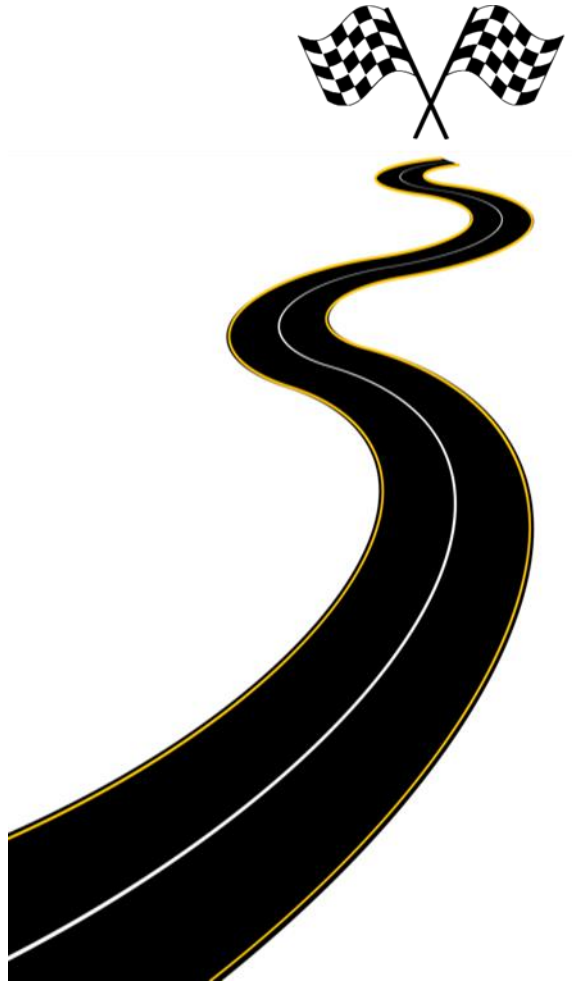
Datasets, Tasks, and Training Methods for Large-scale Hypergraph Learning

Sunwoo Kim*, Dongjin Lee*, Yul Kim,

Jungho Park, Taeho Hwang, Kijung Shin

Roadmap

- Overview
- Proposed Tasks
- Proposed Datasets
- Proposed Training Method
- Experimental Results
- Conclusions



Group Interactions are EVERYWHERE!

Group Chat

How Transitive Are Real-World Group Interactions? - Measurement and Reproduction

Sunwoo Kim
KAIST
kswoo97@kaist.ac.kr

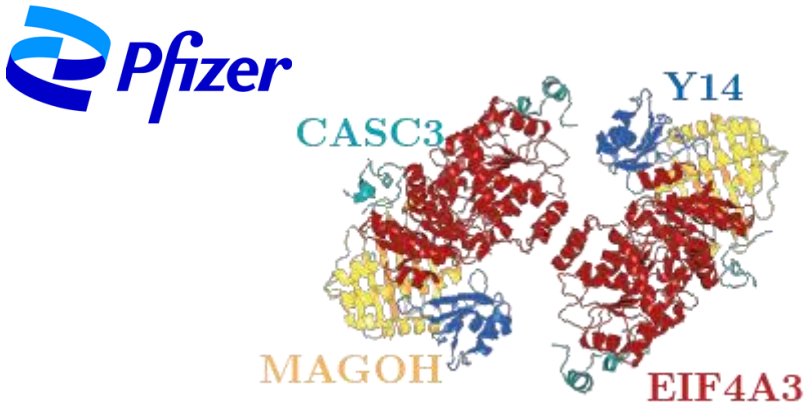
Fanchen Bu
KAIST
boqvezen97@kaist.ac.kr

Minyoung Choe
KAIST
minyoung.choe@kaist.ac.kr

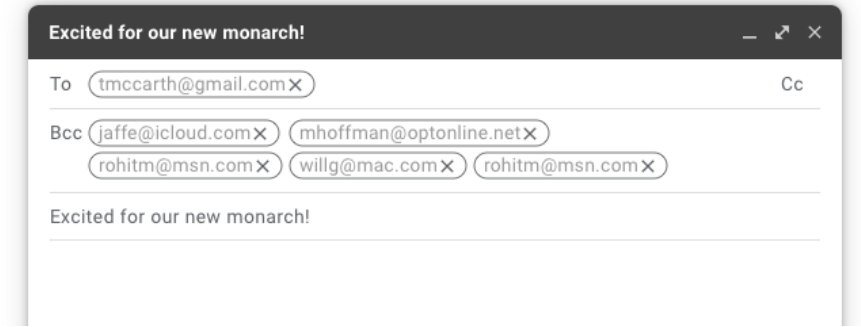
Jaemin Yoo
Carnegie Mellon University
jaeminyoo@cmu.edu

Kijung Shin
KAIST
kijungs@kaist.ac.kr

Co-Authorship



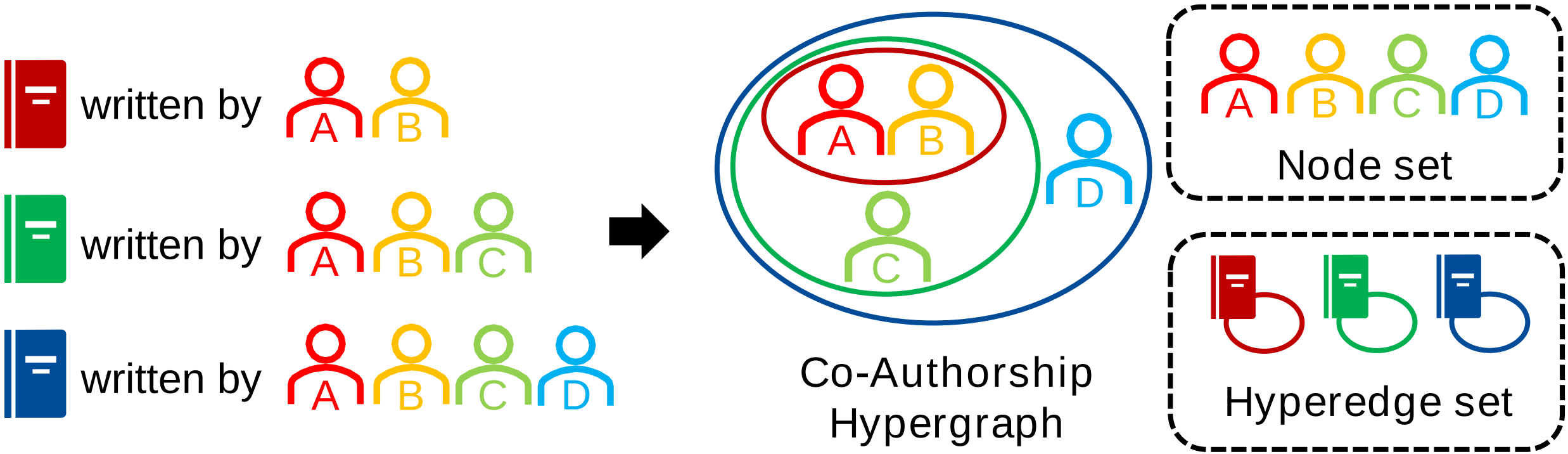
Protein Interaction



Email

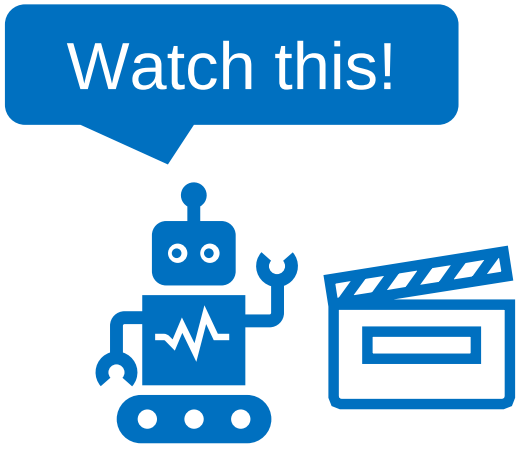
Group Interactions → HYPERGRAPH!

- Group interactions can be modeled as a hypergraph.
- A hypergraph consists of a node set and a hyperedge set.



Applications of Hypergraphs

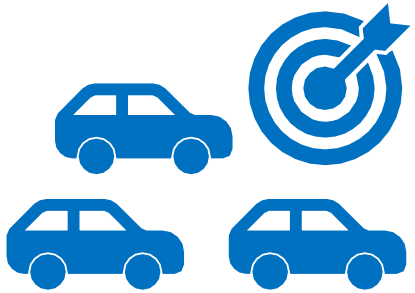
- Hypergraphs are widely used in various domains.



