in the disk. This can be regarded as a regularization.

$$L_{\text{anc}} := \frac{1}{|V|} \sum_{v \in V} \text{ReLU}(d(c_v, x_v) - r_v + \mu). \quad (3)$$

The total loss function of the DANCAR can be written as the weighted sum of the above loss functions:

$$L_{\text{DANCAR}}(\{c_v\}_{v \in V}, \{r_v\}_{v \in V}, \{x_v\}_{v \in V}) := L_{\text{pos}} + \lambda_{\text{neg}} L_{\text{neg}} + \lambda_{\text{anc}} L_{\text{anc}}, \quad (4)$$

where $\lambda_{\text{neg}} \geq 0$ and $\lambda_{\text{anc}} \geq 0$ are hyperparameters.

An embedding is obtained by optimizing the total loss function by a stochastic gradient descent.

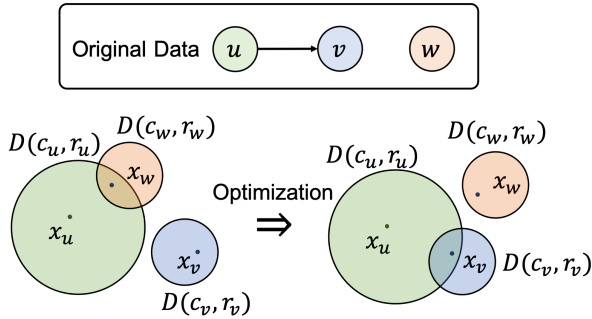


Figure 9. Illustration of an optimization process.

Implementation

All experiments were implemented in Chainer 7.4.0. Source code is publicly available at <https://github.com/KyushuUniversityMathematics/DANCAR>.

4. Experiment: Visualization

In this section, we show the potential of the DANCAR in visualizing graphs.

Figure 10 illustrates with a toy example that the DANCAR embedding is sensitive to the change in the topology of the graph. The embedding faithfully captures the difference

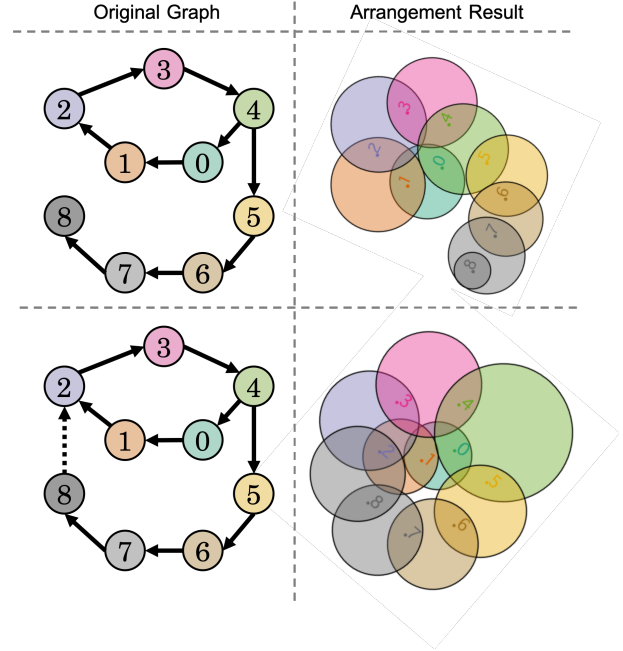


Figure 10. Embeddings by the DANCAR are sensitive to the change in the topology of the graph.

of two graphs, which is the existence of the edge $(8, 2)$, illustrated as a broken line.

For a practical application, we applied the DANCAR to a subgraph of the Twitter network¹. Each directed edge (u, v) represents that account v follows account u . We randomly picked 1,000 accounts, keeping the weak connectivity of the graph. The graph has 3,188 edges with the maximum in-degree 405 and the maximum out-degree 47. The result of a DANCAR embedding of the graph into the two-dimensional Euclidean space is shown in Figure 11. Both clusters and hierarchies of the graph can be observed through the visualization. The size of each disk roughly corresponds to the out-degree of the node. In fact, Spearman's rank correlation coefficient between the radius and out-degree is 0.628.

Figure 11 (C) shows the account v_{out} with the highest out-degree and its successors (followers). The disk of v_{out} is depicted by the large black circle as v_{out} is followed by a large number of other accounts. On the other hand, we see that most of the accounts that follow v_{out} have small radii (except for the gray one), reflecting the fact that they are not followed by many accounts; in fact, they are followed by at most one account in the original network.

