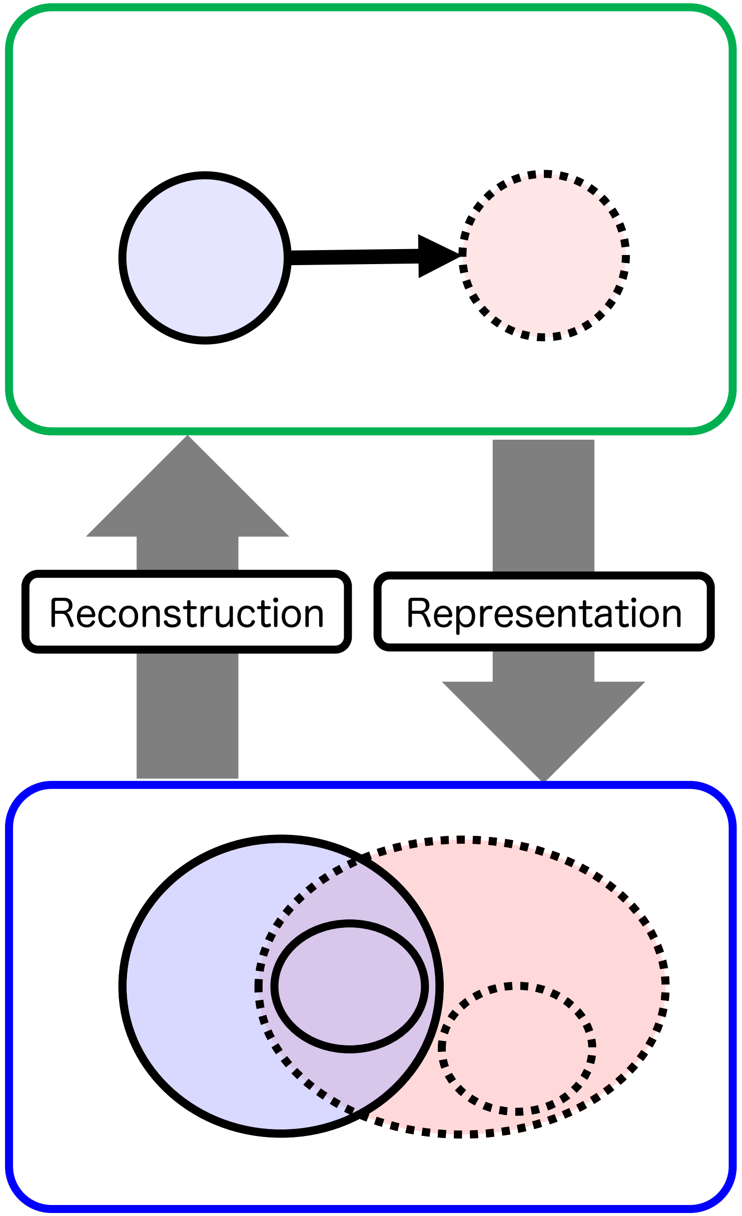


Nested Subspace Arrangement for Representation of Relational Data

Nozomi Hata, Shizuo Kaji,
Akihiro Yoshida, Katsuki Fujisawa

Kyushu University, Japan

ICML 2020

Contribution

We propose

- Nested SubSpace (NSS) arrangement: a general framework for representation of relational data in continuous space

- Disk-ANChor ARrangement (DANCAR): to represent directed graphs

The DANCAR can be used for :

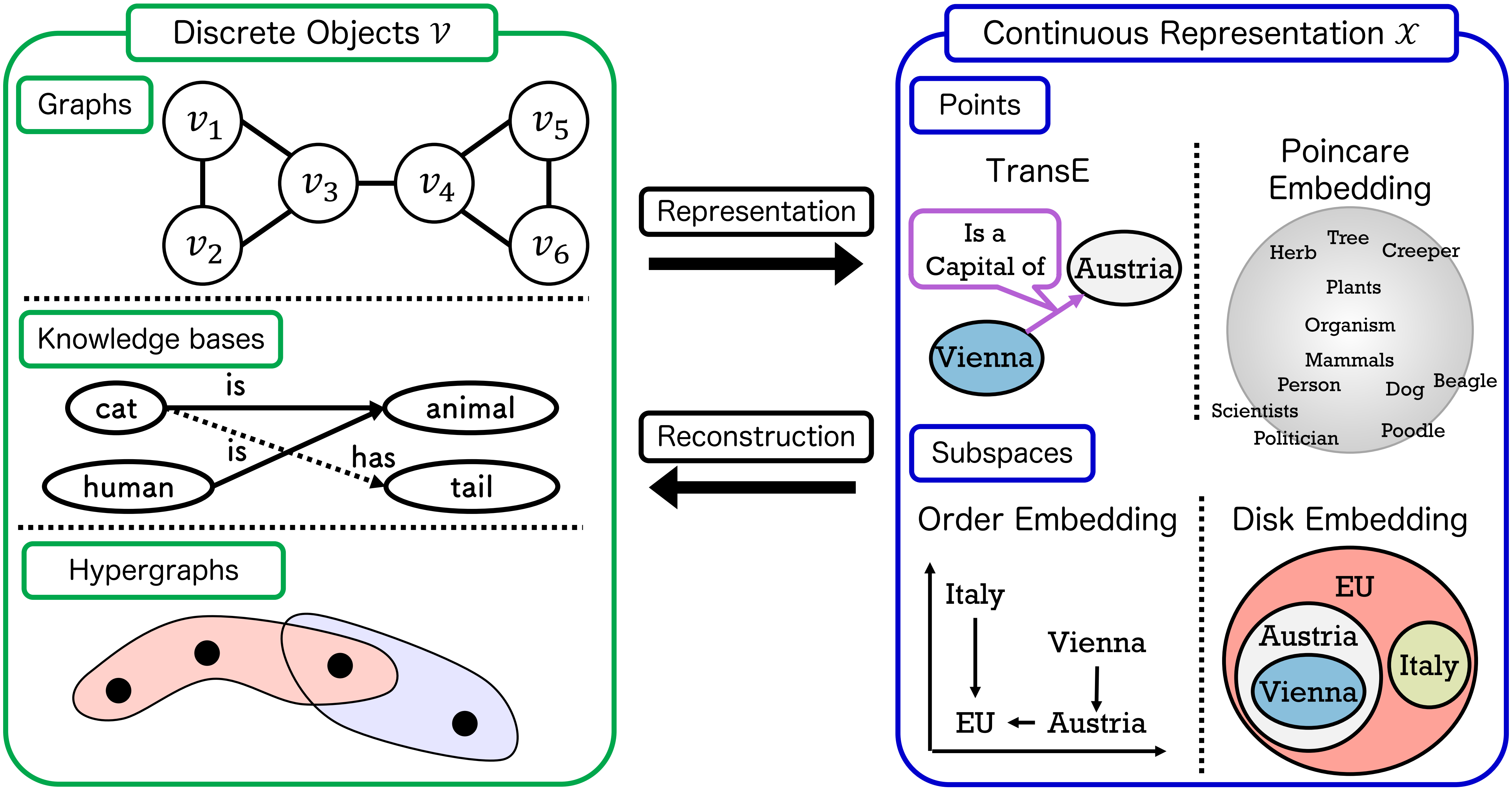
- Visualization of a large-scale network to reveal both cluster structure and hierarchical structure

- Representation of a directed graph accurately in terms of the edge reconstruction task

a python implementation is available at https://github.com/KyushuUniversityMathematics/DANCAR

Contents

1. Introduction



Contents

2. Definition : Relational Data

Discrete Objects $\mathcal{V}$

⇩

Relational data

- Entities V
- Types of relations L
- Relational Structure $\phi_* = \{\phi_i : V^i \to L\}_{i=1}^{\infty}$

  ϕ_i : relations among i entities

ex) knowledge base → Relational Data

$V = \{1, 2, 3\}$

$L = \{\text{cat, animal, happy, hypernym, hyponym}, 0\}$

$$\phi_* = \left[\underbrace{\begin{matrix} \phi_1(1) = \textbf{cat} \\ \phi_1(2) = \textbf{animal} \\ \phi_1(3) = \textbf{happy} \end{matrix}}_{\phi_1}, \underbrace{\begin{matrix} \phi_2(1,2) = \textit{hypernym} \\ \phi_2(2,1) = \textit{hyponym} \\ \phi_2(1,3) = 0 \\ \vdots \end{matrix}}_{\phi_2} \begin{matrix} \phi_3 \equiv 0, \\ \phi_4 \equiv 0, \\ \vdots \end{matrix} \right]$$

2. Definition : Relational Data

Discrete Objects $\mathcal{V}$

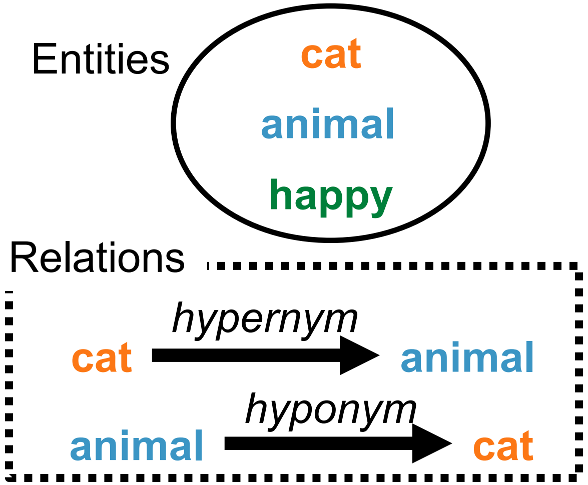
⇩

Relational data

- Entities V
- Types of relations L
- Relational Structure $\phi_* = \{\phi_i : V^i \to L\}_{i=1}^{\infty}$

  ϕ_i : relations among i entities

ex) knowledge base → Relational Data



⇨

$$V = \{1, 2, 3\}$$

$$L = \{\text{cat, animal, happy, hypernym, hyponym}, 0\}$$

$$\phi_* = \left\{ \underbrace{\begin{matrix} \phi_1(1) = \textbf{cat} \\ \phi_1(2) = \textbf{animal} \\ \phi_1(3) = \textbf{happy} \end{matrix}}_{\phi_1}, \underbrace{\begin{matrix} \phi_2(1{,}2) = \textit{hypernym} \\ \phi_2(2{,}1) = \textit{hyponym} \\ \phi_2(1{,}3) = 0 \\ \vdots \end{matrix}}_{\phi_2} \begin{matrix} \phi_3 \equiv 0, \\ \phi_4 \equiv 0, \\ \vdots \end{matrix} \right\}$$

2. Definition : Nested SubSpace (NSS)

Continuous Representation $\mathcal{X}$

⇩

Nested SubSpace (NSS)