Recommender Systems

Financial Analysis

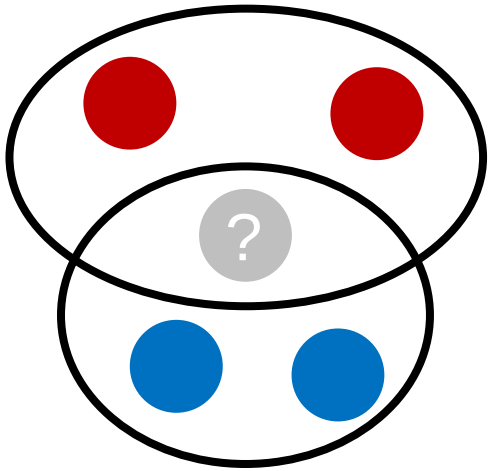
Drug Analysis



Traffic Forecasting

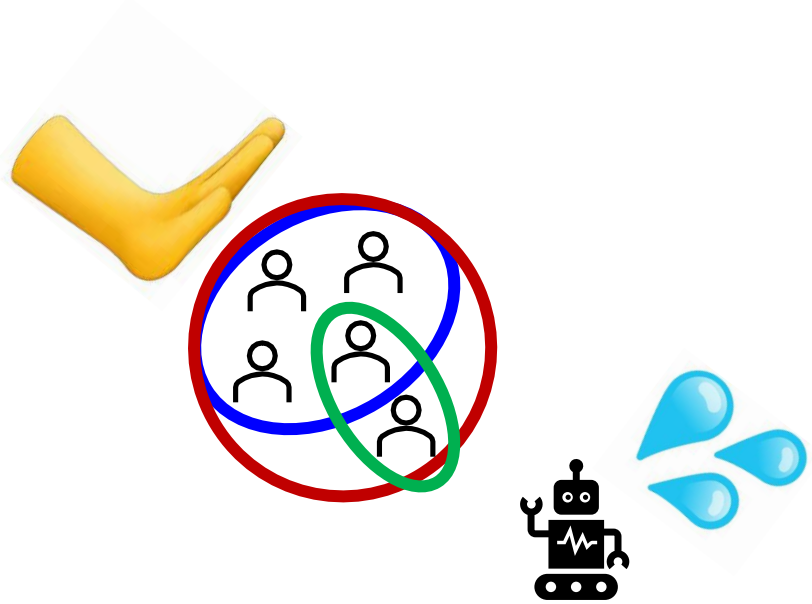
Challenges in Hypergraph Learning

- The current hypergraph representation learning field has three challenges:



Limited downstream task

Small benchmark datasets

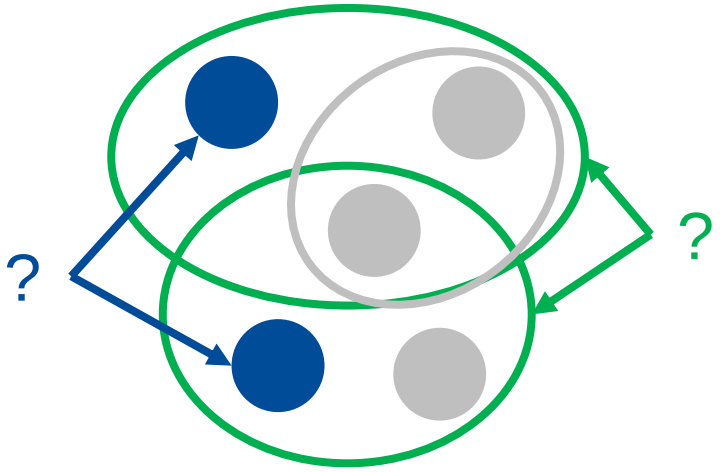


No scalable training strategy

Our Contribution

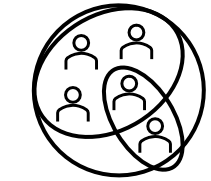
- Our contribution is three-fold.

C1) Task

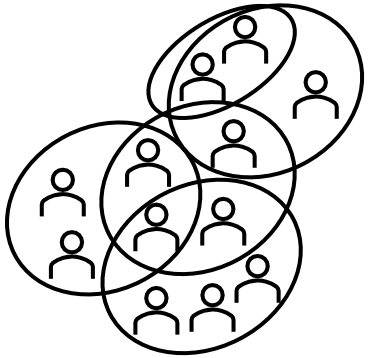


Pair-level tasks

C2) Dataset



Existing

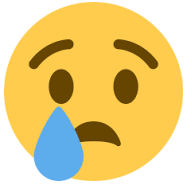


New

Large-scale hypergraph datasets

C3) Training method

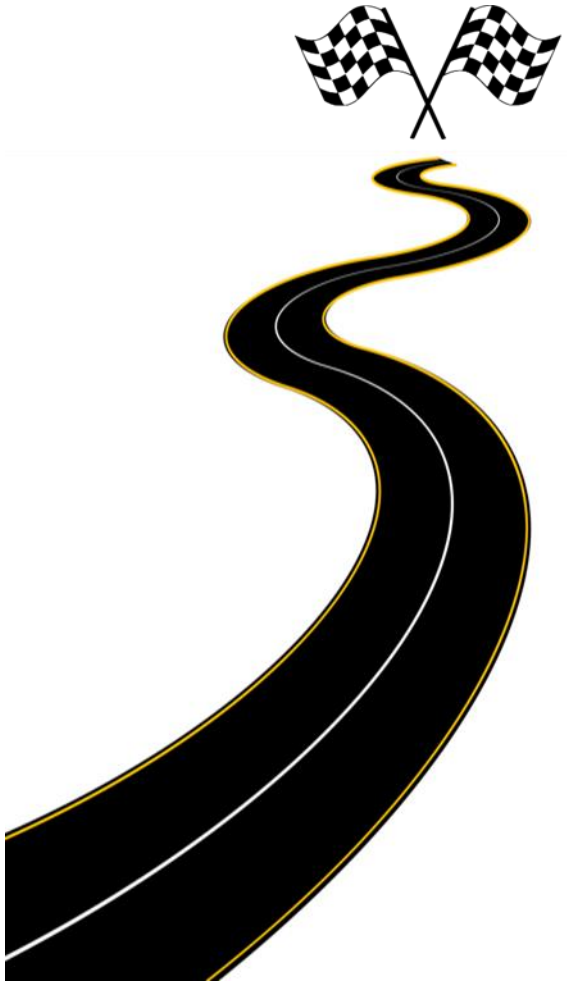
Existing	Ours
O.O.M.	Success!



A scalable training strategy

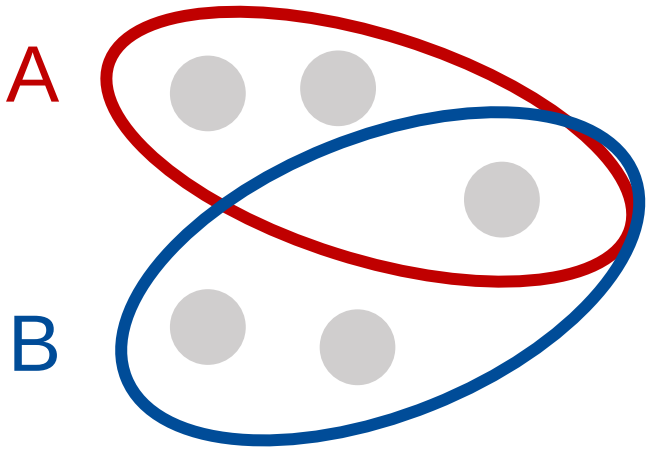
Roadmap

- Overview
- Proposed Tasks
- Proposed Datasets
- Proposed Training Method
- Experimental Results
- Conclusions

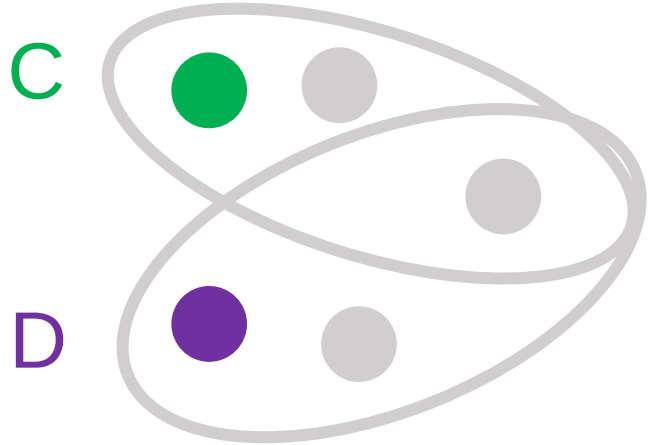


Proposed Tasks

- We propose two pair-level downstream tasks in hypergraphs.
  - Task 1) Hyperedge disambiguation, classifying a pair of hyperedges.
  - Task 2) Local clustering, classifying a pair of nodes.



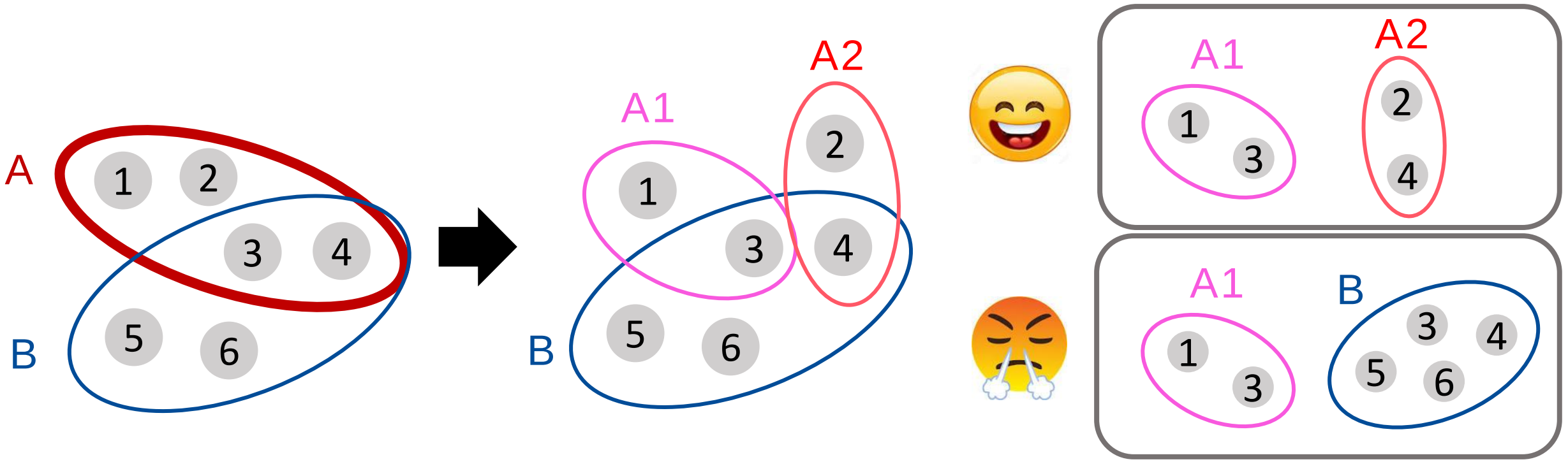
Task 1: A ↔ B?