Figure 11 (D) focuses on the account v_{in} with the highest in-degree and its predecessors. The anchor and the disk (of tiny radius) of v_{in} are depicted by the yellow dot. We

¹<https://snap.stanford.edu/data/twitter-2010.html>

observe that there are disks of various sizes around v_{in} . The observation reflects the fact that v_{in} follows both popular and non-popular accounts.

5. Experiment: Reconstruction and Link prediction

In this section, we evaluate how well DANCAR can represent directed graphs by the reconstruction and the link prediction tasks. The purpose of DANCAR is to faithfully capture the existence and the non-existence of edges. Note that this is, in some sense, opposite to the link prediction task (Figure 12). Nonetheless, we show that DANCAR performs well for both tasks by choosing appropriate embedding dimensions.

For the reconstruction experiment, we compared the edge existence in the original graph and the reconstructed graph from the embedding. For the link prediction experiment, we computed the embedding using a half of edges randomly chosen from the original graph and compared the edge existence in the original graph and the reconstructed graph from the embedding.

As a practical target graph², we used the largest weakly-connected component of a noun closure of the **WordNet** (Miller, 1995). We removed the root and then took the transitive closure and obtained a DAG consisting of 82,076 nodes and 660,846 directed edges.

The hyper-parameters for the DANCAR were chosen as follows. The margin parameter μ was fixed to 0.01. We tested with \mathbb{R}^{10} and \mathbb{R}^{20} as the embedding space. We experimented with the hyper-parameters $8 \leq \lambda_{neg} \leq 1000$, $\lambda_{anc} \in \{1, 10\}$ and the best results were chosen. We observed that with higher embedding dimension, smaller λ_{neg} performed better. For optimization with a stochastic gradient descent, we used two different batch sizes $b_1 = 10,000$ for the positive loss and the vertex loss, and $b_2 = 100,000$ for the negative loss to account for the sparsity of the graph. We randomly selected the negative samples for each iteration. We used the Adam (Kingma & Ba, 2015) optimizer with parameters $\alpha = 0.05$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

Initialization of the parameters c_v, r_v, x_v have been observed to be important, and we set:

- c_v were sampled from the uniform distribution on $[-1, 1]^k$ ($k = 10, 20$).
- $r_v = 0.1$ for any $v \in V$.
- $x_v = c_v$ for any $v \in V$.

This initial arrangement represents a graph with few edges.

²We observed that the subgraph of the Twitter network we used in the previous section was successfully embedded in the 10 dimensional Euclidean space perfectly faithfully (that is, with F1 score one).

Thus, in the beginning, the positive loss was dominant, and the chance of the gradient vanishing problem was reduced.

As a comparison, we performed the same task with the Poincaré embedding and the Disk Embedding. For the Poincaré embedding, we used the implementation made available by the original authors of (Nickel & Kiela, 2017)³. We took 50 negative samplings per positive sample for the optimization of the Poincaré Embedding. For the Poincaré embedding, instead of choosing a single radius for all vertices for reconstruction, we used the mean average precision (mAP) for evaluation as was done in the original paper (Nickel & Kiela, 2017). Note that this is in a sense choosing an optimal radius for each vertex. There is trade-off between precision and recall, and the F1 score is maximized when they agree. Therefore, in most cases mAP is much higher than the F1 score computed for a choice of the radius.

For the Disk Embedding, we used our own implementation since our implementation is quite similar to the implementation of the Disk Embedding. We used the same parameters and as the DANCAR except for the parameters for the anchors.

The result is shown in Table 2. We observed that our DANCAR performed considerably better than the Poincaré embedding and the Disk Embedding. We speculate that the absence of the root node in the graph has affected the performance of the Poincaré embedding and the Disk Embedding. In contrast, our method does not depend on the existence of the root and was able to reconstruct the graph effectively. It should be noted that in the link prediction task, due to the high representational capacity of DANCAR, the higher dimensional embeddings result in lower recall rate (see Figure 13). When the embedding dimension is high enough, the DANCAR faithfully captures the (non-)existence of edges, and those edges which were not present in the training data were not reconstructed. A similar phenomena should be observed for the Disk Embedding and the Poincaré embedding when we use a huge dimensional embedding space.

6. Conclusion

In this study, we introduced the Nested SubSpace (NSS) arrangement, which generalizes many of the existing methods for relational data representation. As a special case of NSS arrangement, we provided a practical implementation of the Disk-ANCHor ARrangement (DANCAR).