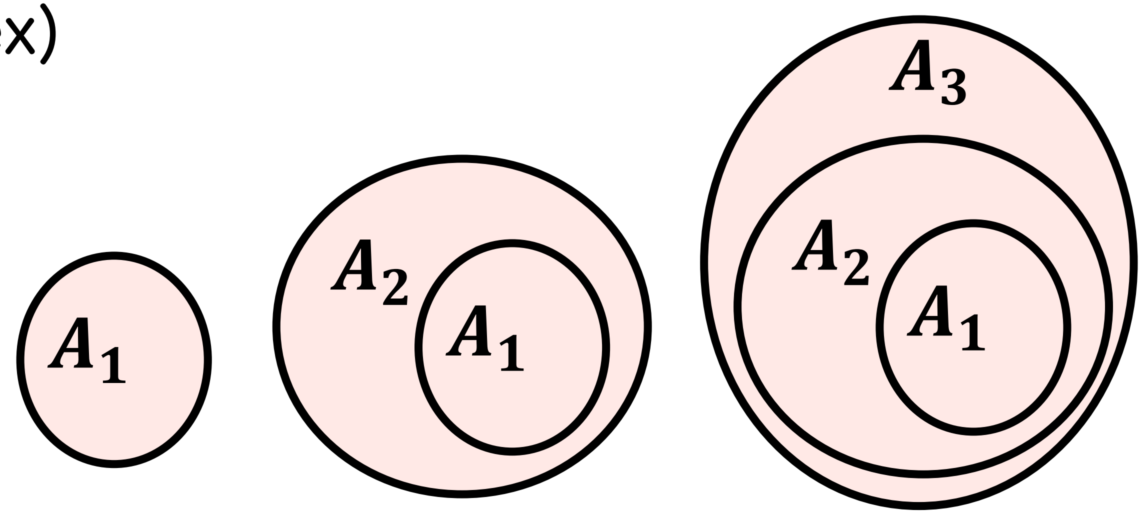
- NSS with depth n : sequence of spaces

$$A_1 \subset \cdots \subset A_n (\subset X) \in \mathcal{S}_n(X)$$

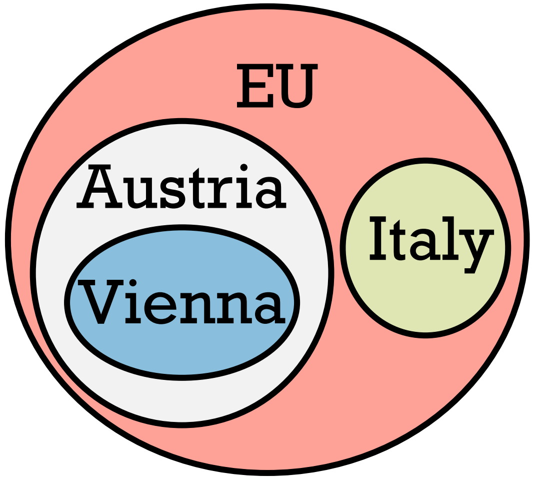
- NSS arrangement with depth n

= collection of NSSs with depth n

ex)

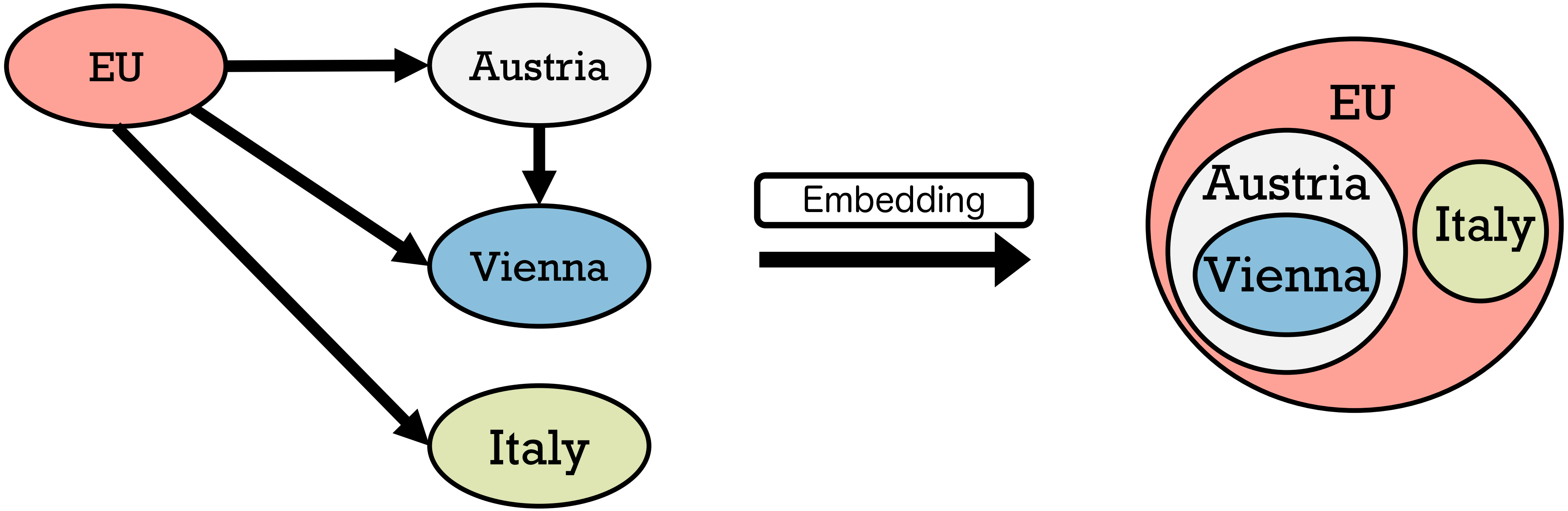


An example of an NSS with depth 1, 2, 3.



The result of Disk embedding = NSS arrangement with depth 1

2. Definition : Nested SubSpace (NSS)



Disk embedding = NSS arrangement with depth 1

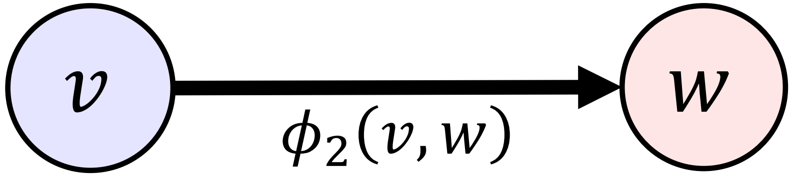
- all nodes is assigned to a disk
- a disk is an NSS with depth 1

2. NSS arrangement

Relational Data

- Nodes $V = \{v_1, \dots, v_m\}$
- Types of relations L
- Relational Structure

$\phi_* = \{\phi_i : V^i \to L\}_{i=1}^{\infty} \in \Phi_m(L)$



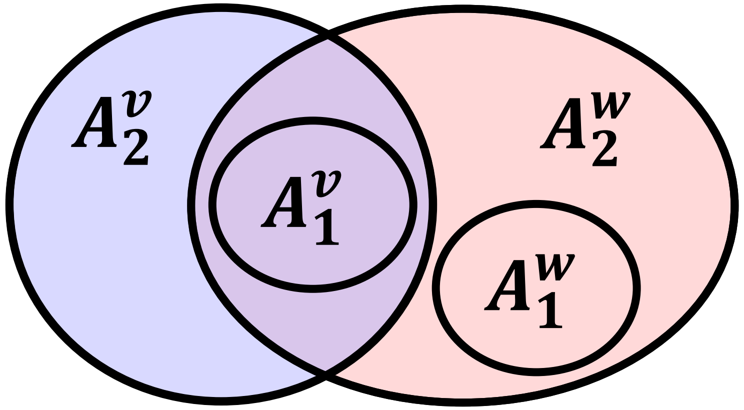
Embedding

Reconstruction

NSS Arrangement

Collection of NSSs

$\{(A_1^v, \cdots, A_n^v)\}_v$



- Embedding
  = map $f: v \mapsto (A_1^v, \cdots, A_n^v)$
- Reconstruction
  = collection of maps
  $g_m: \{(A_1^v, \cdots, A_n^v)\}_v \mapsto \phi_*$

Embedding

Reconstruction

Reconstruction task

= find map f s.t.

$g_{|V|}(f(V)) = \phi_*$

2. NSS arrangement : Embedding

Relational Data

- Nodes $V = \{v_1, \dots, v_m\}$
- Types of relations L
- Relational Structure

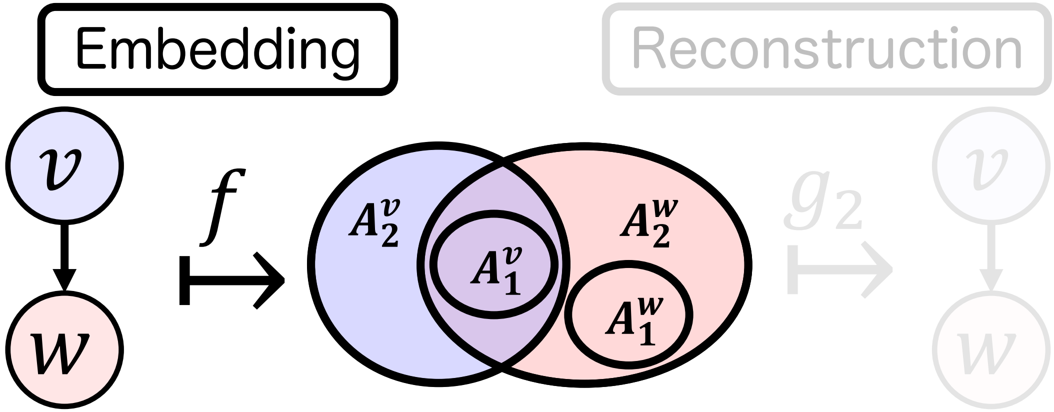
$\phi_* = \{\phi_i : V^i \to L\}_{i=1}^{\infty} \in \Phi_m(L)$

NSS Arrangement

Collection of NSSs

$\{(A_1^v, \cdots, A_n^v)\}_v$

- Embedding
  = map $f : v \mapsto (A_1^v, \cdots, A_n^v)$
- Reconstruction
  = collection of maps
  $g_m : \{(A_1^v, \cdots, A_n^v)\}_v \mapsto \phi_*$



Reconstruction task

= find map f s.t.

$g_{|V|}(f(V)) = \phi_*$

2. NSS arrangement : Reconstruction

Relational Data

- Nodes $V = \{v_1, \dots, v_m\}$
- Types of relations L
- Relational Structure

$\phi_* = \{\phi_i : V^i \to L\}_{i=1}^{\infty} \in \Phi_m(L)$

Embedding

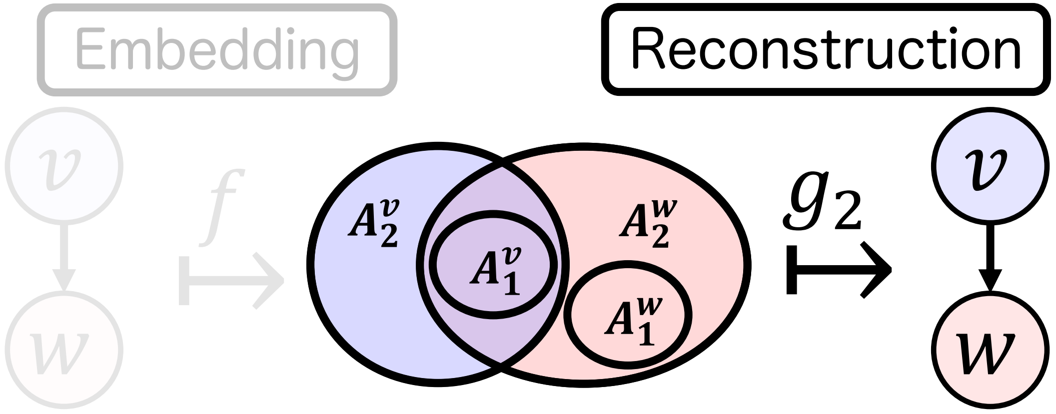
Reconstruction

NSS Arrangement

Collection of NSSs

$\{(A_1^v, \cdots, A_n^v)\}_v$

- Embedding
  = map $f : v \mapsto (A_1^v, \cdots, A_n^v)$
- Reconstruction
  = collection of maps
  $g_m : \{(A_1^v, \cdots, A_n^v)\}_v \mapsto \phi_*$