Task 2: C ↔ D?

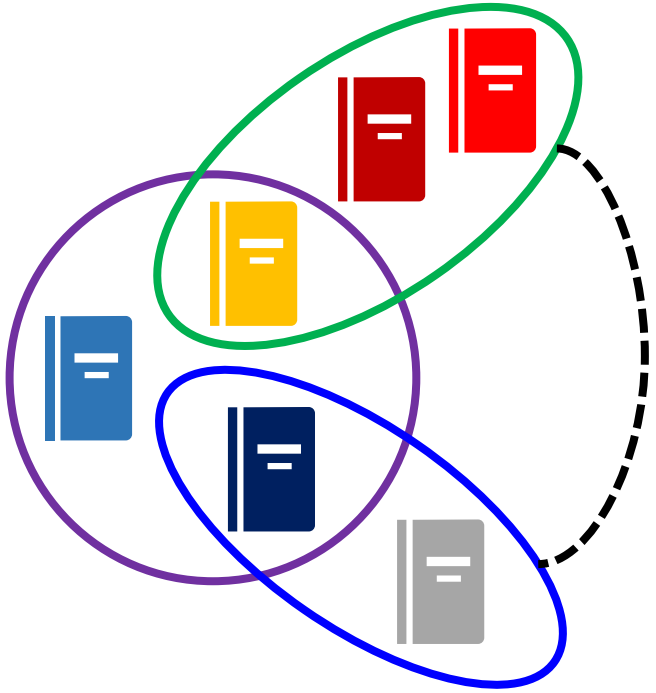
Proposed Task 1: Formulation

- Setting: Certain hyperedges are split into two (or more) hyperedges.
- Goal: Identifying whether a pair of hyperedges are originally identical hyperedges or not.



Proposed Task 1: Application

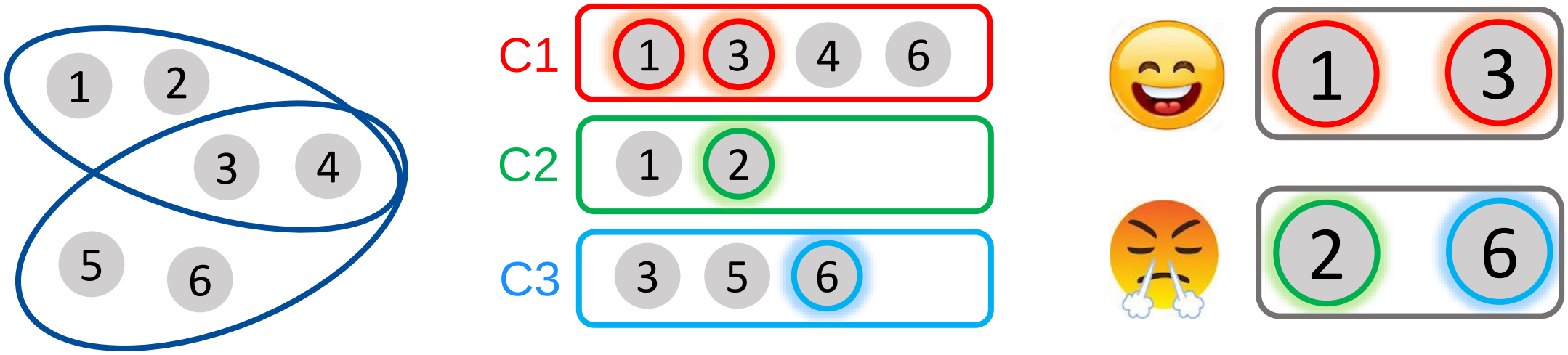
- One application of task 1 is author disambiguation.
  - Goal: Identifying whether two authors are identical authors or not.
  - Hypergraph formulation: Nodes (publications) & Hyperedges (authors)



Are they the same author?

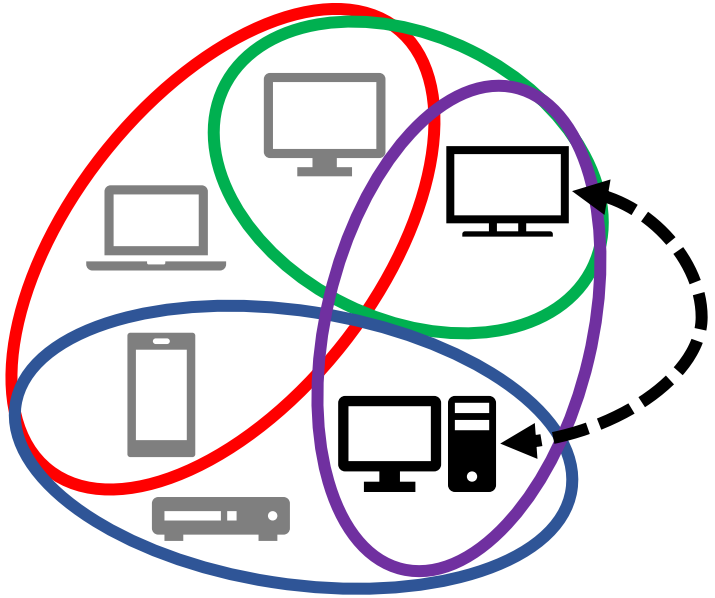
Proposed Task 2: Formulation

- Setting: Each node belongs to at least one cluster.
- Goal: Identifying whether a pair of nodes belong to the same cluster or not.

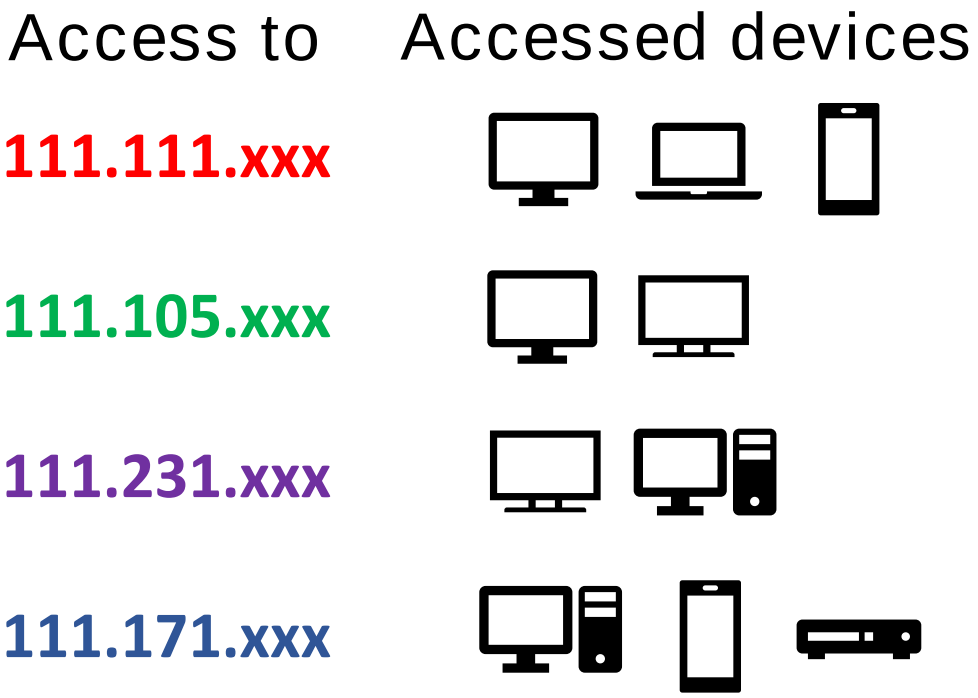


Proposed Task 2: Application

- One application of task 2 is device-household matching.
  - Goal: Identifying whether two devices belong to the same household or not.
  - Hypergraph formulation: Nodes (devices) & Hyperedges (IP - access) & Clusters (households).



Belong to the same household?