The visualization and large-scale embedding experiments highlighted the representation capacity of the DANCAR. We observed that the DANCAR captures the cluster and hierarchy structures simultaneously. The DANCAR outperformed existing methods in the reconstruction and the link

³<https://github.com/facebookresearch/poincare-embeddings>

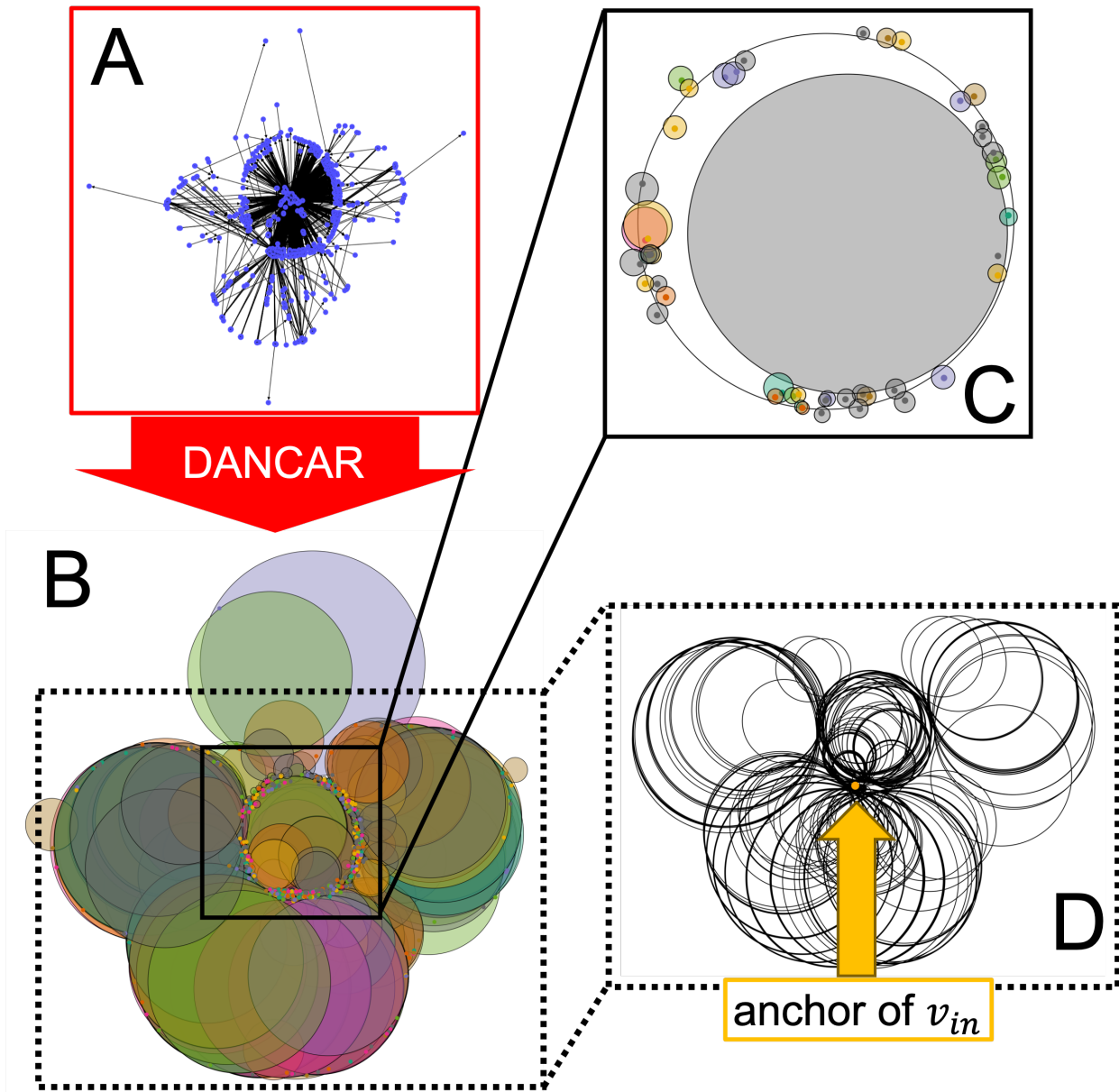


Figure 11. The visualization of a part of the Twitter network by the DANCAR. (A) a part of the Twitter network, (B) its DANCAR embedding in \mathbb{R}^2 , (C) Neighbor of the node with the highest out-degree, (D) Neighbor of the node with the highest in-degree.

Table 2. The precision and the F1 score of the reconstruction (100% training) and the link prediction (50% training) tasks. For the link prediction task, the evaluation scores were computed for the entire edges.