Reconstruction task

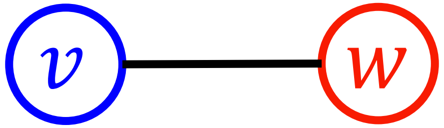
= find map f s.t.

$$g_{|V|}(f(V)) = \phi_*$$

2.A. Poincaré Embedding as NSS Arrangement

Undirected Graph

- Nodes V
- Edges $E \Rightarrow \phi_2 : V^2 \to \{0,1\}$



$$g_{|V|}\left(\left(\{c_i\}, D(c_i, \varepsilon)\right)_{i \in V}\right)_2 (v, w)$$

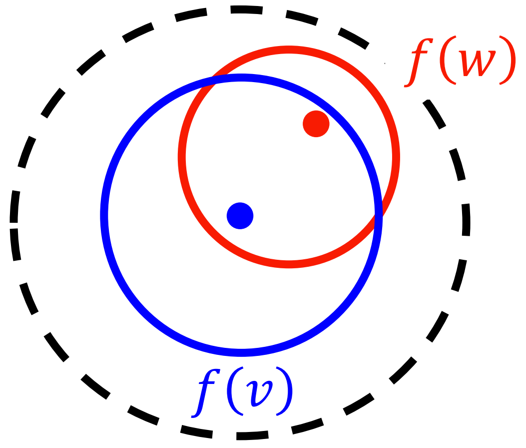
$$= \begin{cases} 1 \ (\text{if } \{c_w\} \subset D(c_v, \varepsilon)) & \Rightarrow v - w \\ 0 \ (\text{if } \{c_w\} \not\subset D(c_v, \varepsilon)) & \Rightarrow v \quad w \end{cases}$$

Embedding →

← Reconstruction

Center ● + Disk ○

$$f(v) = \left(\{c_v\}, D(c_v, \varepsilon)\right)$$



Poincaré ball

- Embedding
  = map $f : v \mapsto (A_1^v, \cdots, A_n^v)$
- Reconstruction
  = collection of maps
  $g_m : \{(A_1^v, \cdots, A_n^v)\}_v \mapsto \phi_*$

Embedding

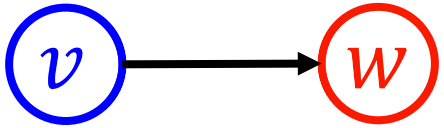
Reconstruction

Reconstruction task
= find map f s.t.
$g_{|V|}(f(V)) = \phi_*$

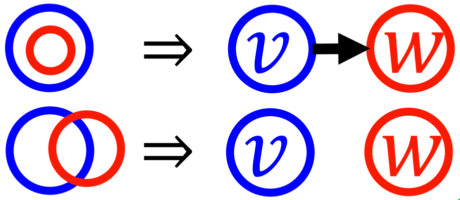
2.D. Disk Embedding as NSS Arrangement

Directed Acyclic Graph

- Nodes V
- Edges $E \Rightarrow \phi_2: V^2 \to \{0,1\}$



$$g_{|V|}\left(\left(D(c_i,r_i)\right)_{i\in V}\right)_2 (v,w) = \begin{cases} 1 \; (\text{if } D(c_w,r_w) \subset D(c_v,r_v)) \\ 0 \; (\text{if } D(c_w,r_w) \not\subset D(c_v,r_v)) \end{cases}$$

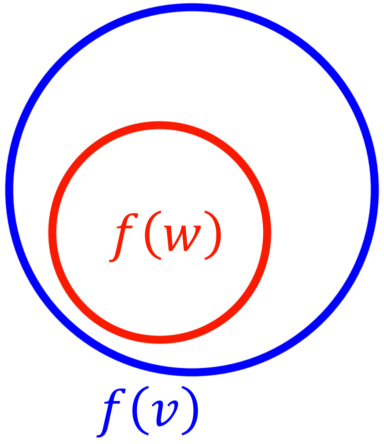


Embedding

Reconstruction

Disk ○

$$f(v) = D(c_v, r_v)$$



- Embedding
  = map $f: v \mapsto (A_1^v, \cdots, A_n^v)$
- Reconstruction
  = collection of maps
  $g_m: \{(A_1^v, \cdots, A_n^v)\}_v \mapsto \phi_*$

Embedding

Reconstruction

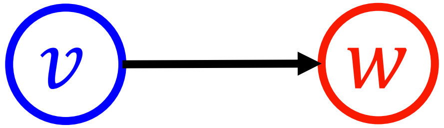
Reconstruction task
= find map f s.t.
$g_{|V|}(f(V)) = \phi_*$

Contents

3.1. DANCAR : Disk-ANChor ARrangement

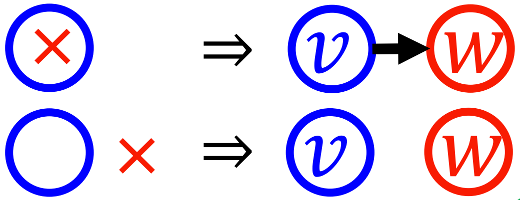
Directed Graph

- Nodes V
- Edges $E \Rightarrow \phi_2 : V^2 \to \{0,1\}$



$$g_{|V|}\left(\left(\{x_i\}, D(c_i, r_i)\right)_{i \in V}\right)_2 (v, w)$$

$$= \begin{cases} 1 \ (\text{if } \{x_w\} \subset D(c_v, r_v)) \\ 0 \ (\text{if } \{x_w\} \not\subset D(c_v, r_v)) \end{cases}$$

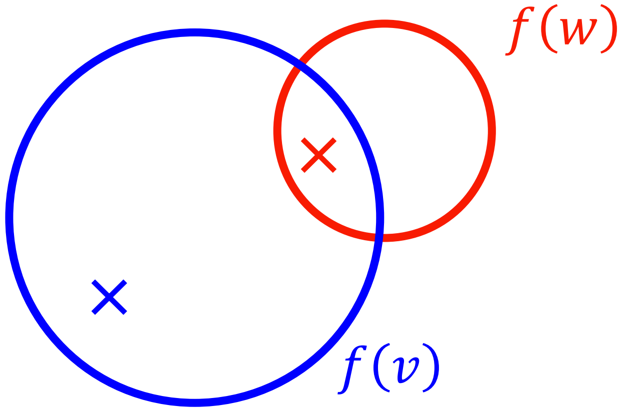


Embedding

Reconstruction

Anchor× & Disk○

$$f(v) = \left(\{x_v\}, D(c_v, r_v)\right)$$



- Embedding = map $f : v \mapsto (A_1^v, \cdots, A_n^v)$
- Reconstruction = collection of maps $g_m : \{(A_1^v, \cdots, A_n^v)\}_v \mapsto \phi_*$