Roadmap

- Overview
- Proposed Tasks
- Proposed Datasets
- Proposed Training Method
- Experimental Results
- Conclusions

Proposed Dataset: Overview

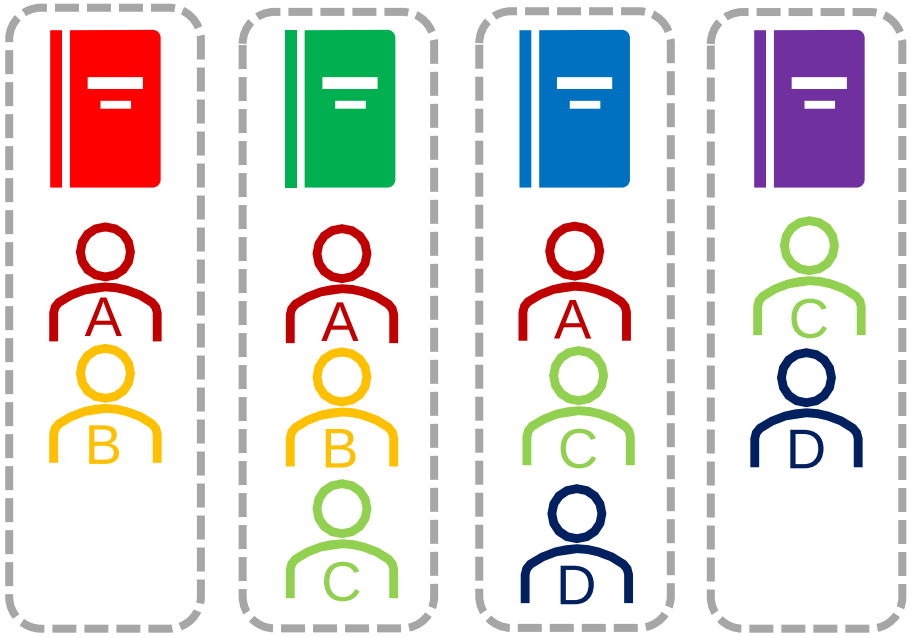
- We propose two large-scale co-authorship hypergraph datasets:

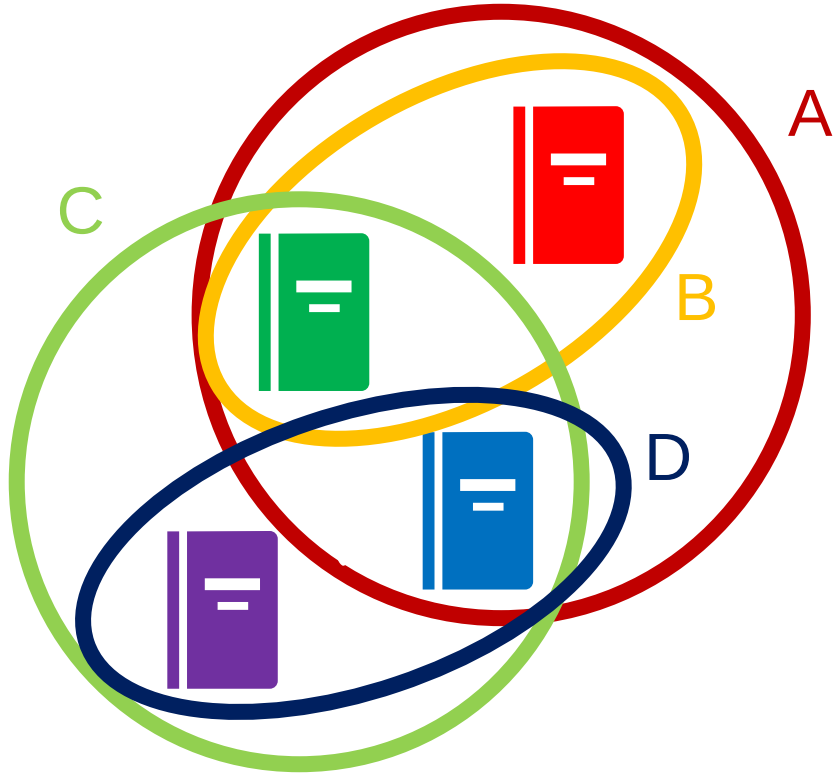
Dataset	# of Nodes	# of Hyperedges	# Features	# of Classes
AMiner	13,262,573	22,552,647	300	257
MAG	27,320,375	30,175,013	300	247

Proposed Dataset: Details

** https://www.aminer.cn/oag-2-1

- Source of datasets: Open Academic Graph 2.1
- Nodes: Publications
- Hyperedges: Authors





Proposed Dataset: Details

** https://www.aminer.cn/oag-2-1

- Node features: An embedding of corpus, which consists of title and keywords, transformed via Sentence2Vec (S2V).
- Node labels: Research field of the corresponding publication.

Title: Temporal patterns of real-world group interactions.

Keywords: Hypergraph, Group interaction, Time series

Field: Data Mining & Analysis

temporal patterns of real-world group interactions hypergraph group interaction time series

$\in \mathbb{R}^{300}$

Node features:

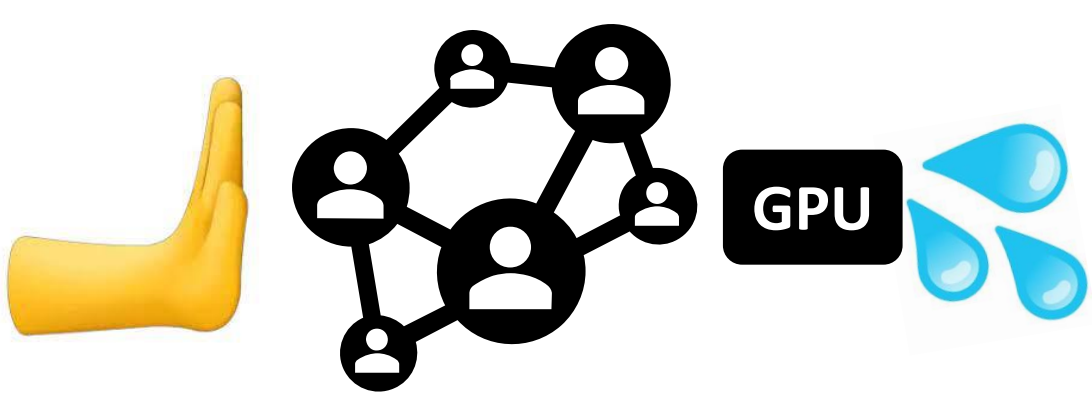
Node label: Data Mining & Analysis

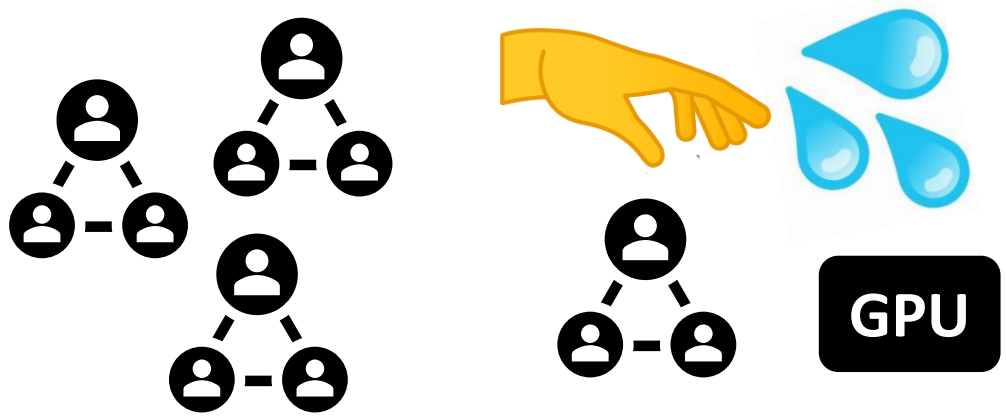
Roadmap

- Overview
- Proposed Tasks
- Proposed Datasets
- Proposed Training Method
- Experimental Results
- Conclusions

Challenges in Training HNNs on Large-Scale HG

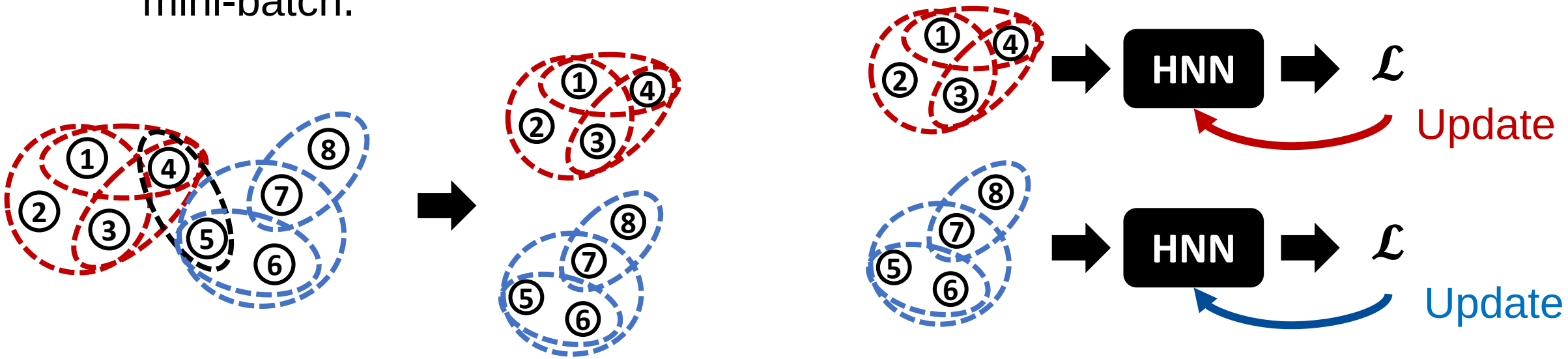
- Challenge 1) Considerable time and space cost.
  - Full-batch training: Can’t load the entire hypergraph into GPU memory.
  - Mini-batch training: Computational overhead occurs in obtaining sub-hypergraphs and loading them into GPU memory.