WordNet	method	100% training				50% training			
		10dim		20dim		10dim		20dim	
		F1 score	mAP	F1 score	mAP	F1 score	mAP	F1 score	mAP
	DANCAR (Proposed)	0.982	-	0.993	-	0.787	-	0.709	-
	Poincaré Embedding	-	0.635	-	0.654	-	0.675	-	0.675
	Disk Embedding	0.057	-	0.052	-	0.151	-	0.114	-

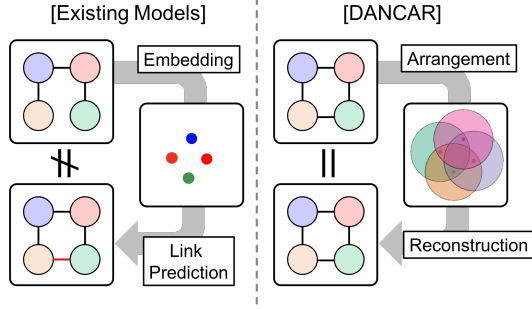


Figure 12. Non-existing edges in the training graph should be reconstructed in the link prediction task, where as they should not in the reconstruction task.

prediction tasks of a large-scale DAG in a relatively low dimensional space. The rich combinatorial structure of the DANCAR lead to an accurate representation of graphs.

Although in this paper we focused mainly on the DANCAR, our general framework of the NSS arrangement could be used for learning representation of various relational data.

Algorithm 1 Tree embedding

Input: tree $G = (V, E)$ with the root $z \in V$
Output: the set of embedded disks $\{D(c_v, r_v)\}_{v \in V}$
 Let $n = \max_{v \in V}(\#\mathcal{N}(v))$, $\alpha = -\frac{(n-1)\pi}{2n}$, $p = \cos(\alpha)$,
 $q = \cos(2\alpha)$
 $t = \frac{\sqrt{(p+q)^2 + 4p - p + q}}{2(q+1)}$, $k = \frac{1}{\sqrt{1+t^2}}$
 $c_z = \mathbf{0}$, $r_z = 1$, $\theta_z = 0$, $S = \{z\}$
while $S \neq \emptyset$ **do**
 pop u from S
 Let $\varphi = \alpha$
 for $v \in \mathcal{N}(u)$ **do**
 $\theta_v = \theta_u + \varphi$
 $c_v = c_u + r_u k(\cos(\theta_v), \sin(\theta_v))$
 $r_v = tr_u$
 push v to S
 $\varphi = \varphi + \frac{\pi}{n}$
 end for
end while

Appendix: Proof of Proposition 2

Let $d(x, y)$ be the metric on the Poincaré ball, i.e.,

$$d(x, y) := \operatorname{arcosh} \left(1 + 2 \frac{\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)} \right).$$

Let $D_P(x, r)$ (respectively, $D_E(x, r)$) be the closed ball centered at x and of radius $r > 0$ with respect to the metric d (respectively, the Euclidean metric).

For all $r > 0$ and $a, x \in D_E(0, 1)$, we have the following

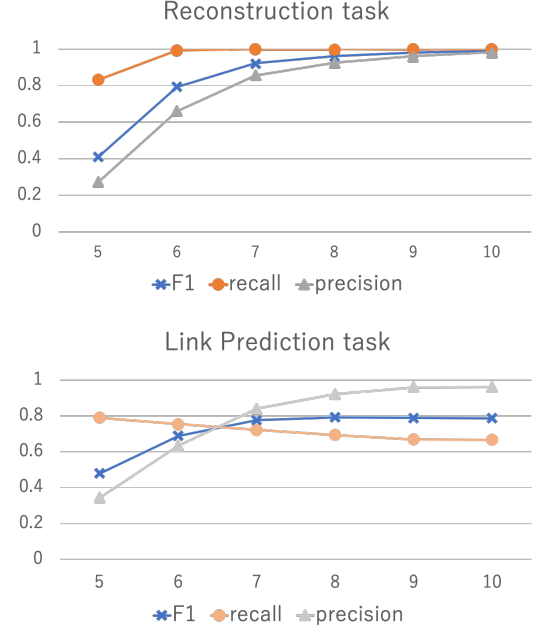


Figure 13. The performance of the reconstruction and the link prediction tasks with varying embedding dimensions.

equivalence.