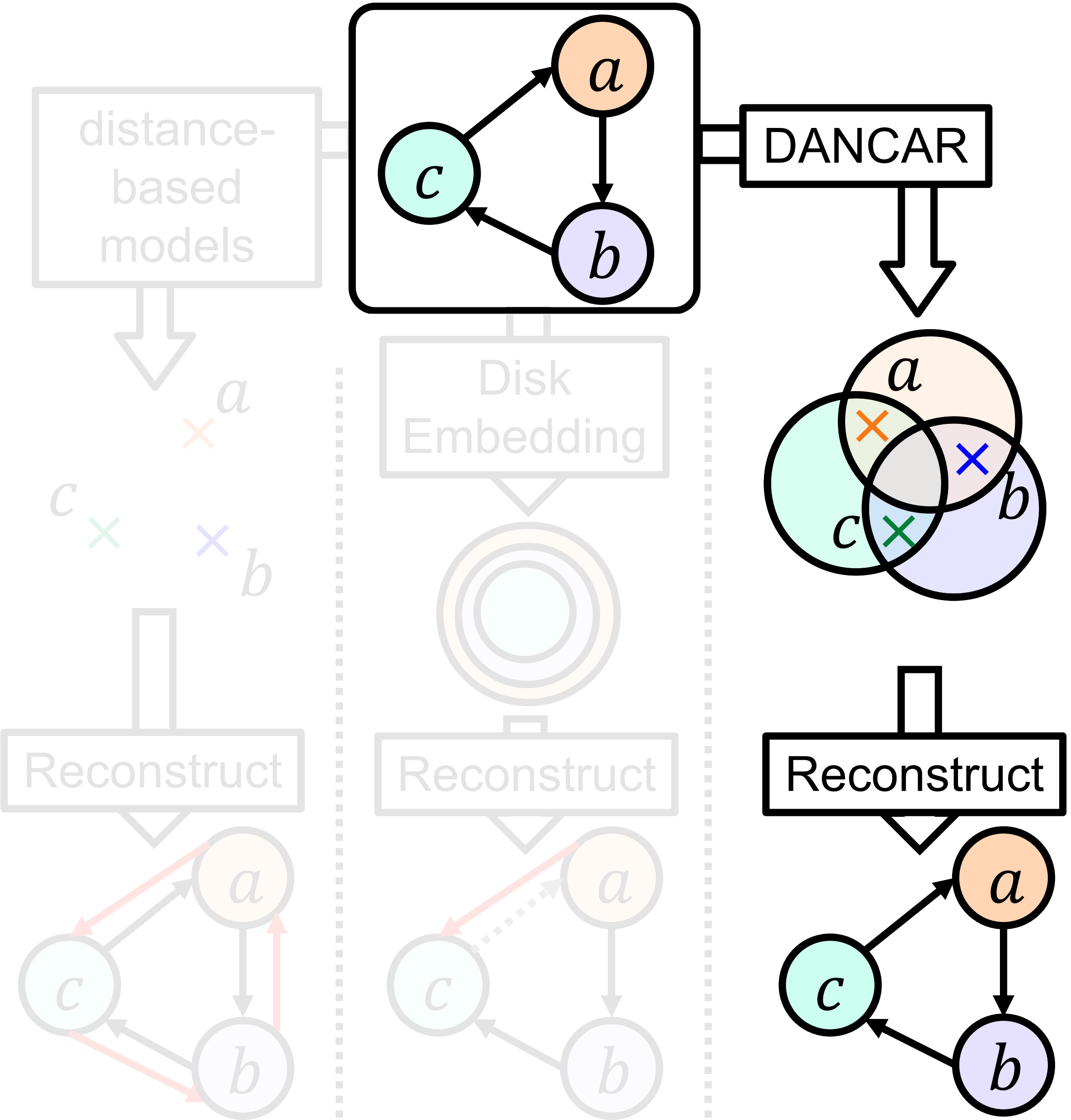
Embedding

Reconstruction

Reconstruction task = find map f s.t. $g_{|V|}(f(V)) = \phi_*$

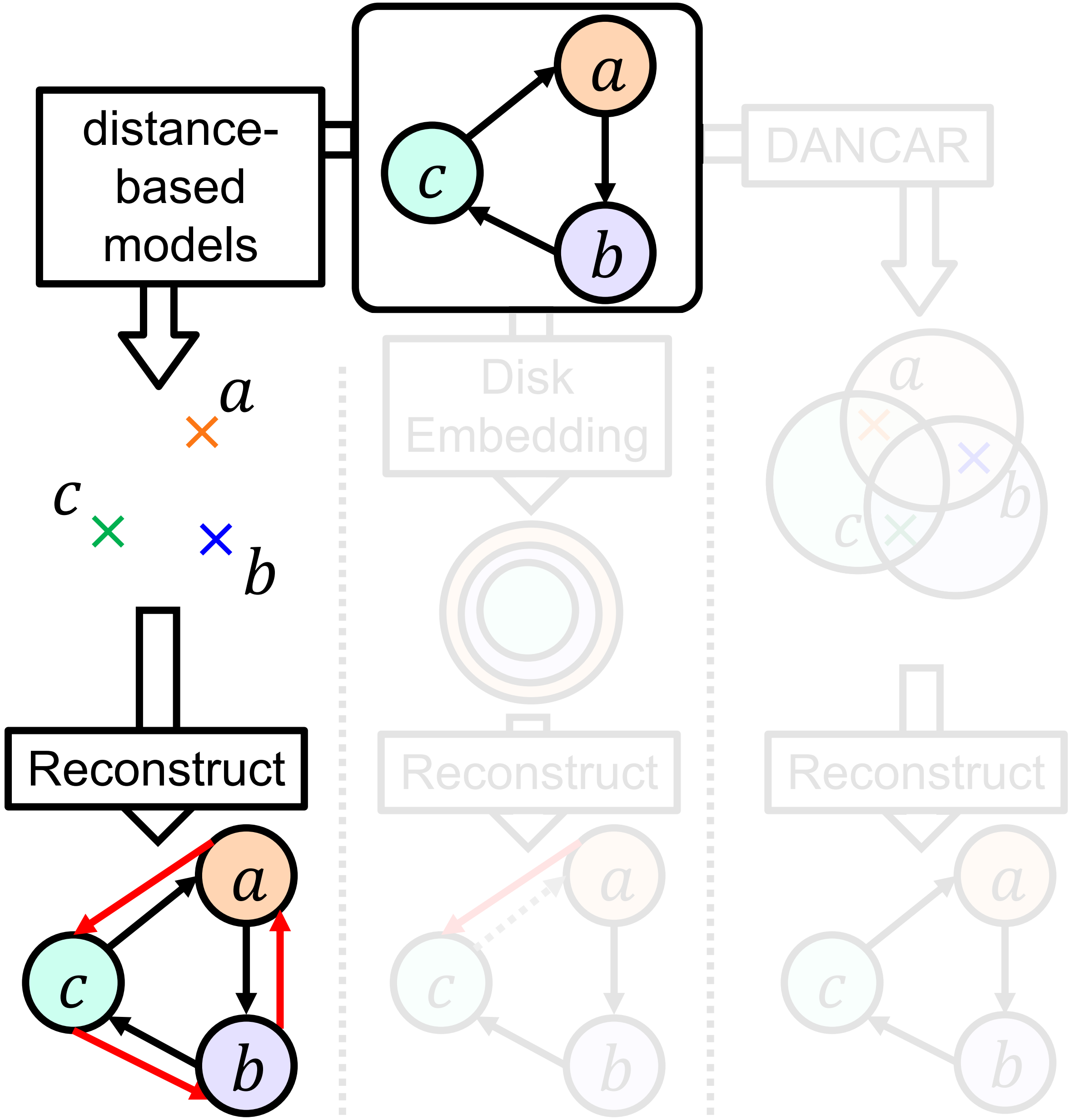
3.2. Representability of DANCAR

- Able to represent directed cycle
  - By introducing anchors
  - distance-based models
    →cannot represent direction
  - embedding as disk
    →cannot represent cycle



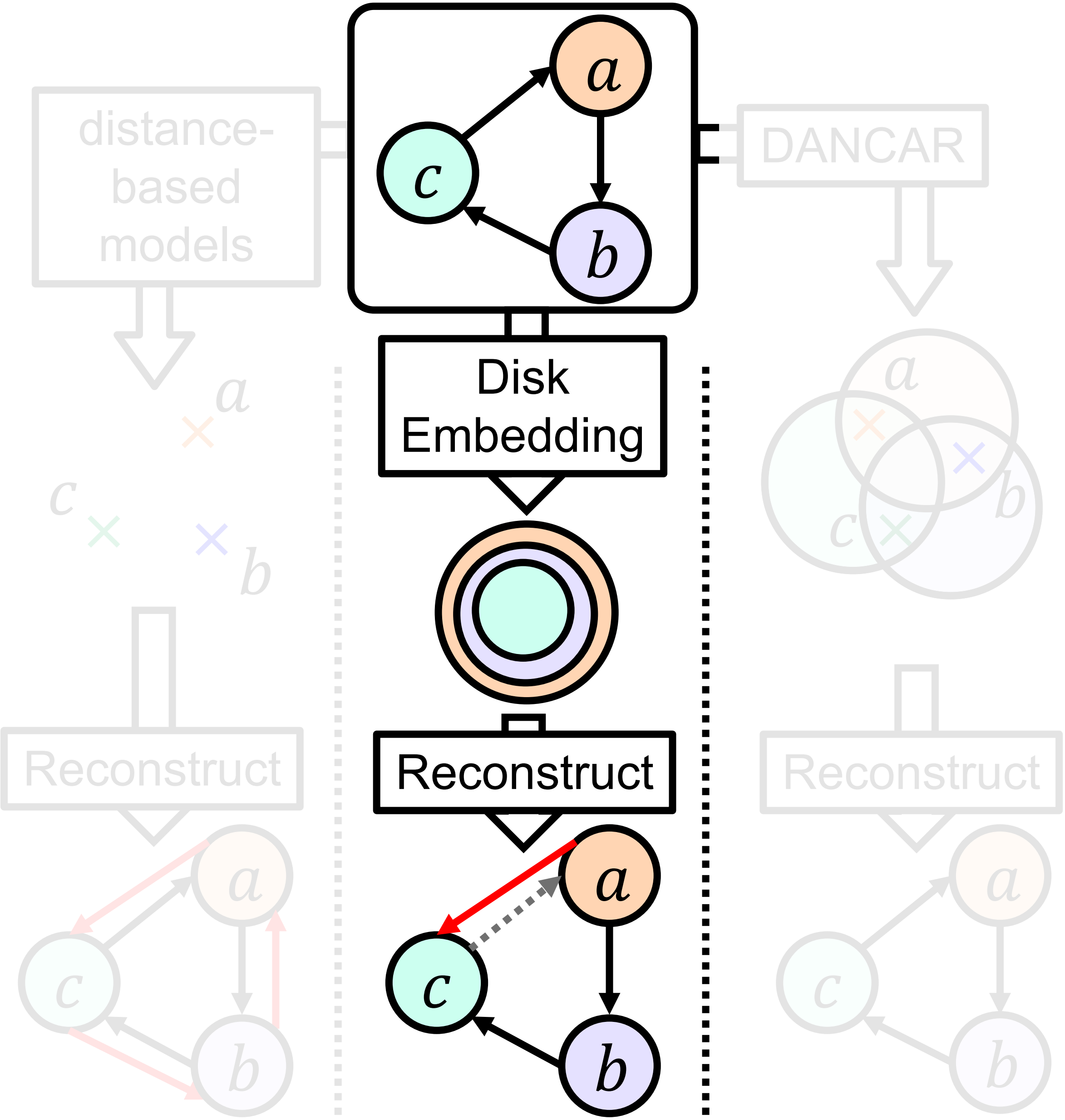
3.2. Representability of DANCAR

- Able to represent directed cycle
  - By introducing anchors
  - distance-based models
    →cannot represent direction
  - embedding as disk
    →cannot represent cycle



3.2. Representability of DANCAR

- Able to represent directed cycle
  - By introducing anchors
  - distance-based models
    →cannot represent direction
  - embedding as disk
    →cannot represent cycle



3.2. Representability of DANCAR

- Able to represent directed cycle
  - By introducing anchors
  - Unlike embedding as points

- Able to represent directed tree
  - By optimizing radii, position

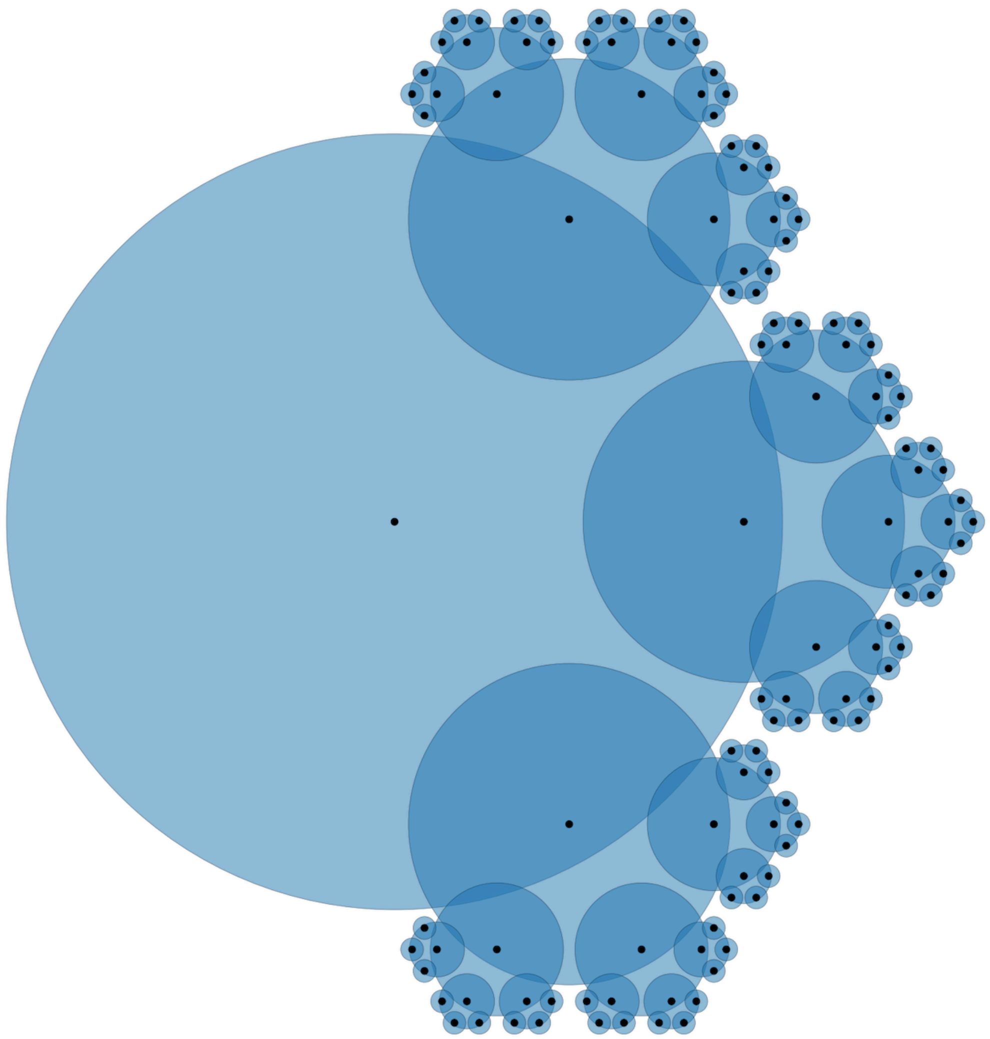
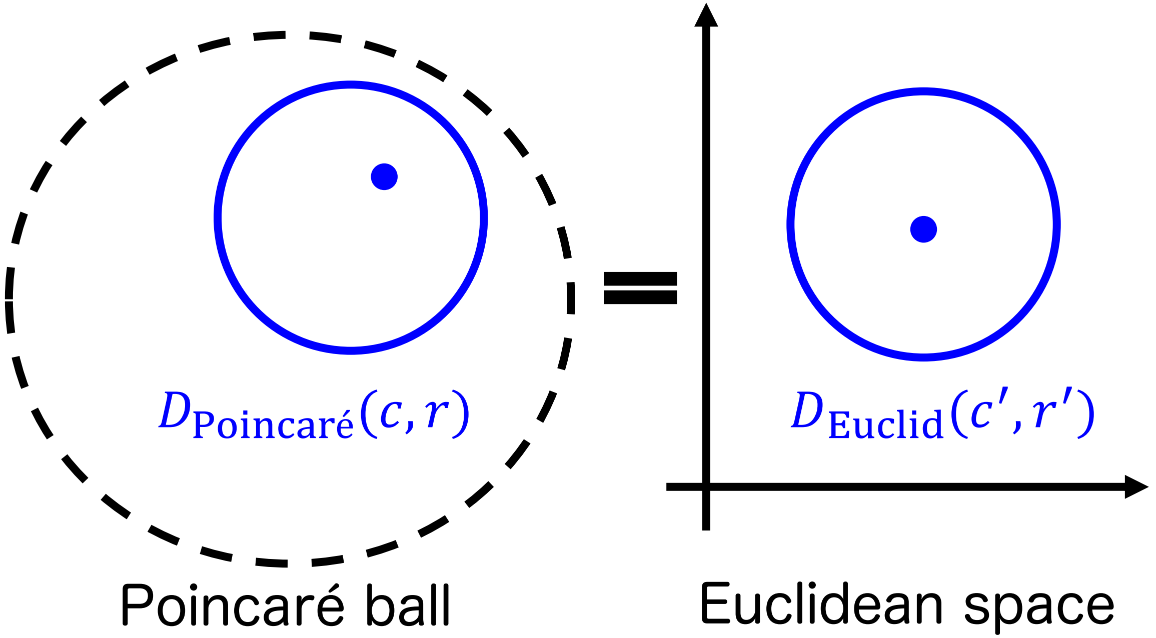