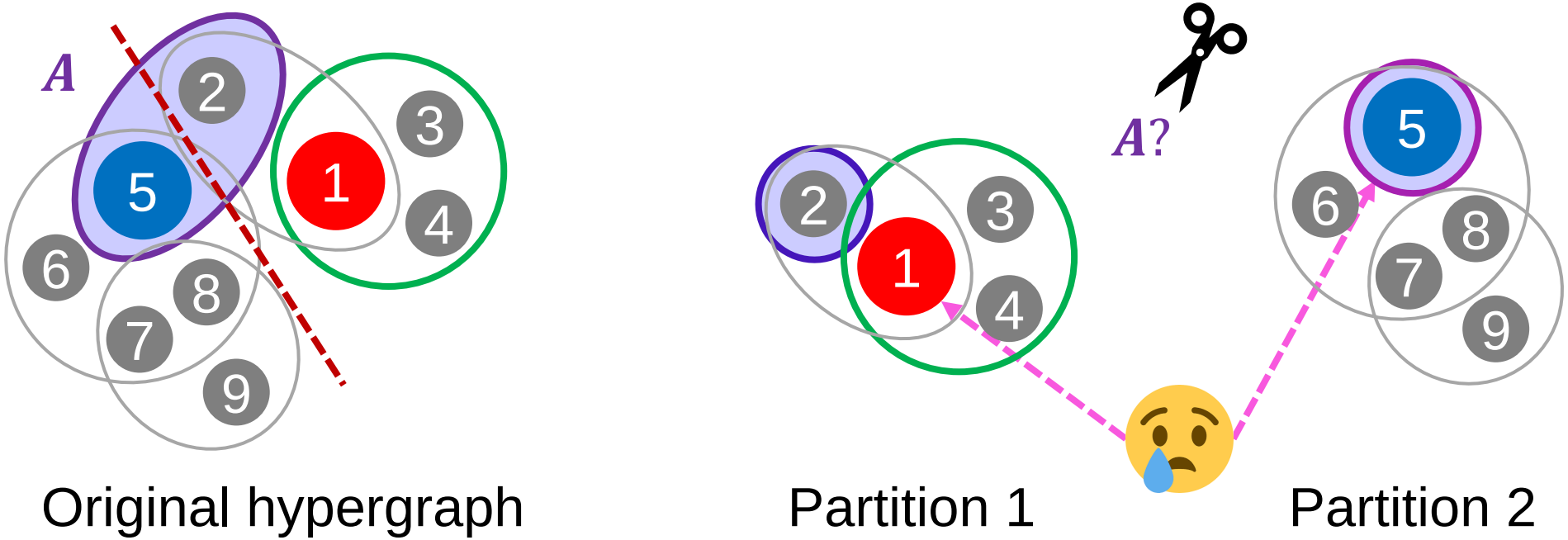
Proposed Learning Method: Solution 1

- Solution 1) Partitioning the input hypergraph.
  - A partitioning strategy is efficient in training graph neural networks (Chiang et al. 2019).
  - We use partitioning, treating each hypergraph partition as a single mini-batch.



Challenges in Training HNNs on Large-Scale HG

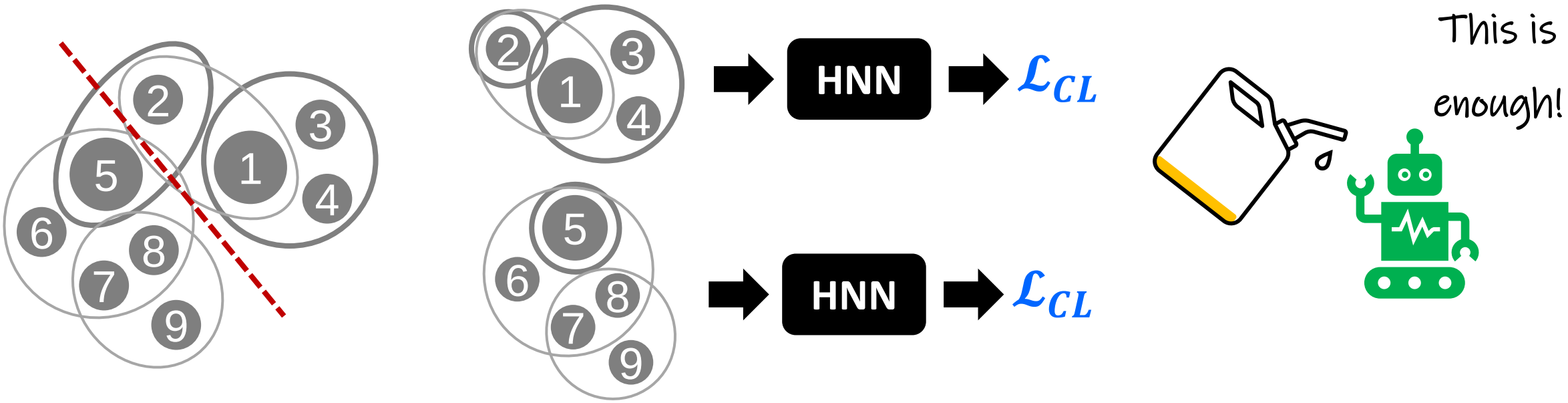
- Challenge 2) Loss of pair-label supervision due to partitioning.
  - When we process a single partition at a time, certain pair-labels cannot be utilized.



Original hypergraph Partition 1 Partition 2

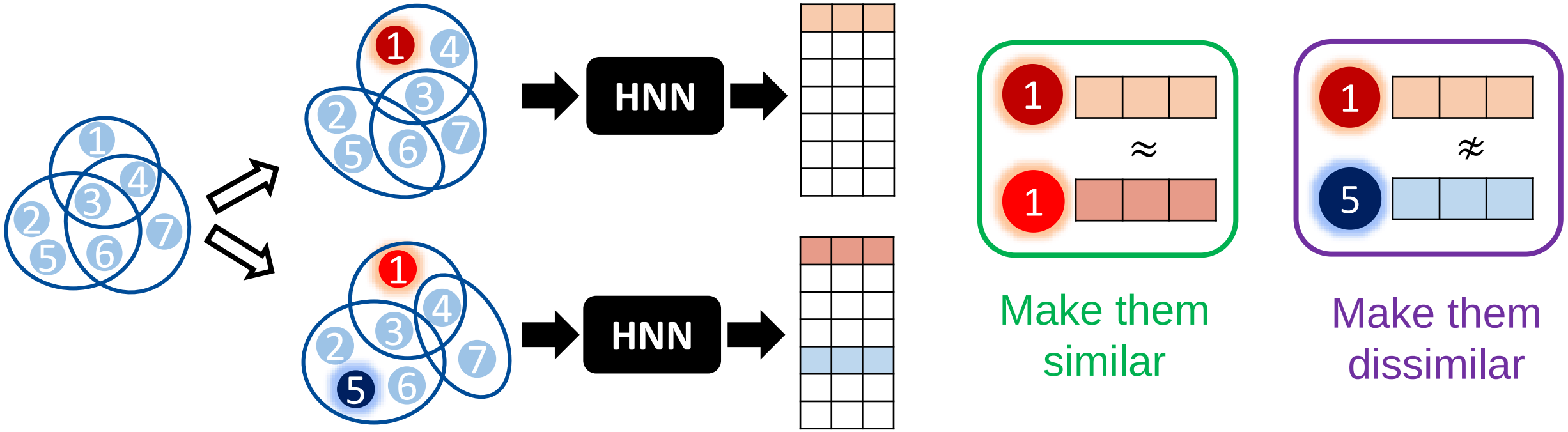
Proposed Learning Method: Solution 2

- Solution 2) Utilizing Contrastive Learning (CL).
  - CL can make effective representations under label-scarce scenarios (Chen et al. ICML 2020, You et al. NeurIPS 2020).



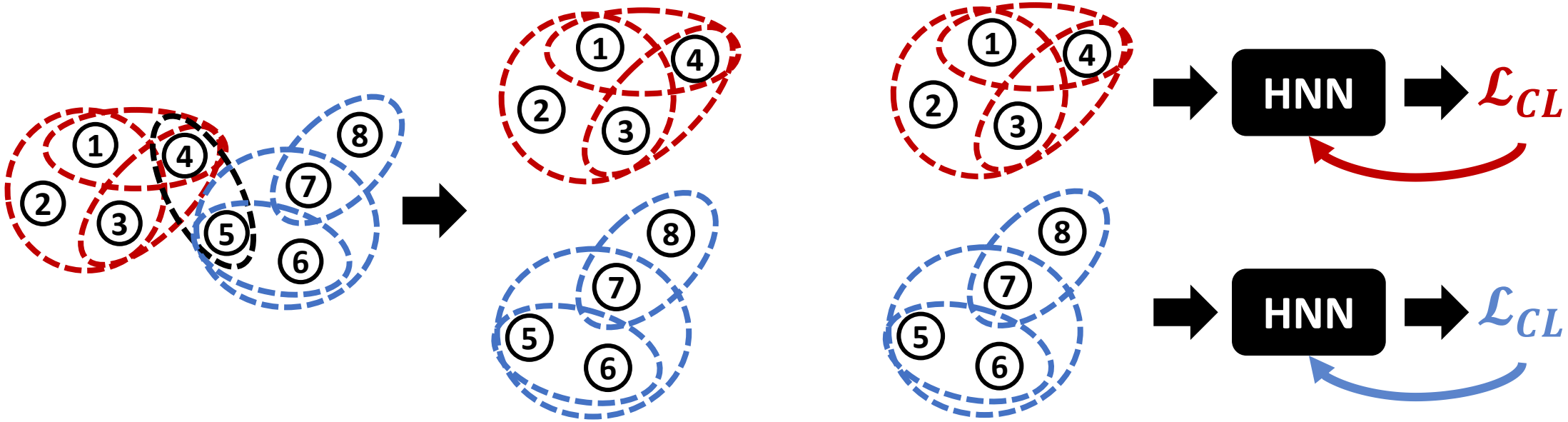
Proposed Learning Method: Solution 2

- Overview of CL is as follows:
  - CL first creates multiple views of a hypergraph.
  - Then, it lets HNN learn the agreement of two views.