$$\begin{aligned} x \in D_P(a, r) &\iff d(x, a) \leq r \\ &\iff \frac{\|x - a\|^2}{(1 - \|x\|^2)(1 - \|a\|^2)} \leq \frac{\cosh r - 1}{2} \\ &\iff \|x\|^2 - 2\langle x, a \rangle + \|a\|^2 \leq K(1 - \|x\|^2) \\ &\iff (K + 1)\|x\|^2 - 2\langle x, a \rangle \leq K - \|a\|^2 \\ &\iff (K + 1) \left\| x - \frac{1}{K + 1} a \right\|^2 - \frac{\|a\|^2}{K + 1} \leq K - \|a\|^2 \\ &\iff \left\| x - \frac{1}{K + 1} a \right\|^2 \leq \frac{K}{K + 1} \left(1 - \frac{1}{K + 1} \|a\|^2 \right) \\ &\iff x \in D_E \left(\frac{1}{K + 1} a, \sqrt{\frac{K}{K + 1} \left(1 - \frac{1}{K + 1} \|a\|^2 \right)} \right) \end{aligned}$$

where

$$K := \frac{\cosh r - 1}{2} (1 - \|a\|^2).$$

Acknowledgement

This research project was supported by the Japan Science and Technology Agency (JST), the Core Research of Evolutionary Science and Technology (CREST), the Center of Innovation Science and Technology based Radical Innovation and Entrepreneurship Program (COI Program), JSPS KAKENHI Grant No. JP 16H01707

References

- Balasubramanian, M. and Schwartz, E. L. The isomap algorithm and topological stability. *Science*, 295(5552): 7–7, 2002.
- Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pp. 2787–2795, 2013.
- Cao, S., Lu, W., and Xu, Q. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 891–900, 2015.
- Chami, I., Ying, Z., Ré, C., and Leskovec, J. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 4869–4880, 2019.
- Chung, F. R. K. *Spectral Graph Theory*. American Mathematical Society, Providence, RI, 1997. ISBN 0821803158 9780821803158.
- Cox, T. F. and Cox, M. A. *Multidimensional scaling*. Chapman and hall/CRC, 2000.
- Dong, Y., Chawla, N. V., and Swami, A. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 135–144, 2017.
- Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pp. 2224–2232, 2015.
- Ebisu, T. and Ichise, R. Toruse: Knowledge graph embedding on a lie group. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- Feng, Y., You, H., Zhang, Z., Ji, R., and Gao, Y. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019.
- Ganea, O., Becigneul, G., and Hofmann, T. Hyperbolic entailment cones for learning hierarchical embeddings. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1646–1655, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/ganea18a.html>.
- Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864. ACM, 2016.
- Khasanova, R. and Frossard, P. Graph-based isometry invariant representation learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1847–1856. JMLR. org, 2017.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*. 2015.
- Kipf, T. N. and Welling, M. Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*, 2016.
- Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. S. Gated graph sequence neural networks. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1511.05493>.
- Miller, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., and Bronstein, M. M. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5115–5124, 2017.
- Nguyen, D. Q., Sirts, K., Qu, L., and Johnson, M. STransE: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 460–466, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1054. URL <https://www.aclweb.org/anthology/N16-1054>.
- Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30*. 2017.
- Ou, M., Cui, P., Pei, J., Zhang, Z., and Zhu, W. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1105–1114, 2016.
- Pan, S., Wu, J., Zhu, X., Zhang, C., and Wang, Y. Tri-party deep network representation. *Network*, 11(9):12, 2016.
- Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the*

- 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710. ACM, 2014.
- Singh, A. P. and Gordon, G. J. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 650–658, 2008.
- Suzuki, A., Enokida, Y., and Yamanishi, K. Riemannian transe: Multi-relational graph embedding in non-euclidean space. 2018.
- Suzuki, R., Takahama, R., and Onoda, S. Hyperbolic disk embeddings for directed acyclic graphs. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6066–6075, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL <http://proceedings.mlr.press/v97/suzuki19a.html>.
- Tu, K., Cui, P., Wang, X., Wang, F., and Zhu, W. Structural deep embedding for hyper-networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- Vendrov, I., Kiros, R., Fidler, S., and Urtasun, R. Order-embeddings of images and language. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1511.06361>.
- Wang, D., Cui, P., and Zhu, W. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1225–1234. ACM, 2016.
- Wang, H., Wang, J., Wang, J., Zhao, M., Zhang, W., Zhang, F., Xie, X., and Guo, M. Graphgan: Graph representation learning with generative adversarial nets. In *Thirty-second AAAI conference on artificial intelligence*, 2018.
- Wang, Z., Zhang, J., Feng, J., and Chen, Z. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.
- Yanardag, P. and Vishwanathan, S. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.
- Yang, D., Qu, B., Yang, J., and Cudre-Mauroux, P. Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach. In *The World Wide Web Conference*, pp. 2147–2157, 2019.