Fig 6. Embedded result of ternary tree with depth=5.

3.2. Representability of DANCAR

- Able to represent directed cycle
  - By introducing anchors
  - Unlike embedding as points
- Able to represent directed tree
  - By optimizing radii
- Generalize Poincaré embedding
  - Ball in Poincaré ball is also ball in Euclidean space
  - Center is different



$$D(c, r) = \{x \in \mathbb{R}^d \mid \|x - c'\| \leq r'\}$$

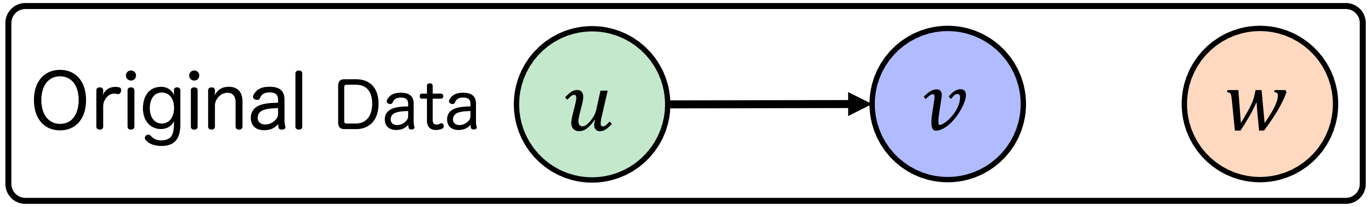
where

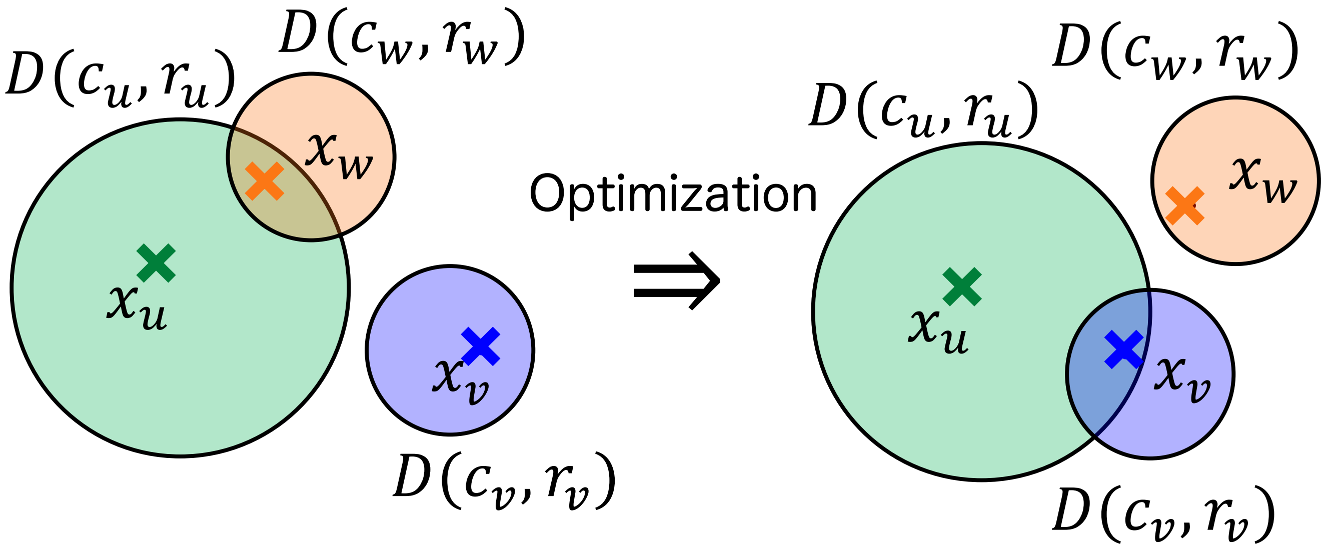
$$c' := \frac{1}{K+1} c, \qquad r' := \sqrt{\frac{K}{K+1}\left(1 - \frac{1}{K+1}\|c\|^2\right)},$$

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

3.3. Loss function

- Positive loss : if $(u, v) \in E$,
  x_v should be in $D(c_u, r_u)$
  $$L_{\mathrm{pos}} \coloneqq \frac{1}{|E|} \sum_{(u,v)\in E} \mathrm{ReLU}(d(c_u, x_v) - r_v + \mu)$$
- Negative loss : if $(u, w) \notin E$,
  x_w should not be in $D(c_u, r_u)$
  $$L_{\mathrm{neg}} \coloneqq \frac{2}{|V|(|V|-1)} \sum_{(u,w)\notin E} \mathrm{ReLU}(r_w - d(c_u, x_w) + \mu)$$
- Anchor loss : for $v \in V$,
  x_v should be in $D(c_v, r_v)$
  $$L_{\mathrm{anc}} \coloneqq \frac{1}{|V|} \sum_{v\in V} \mathrm{ReLU}(d(c_v, x_v) - r_v + \mu)$$



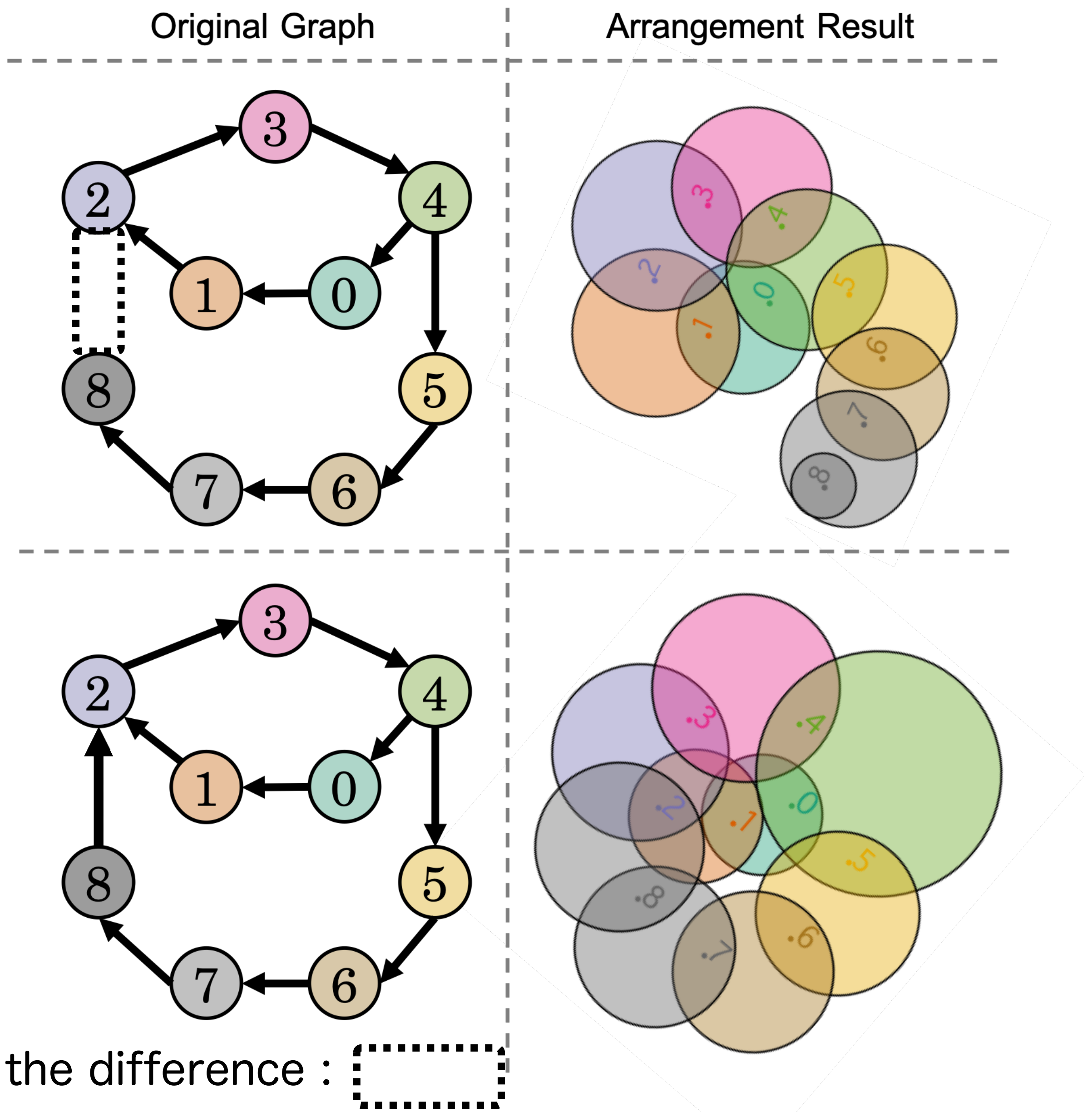


$$L_{\mathrm{DANCAR}} \coloneqq L_{\mathrm{pos}} + \lambda_{\mathrm{neg}} L_{\mathrm{neg}} + \lambda_{\mathrm{anc}} L_{\mathrm{anc}}$$