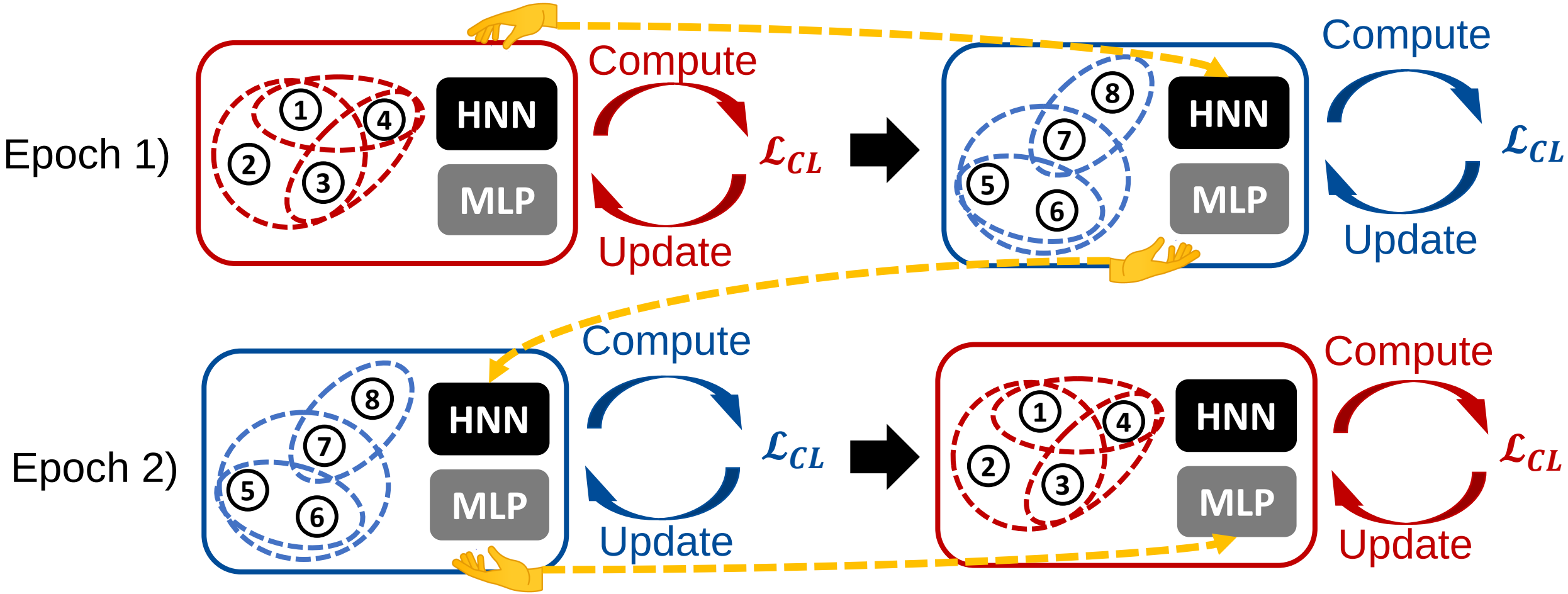
Proposed Learning Method: PCL

- We propose PCL: Partitioning-based Contrastive Learning.
  - We first divide the input hypergraph into several partitions.
  - Then, we train HNN by CL, by regarding each partition as a mini-batch of CL.



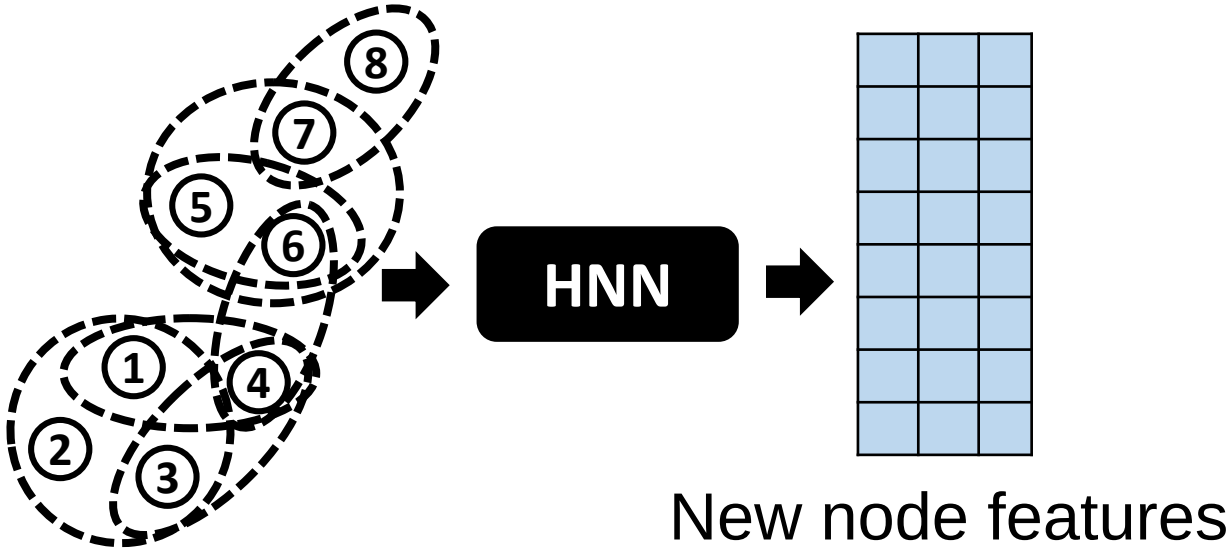
Proposed Learning Method: PCL

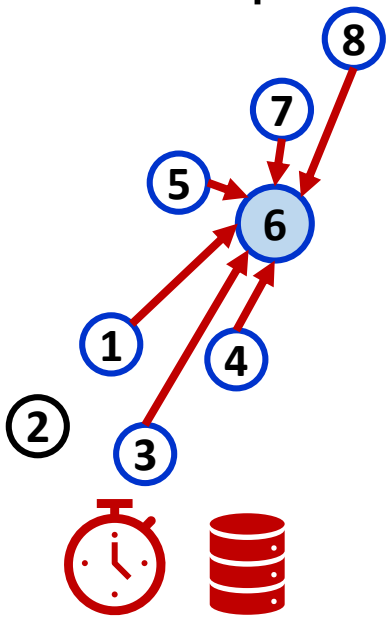
- Specifically, we do CL for each partition, for the given number of epochs.

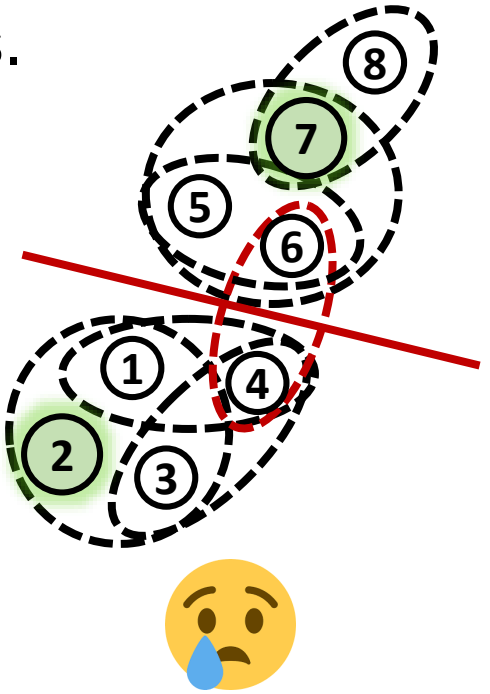


Proposed Learning Method: PCL

- When the CL process is done, we obtain node embeddings with HNN.
- Here, we do not additionally train HNN to:
  - Reduce cost of message passing process.
  - Utilize pair of nodes (hyperedges) in different partitions.



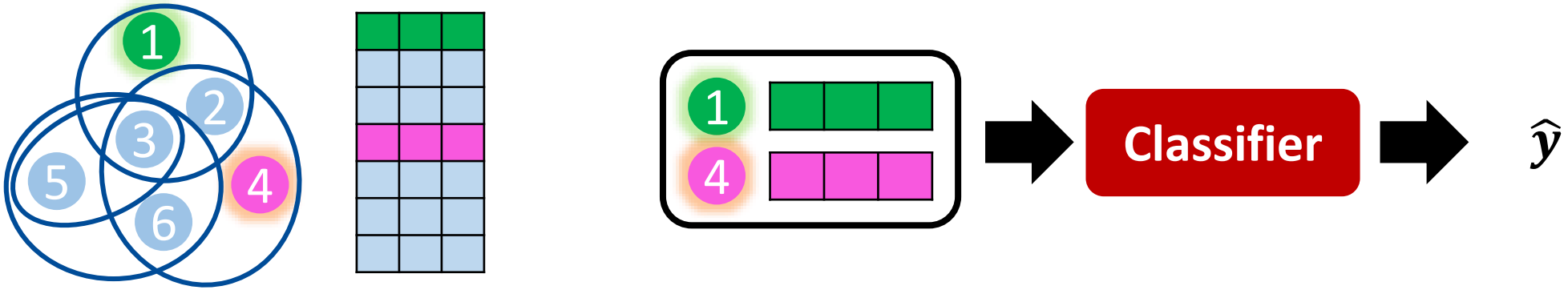




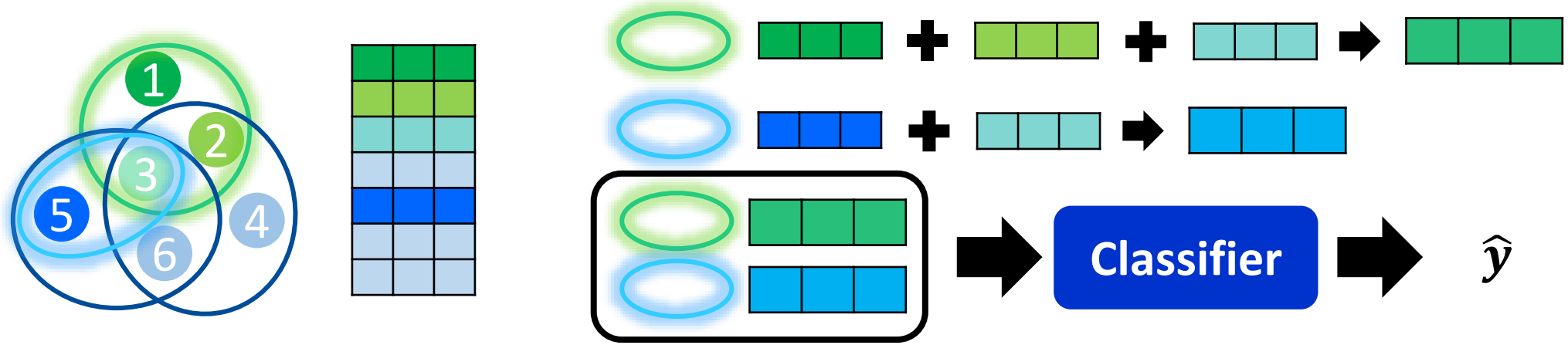
Proposed Learning Method: PCL

- With node embeddings and a classifier, we perform pair-level tasks.

Node level

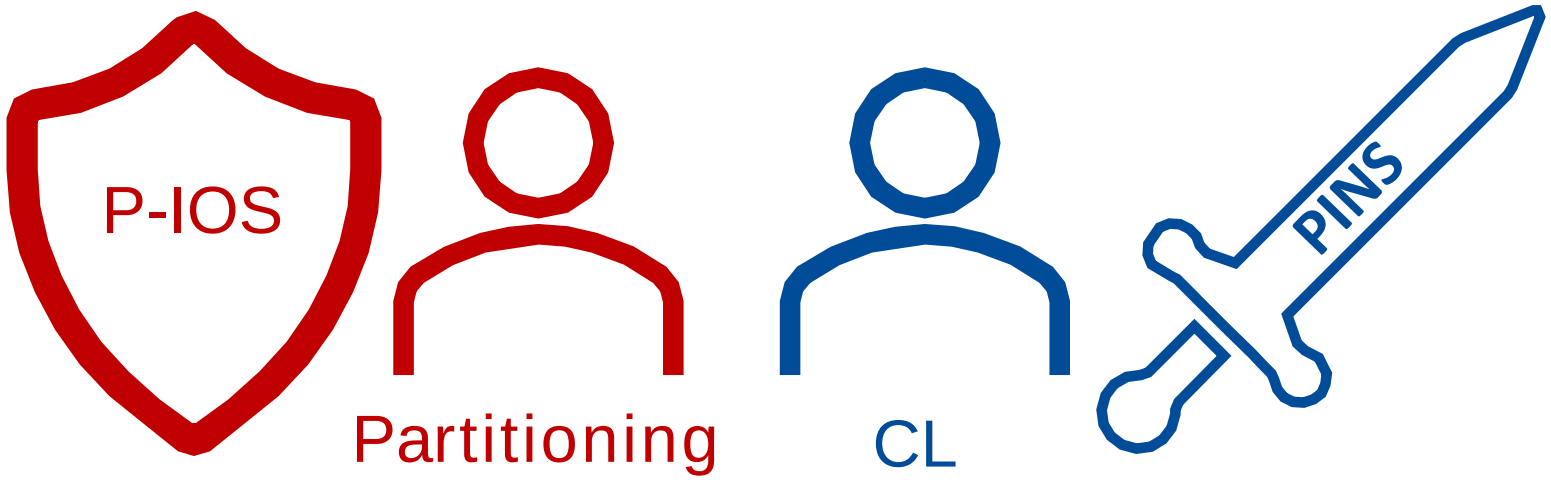


Hyperedge level



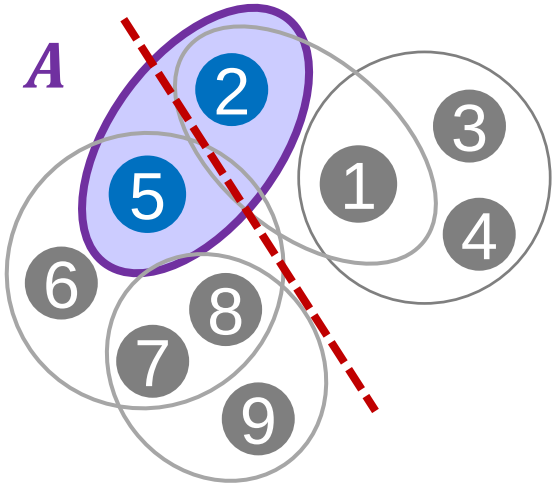
Additional Tools for PCL

- We propose two additional tools for PCL.
  - P-IOS: Partitioning technique that mitigates topological information loss.
  - PINS: CL technique that encourages HNN to learn inter-partition dissimilarity.

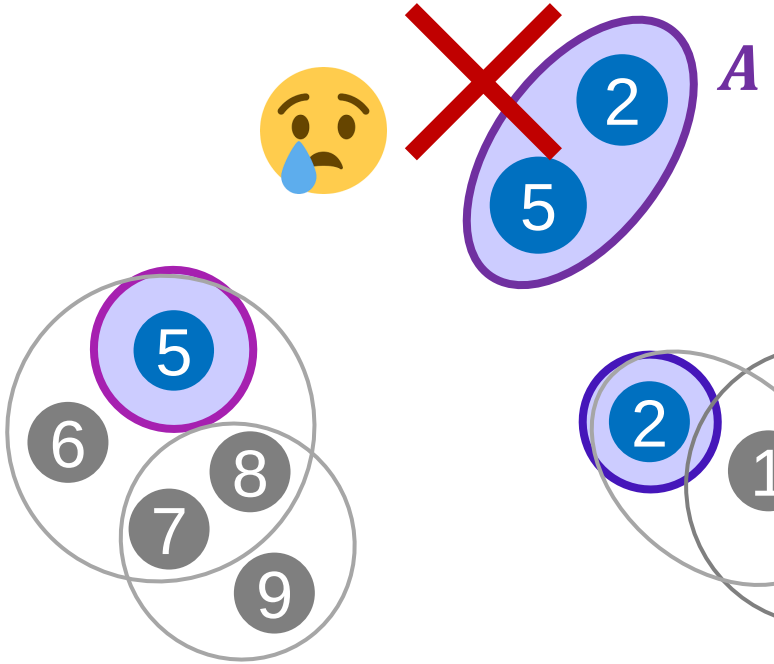


Additional Tools for PCL

- P-IOS: Partitioning with the Inclusion of Outsider Set.
  - Certain hyperedges are destroyed during the partitioning process.



Original hypergraph

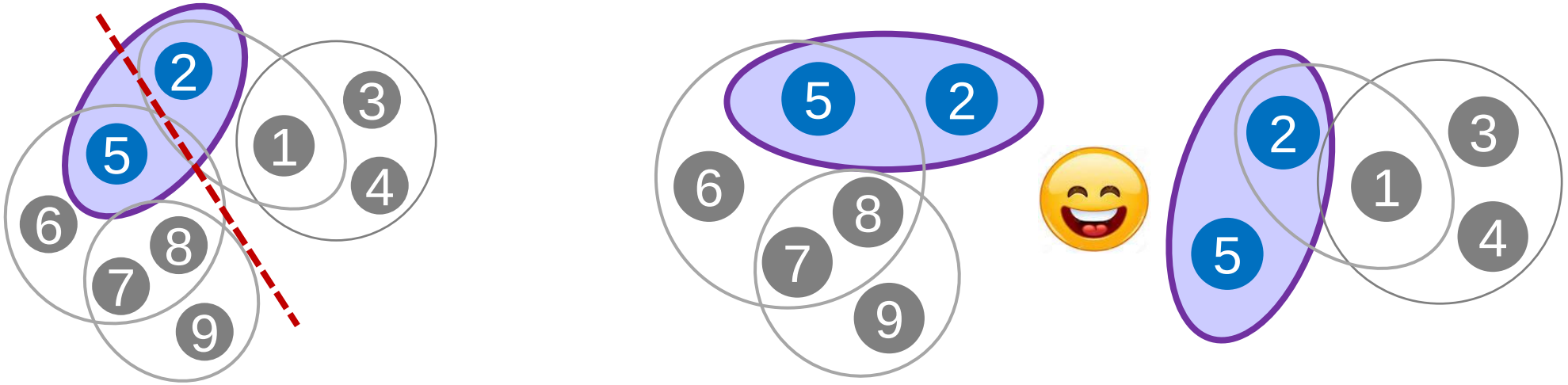


Partition 1

Partition 2

Additional Tools for PCL

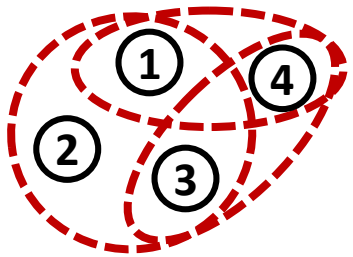
- P-IOS: Partitioning with the Inclusion of Outsider Set.
  - Certain hyperedges are destroyed during the partitioning process.
  - P-IOS recovers destroyed hyperedges and preserves every 1-hop neighboring node.

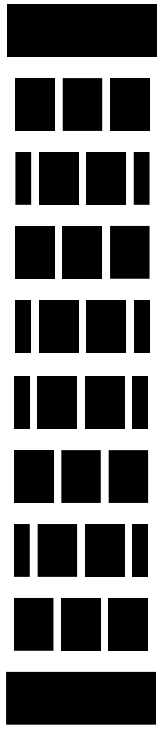


Original hypergraph P-IOS Partition 1 P-IOS Partition 2

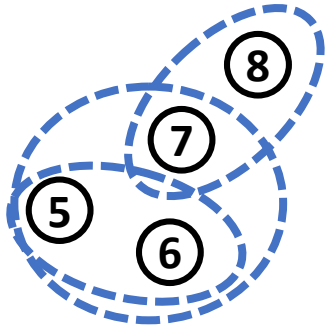
Additional Tools for PCL

- PINS: Previous partItion's Negative Samples.
  - Basic PCL utilizes a single partition at a time, and HNN cannot learn dissimilarity between nodes at different partitions.