Contents

4. Experiments : Visualization

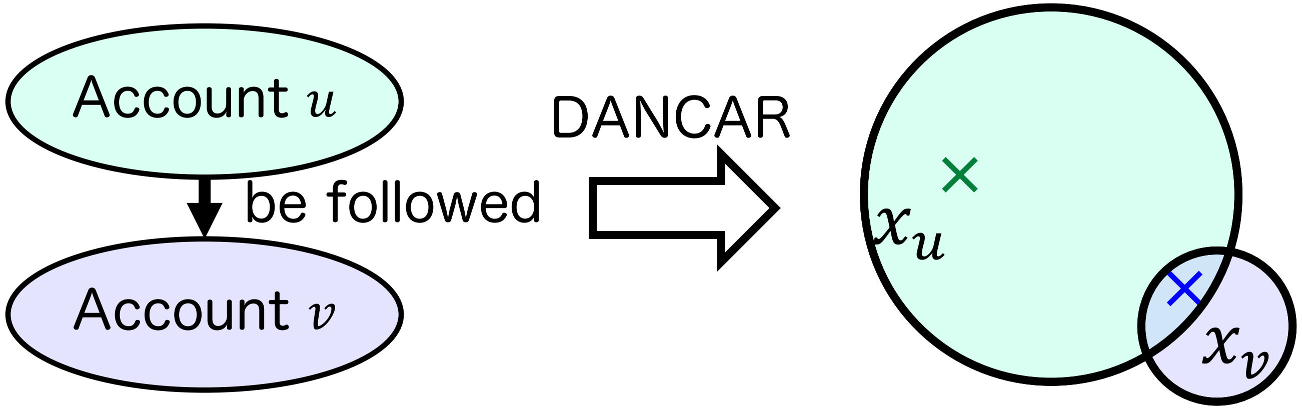
- Sensitive to topology
  - ex) existence of cycles



4. Experiments : Visualization

- Sensitive to topology
  - ex) existence of cycles

Twitter network :



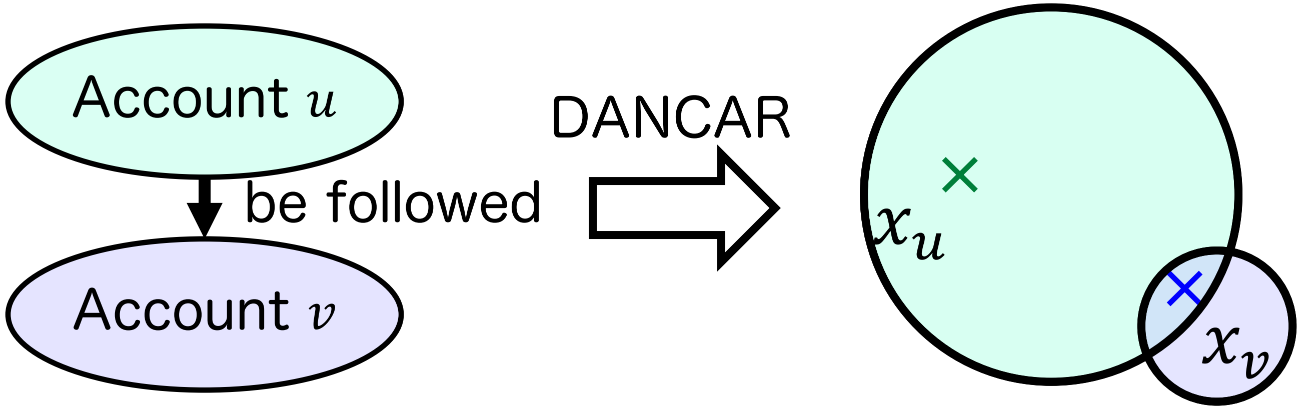
4. Experiments : Visualization

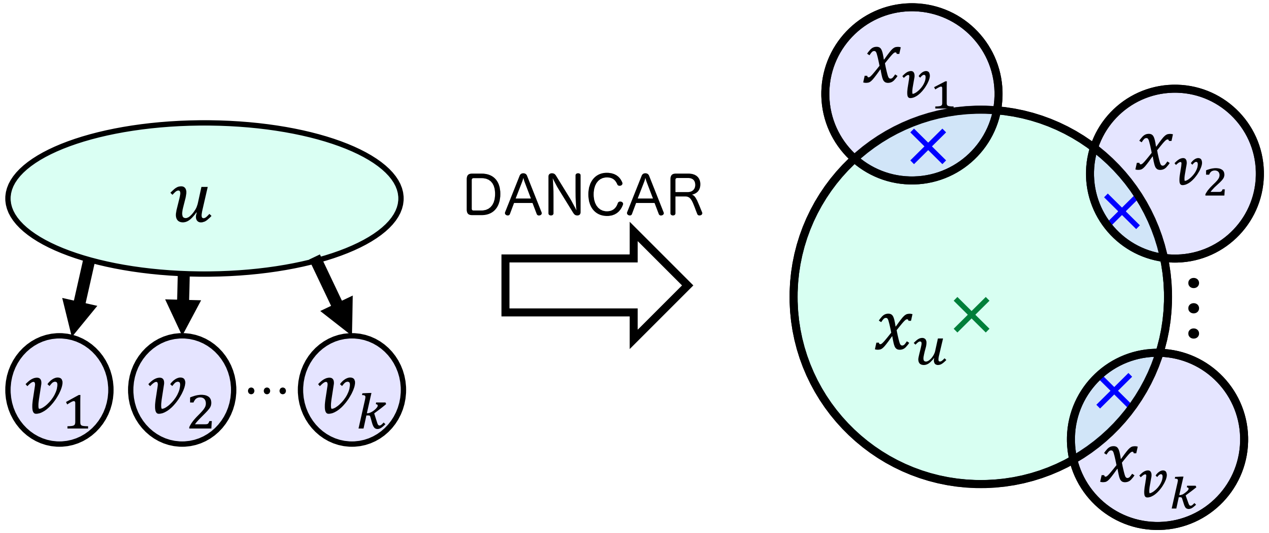
- Sensitive to topology
  - ex) existence of cycles

Twitter network :

→ **Cluster structure**

→ **Hierarchical structure**





4. Experiments : Visualization

- Sensitive to topology
  - ex) existence of cycles

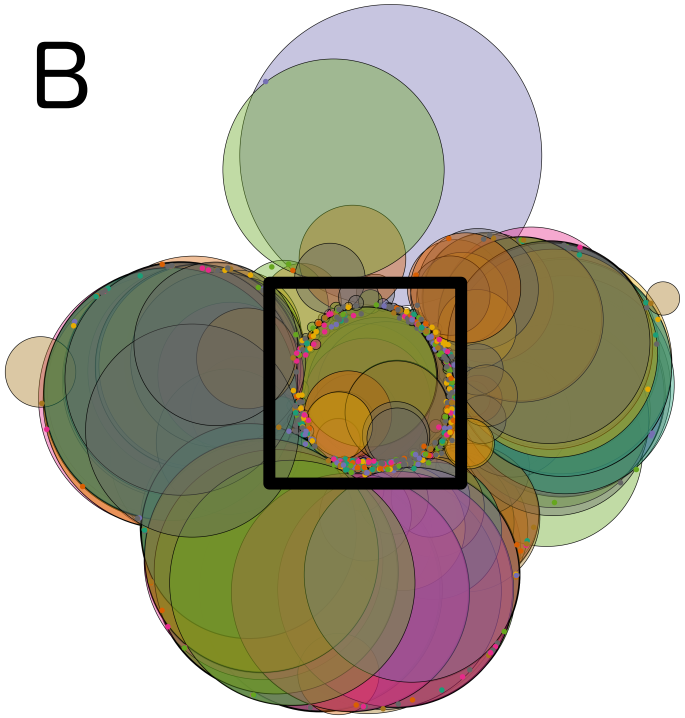
Twitter network :

- Most anchors aggregate in black square

  → **Cluster structure**
- Account v_{out}, with the highest followers, is in the black square
- Most followers of v_{out} have small radii

  → **Hierarchical structure**

Fig 10. Twitter network. dot = anchor, circle = disk.



4. Experiments : Visualization

- Sensitive to topology
  - ex) existence of cycles

Twitter network :

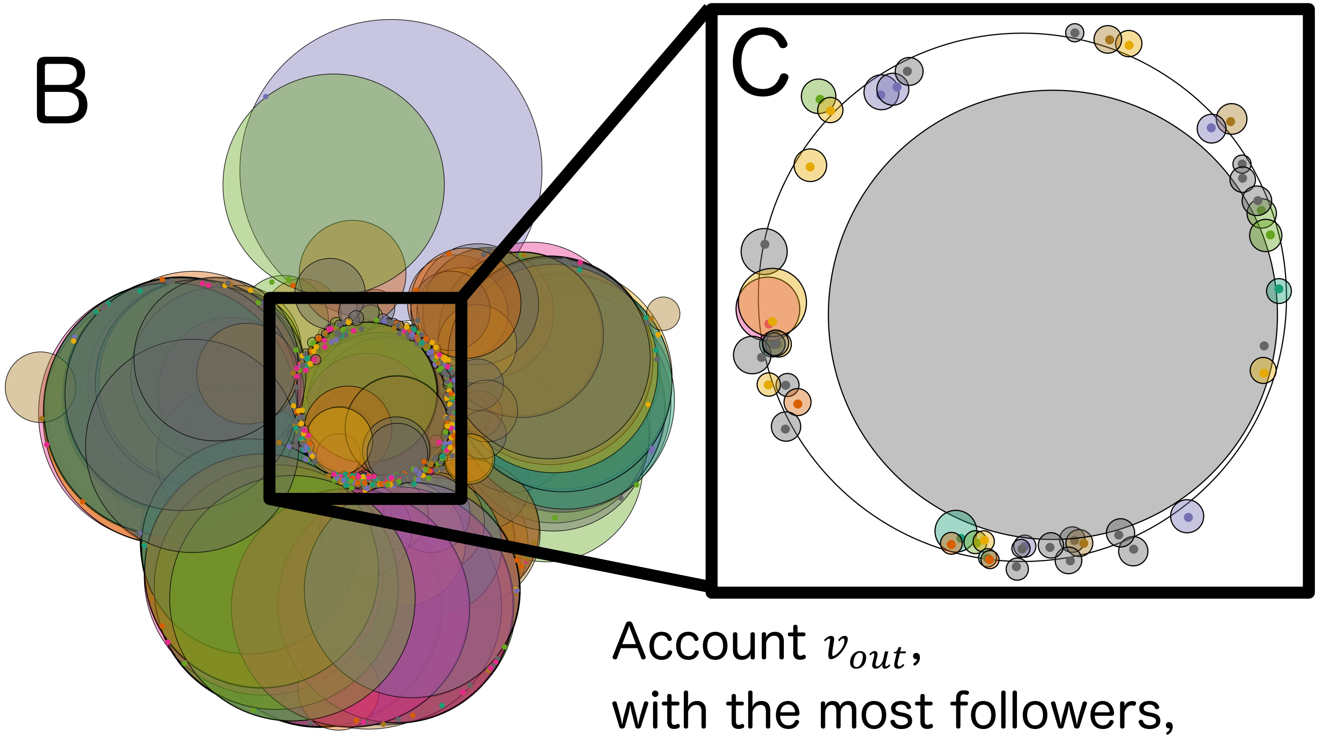
- Most anchors aggregate in black square

  → **Cluster structure**

- Most followers of v_{out} have small radii

  → **Hierarchical structure**

Fig 10. Twitter network. dot = anchor, circle = disk.

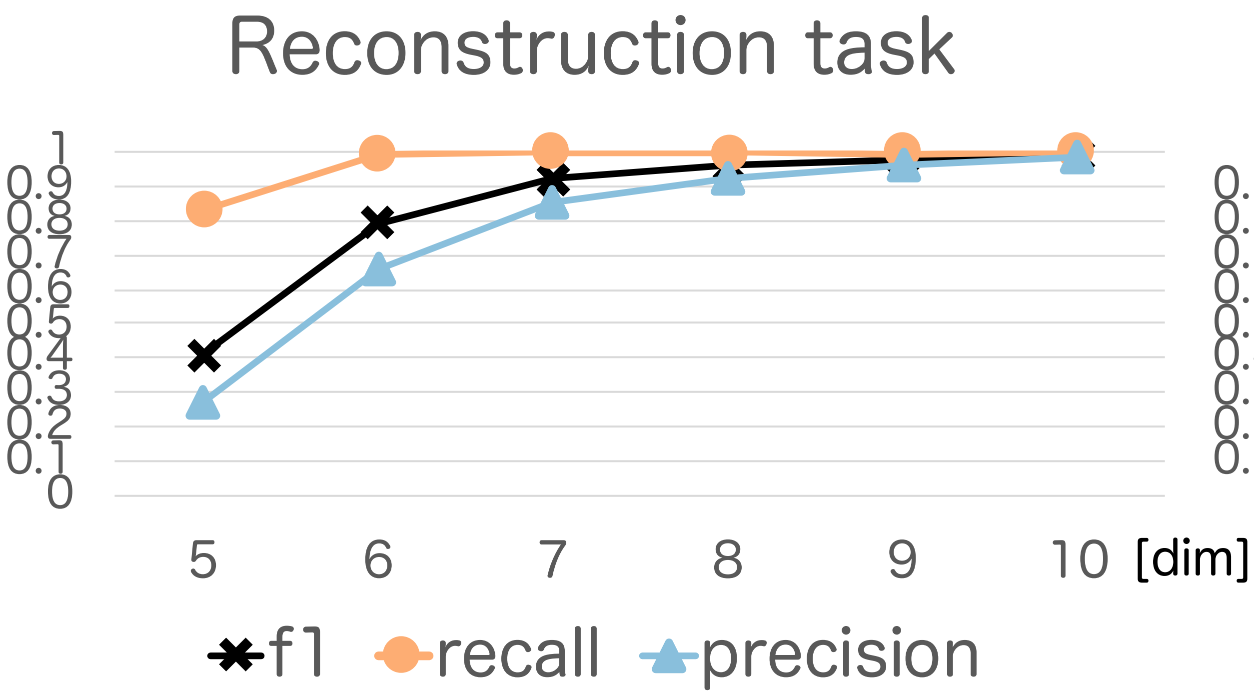


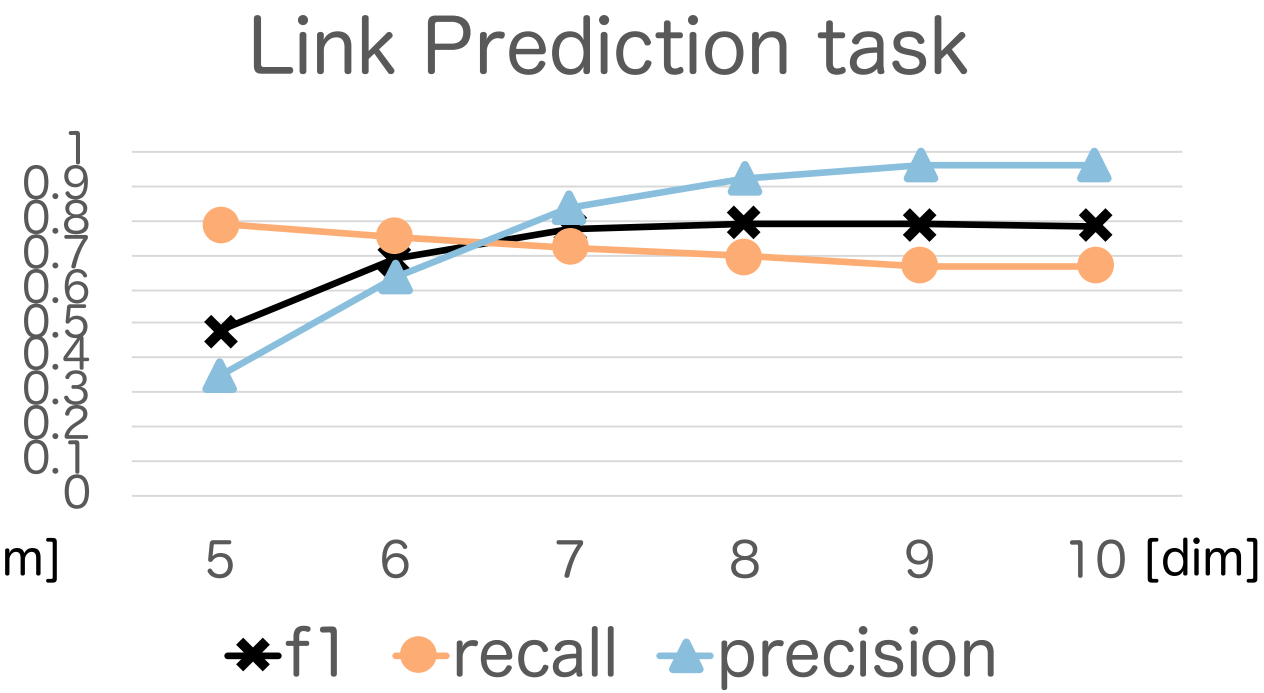
Account v_{out},
with the most followers,
is in the black square

- largest disk : disk of v_{out}
- other disks : follower of v_{out}

Contents

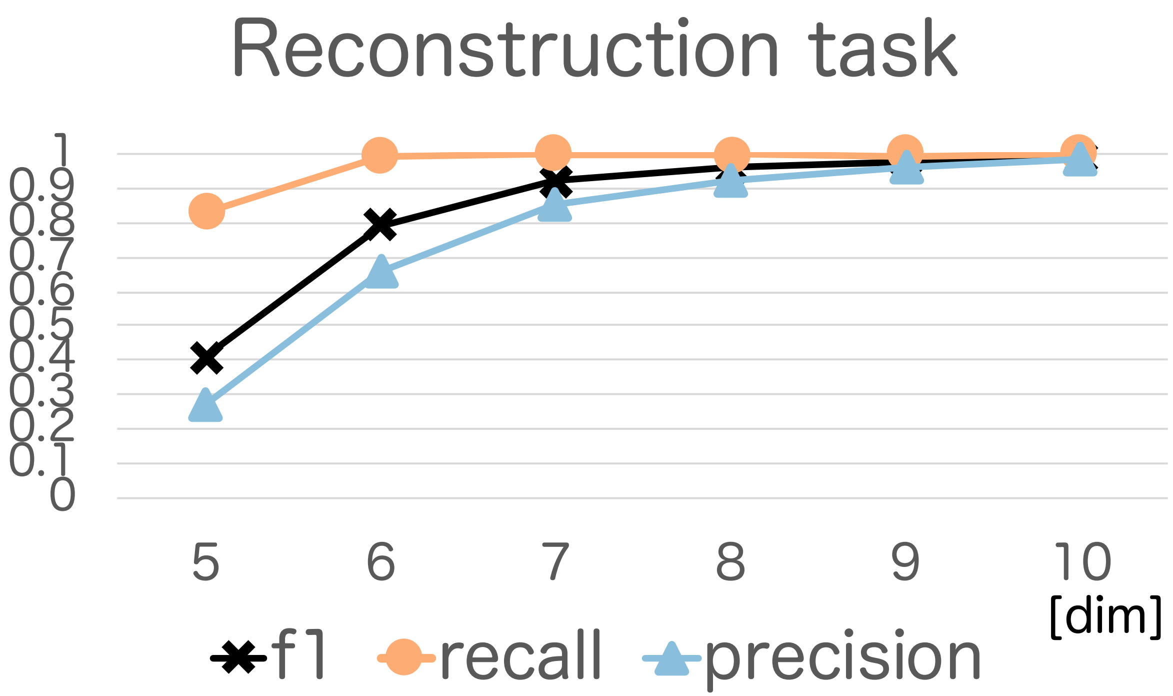
5. Experiments : Reconstruction & Link Prediction





- Dataset : WordNet
  - Removed root and then took transitive closure
- Reconstruction (training with 100% data) & Link prediction task (50%) for dim=5,⋯,10
  - Achieved almost perfect result in reconstruction task

5. Experiments : Reconstruction & Link Prediction



(some results are updated in camera ready version)

	method	100% training				50% training			
		10dim		20dim		10dim		20dim	
		F1 score	mAP	F1 score	mAP	F1 score	mAP	F1 score	mAP
WordNet	**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	-

- Outperforms existing methods with reconstruction task
  - Absence of root affects result
  - (With the root, disk embedding also achieved quite good result)

Contents

6. Conclusion

We propose

- Nested SubSpace (NSS) arrangement: a general framework for representation of relational data in continuous space

- Disk-ANChor ARrangement (DANCAR): to represent directed graphs

The DANCAR can be used for :

- Visualization of a large-scale network to reveal both cluster structure and hierarchical structure

- Representation of a directed graph accurately in terms of the edge reconstruction task

a python implementation is available at https://github.com/KyushuUniversityMathematics/DANCAR

References

1. Nickel, M. and Kiela, D., Poincaré embeddings for learning hierarchical representations. In Advances in Neural Information Processing Systems 30. 2017.

2. Bordes, A., et al., Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pp. 2787–2795, 2013.

3. Vendrov, I., et al., 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.

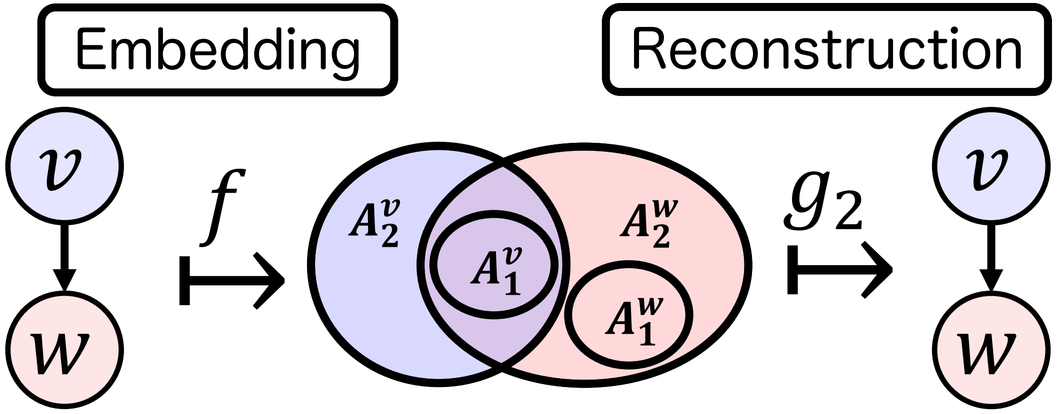
4. Suzuki, R., et al., Hyperbolic disk embeddings for directed acyclic graphs. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6066–6075, Long Beach, California, USA, 09– 15 Jun 2019. PMLR.

Supplementary notes : reconstruction task

$\Phi_m(L)$: the set of ϕ_* on a discrete set of cardinality m with labels in L
$V = \{v_1, \dots, v_m\}$

1. $f: V \to \mathcal{S}_n(X) \Longrightarrow \big(f(v_1), f(v_2), \dots, f(v_m)\big) \in \mathcal{S}_n(X)^m = \mathcal{S}_n(X)^{|V|}$
2. $g_{|V|}: \mathcal{S}_n(X)^{|V|} \to \Phi_{|V|}(L) \Longrightarrow g_m\big(f(v_1), f(v_2), \dots, f(v_m)\big) \in \Phi_{|V|}(L)$
3. an element of $\Phi_{|V|}(L)$ is a relational structure on V labeled on L

- Embedding
  = map $f: V \to \mathcal{S}_n(X)$
- Reconstruction
  = collection of maps
  $g_m: \mathcal{S}_n(X)^m \to \Phi_m(L)$

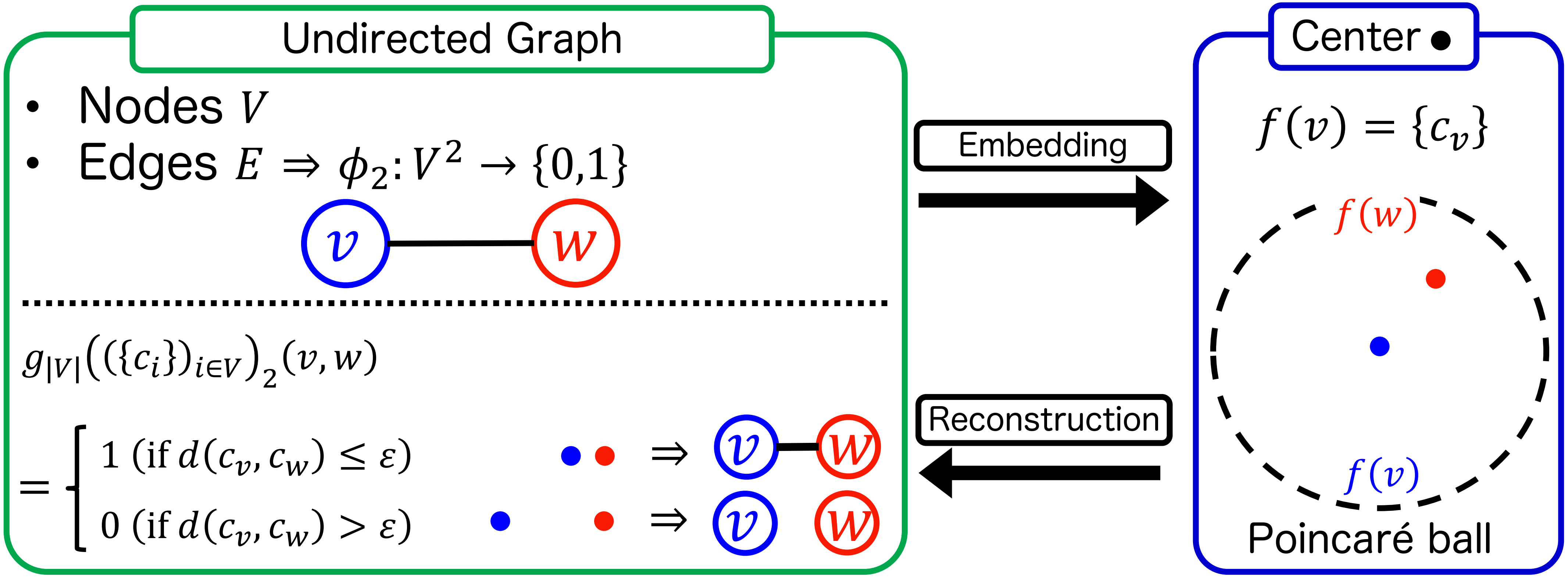


Reconstruction task
= find map f s.t.
$$g_{|V|}\big(f(V)\big) = \phi_*$$

Supplementary notes : Poincaré embedding

Q. In 2.A :
Can we see Poincaré embedding as the NSS arrangement with depth 1?

A. Yes. In that case, f, g can be defined as follows. To prove that the DANCAR generalizes the Poincaré embedding, we converted it to the NSS with depth 2.



Supplementary notes : Existence of Root

Q. In experiments:
Does the existence of the root affect the result?

A. Experimentally, Yes. The table is the result of the numerical experiments with :

- Transitive closure of WordNet
- embedding dim = 5
- reconstruction (training edges = 100%) and link prediction (training edges = 50%)
- evaluated by f1 score

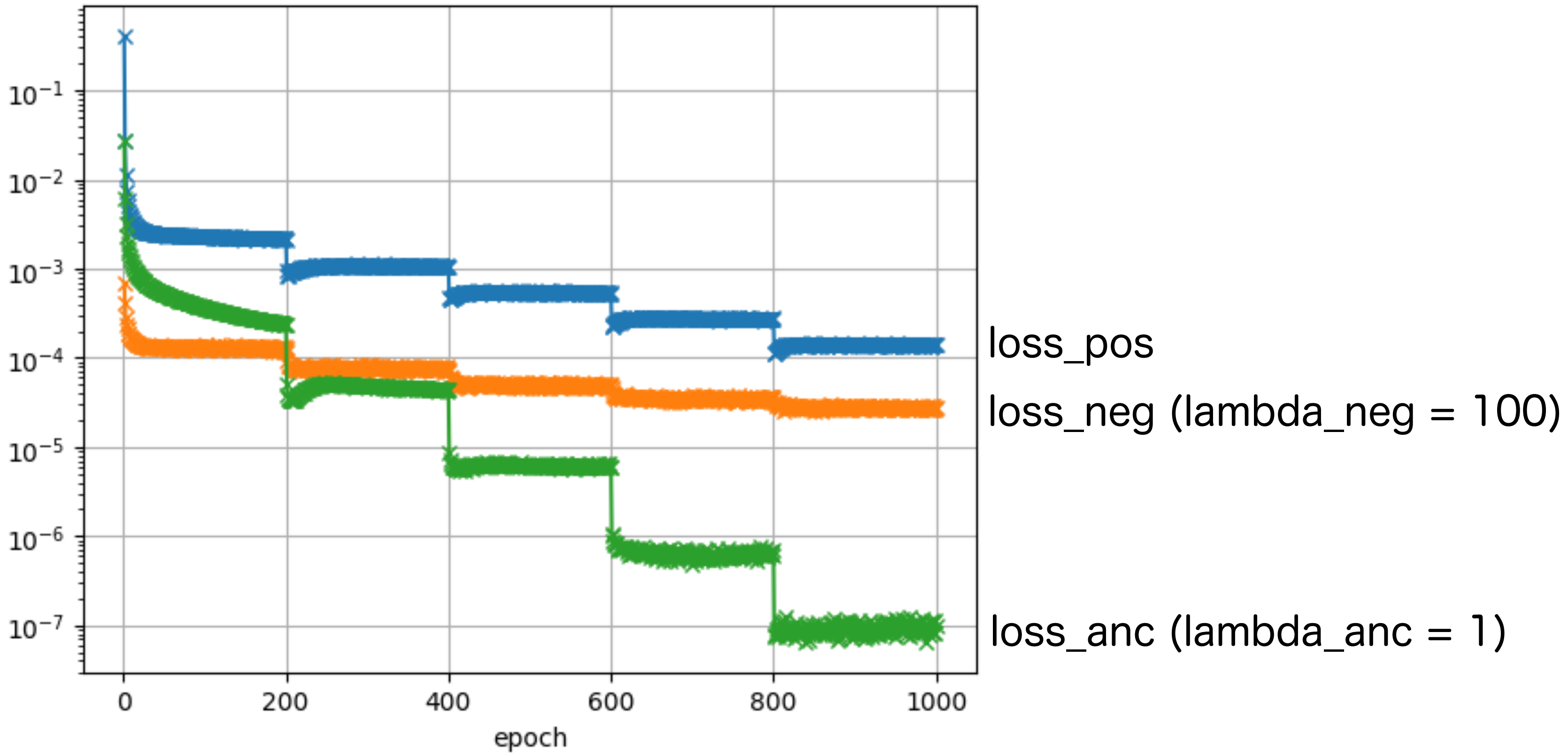
The absence of root node does deteriorate the performance of Disk Embedding whereas our DANCAR is not affected a lot.

With root	Reconstruction	Link Prediction
DANCAR	0.74208	0.72147
Disk Embedding	0.83188	0.77019

Supplementary notes : Calculation Time

Q. How is the transition of the loss function during optimization?
How long did it take to optimize the arrangement?

A. Here is the transition of loss function (DANCAR, 10 dim, took 29 hours).
In the experiment, We halved the parameter α in Adam per 200 epochs.



Supplementary notes : Reconstruction Rule of Figure 4

Q. What is the reconstruction rule of Figure 4?

A. In this example, the following rules are used:

- Each node v is assigned to an NSS with depth 2: $A_1^v \subset A_2^v$.

1. If and only if $A_1^w \subset A_2^v$, an edge (v, w) is present.
2. If and only if $A_1^w \subset A_2^v$ and $A_2^w \subset A_2^u$, a directed hyperedge $(u, (v, w))$ is present.

In the right side of Figure 4, directed hyperedge $(A, (B, C))$ is present by the second rule.

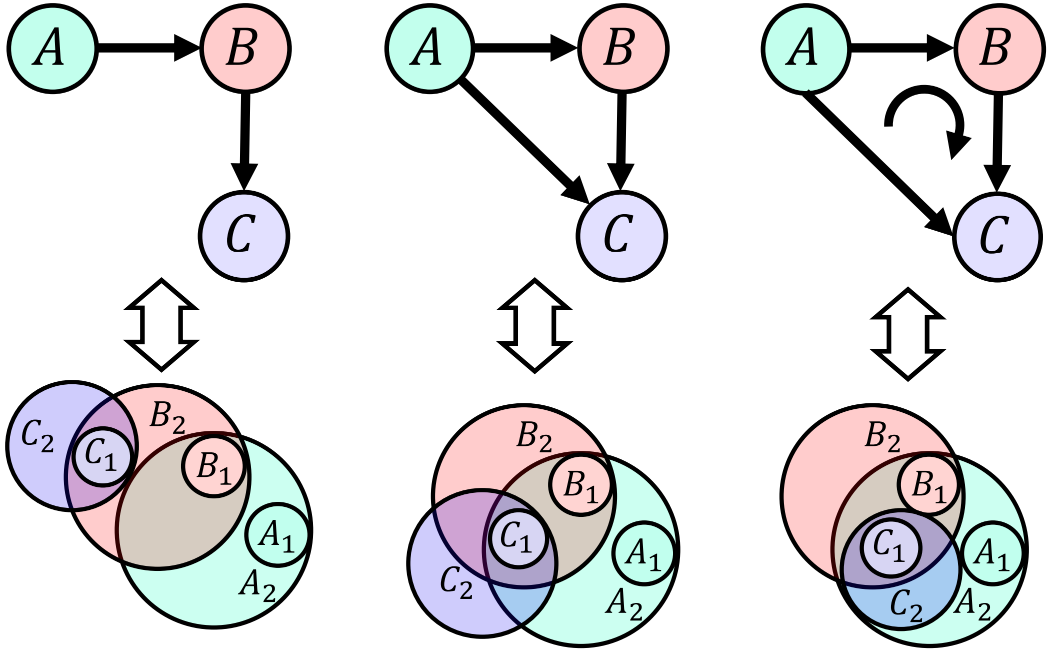


Figure 4