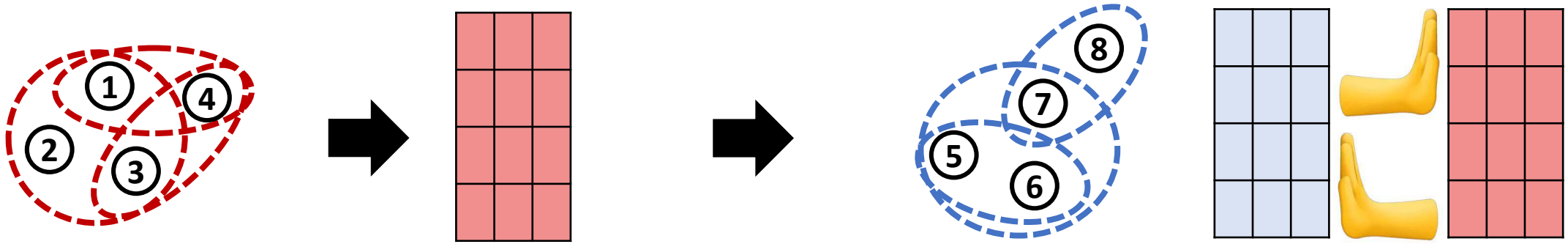


Can't utilize another partition.



Additional Tools for PCL

- PINS: Previous partItion's Negative Samples.
  - Basic PCL utilizes a single partition at a time, and HNN can't learn dissimilarity between nodes at different partitions.
  - PINS uses embeddings of previous partitions as negative samples.



Training done in this partition

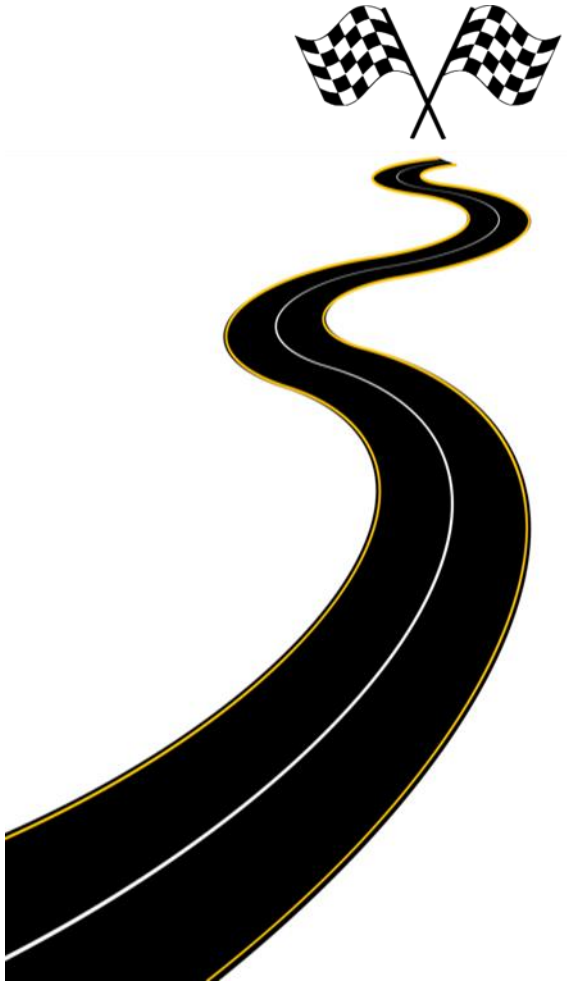
Obtain embedding

Training in the next partition

Push each other

Roadmap

- Overview
- Proposed Tasks
- Proposed Datasets
- Proposed Training Method
- Experimental Results
- Conclusions



Research Questions

- We aim to analyze the following four points:

- RQ1) Accuracy of PCL for the proposed pair-level tasks.

- RQ2) Effectiveness of PINS.

- RQ3) Memory consumption of PINS.

- RQ4) Effectiveness of P-IOS.

Used Datasets

- We utilize 5 real-world hypergraph datasets.

Existing

Proposed

Dataset	# Nodes	# Hyperedges	# Features	# Classes
DBLP	41,302	22,263	1,425	6
Trivago	172,738	233,202	300	160
OGBN-MAG	736,389	1,134,649	128	349
AMiner	13,262,573	22,552,647	300	257
MAG	27,320,375	30,175,013	300	257

trivago

OGB

AMiner

Microsoft Research

Experimental Results

- RQ1) Accuracy of PCL on Task 1 (Hyperedge disambiguation).

Data Type	Methods	DBLP AP	DBLP AUROC	Trivago AP	Trivago AUROC	OGBN-MAG AP	OGBN-MAG AUROC	AMiner AP	AMiner AUROC	MAG AP	MAG AUROC	Avg Rank
Only X	MLP	65.2 ± 3.4	62.2 ± 3.4	60.2 ± 1.6	60.7 ± 2.5	73.1 ± 2.6	71.9 ± 2.8	91.7 ± 0.3	92.9 ± 0.3	91.2 ± 0.6	92.6 ± 0.6	9.8
Full Graph	GCN	83.6 ± 1.2	84.6 ± 1.5	61.6 ± 0.8	58.1 ± 2.4	80.1 ± 1.7	79.8 ± 2.0	OOM	OOM	OOM	OOM	9.7
	GAT	83.5 ± 1.0	84.6 ± 1.0	61.6 ± 0.5	60.0 ± 2.7	76.5 ± 2.3	75.7 ± 3.0	OOM	OOM	OOM	OOM	10.3
	BGRL	84.5 ± 1.2	84.3 ± 1.7	72.0 ± 1.8	72.2 ± 2.2	83.1 ± 0.6	83.9 ± 0.5	OOM	OOM	OOM	OOM	8
	GGD	80.9 ± 1.6	80.6 ± 2.1	71.8 ± 0.9	72.5 ± 1.6	76.1 ± 1.1	76.1 ± 1.3	OOM	OOM	OOM	OOM	9.7
Full Hypergraph	HGNN	77.3 ± 2.5	80.0 ± 3.1	72.1 ± 5.1	76.6 ± 1.4	91.2 ± 0.7	92.9 ± 0.6	OOM	OOM	OOM	OOM	7.7
	UniGCNII	78.1 ± 3.4	77.1 ± 3.9	71.8 ± 1.5	75.6 ± 2.3	86.9 ± 5.6	88.3 ± 6.7	OOM	OOM	OOM	OOM	8.7
	ALLSET	53.1 ± 1.2	55.8 ± 2.2	53.7 ± 1.2	56.8 ± 2.2	65.3 ± 1.4	73.4 ± 1.6	OOM	OOM	OOM	OOM	13.1
	HCL	**91.0 ± 1.7**	**92.6 ± 2.1**	73.7 ± 0.7	78.8 ± 1.1	**94.2 ± 0.4**	**95.5 ± 0.3**	OOM	OOM	OOM	OOM	4.5
Partitioned Graph	GCN	84.5 ± 1.6	85.9 ± 1.6	65.9 ± 1.9	66.7 ± 2.0	82.3 ± 2.6	83.4 ± 2.8	81.2 ± 0.6	81.5 ± 0.6	OOM	OOM	7.9
	GAT	84.7 ± 1.3	86.4 ± 1.2	62.2 ± 3.1	62.5 ± 4.6	78.0 ± 1.8	79.6 ± 1.8	81.3 ± 0.5	81.0 ± 0.6	OOM	OOM	8.2
	BGRL	85.8 ± 1.7	85.6 ± 1.7	82.4 ± 2.2	83.5 ± 2.1	91.4 ± 0.5	92.8 ± 0.5	90.1 ± 0.8	90.9 ± 0.8	89.7 ± 0.7	90.8 ± 0.9	3.9
	GGD	80.1 ± 2.4	80.8 ± 2.3	76.9 ± 1.6	78.3 ± 1.5	88.7 ± 1.0	90.2 ± 0.9	82.5 ± 1.2	83.6 ± 1.5	80.1 ± 1.3	81.2 ± 1.4	6.2
Partitioned Hypergraph	HGNN	84.1 ± 2.0	86.2 ± 1.8	71.8 ± 0.8	75.8 ± 1.0	85.9 ± 1.1	88.1 ± 1.0	91.3 ± 0.3	92.5 ± 0.3	89.2 ± 0.7	91.6 ± 0.5	5.4
	UniGCNII	79.9 ± 1.8	80.0 ± 1.8	71.5 ± 2.6	74.2 ± 3.5	77.4 ± 0.9	75.7 ± 1.2	91.8 ± 0.5	92.1 ± 0.5	90.4 ± 0.4	91.9 ± 0.1	7.9
	ALLSET	54.2 ± 1.6	57.6 ± 2.7	53.6 ± 1.2	56.7 ± 2.0	59.3 ± 1.6	65.6 ± 2.4	61.5 ± 1.8	68.6 ± 2.5	OOM	OOM	13.2
Partitioned Hypergraph	**Basic PCL (proposed)**	87.1 ± 1.9	88.3 ± 2.0	**88.2 ± 0.9**	**88.9 ± 0.7**	94.1 ± 0.4	95.1 ± 0.3	**95.5 ± 0.7**	**96.0 ± 0.8**	**96.2 ± 0.3**	**96.9 ± 0.3**	**1.4**

*bold: best **underline: second-best

Experimental Results

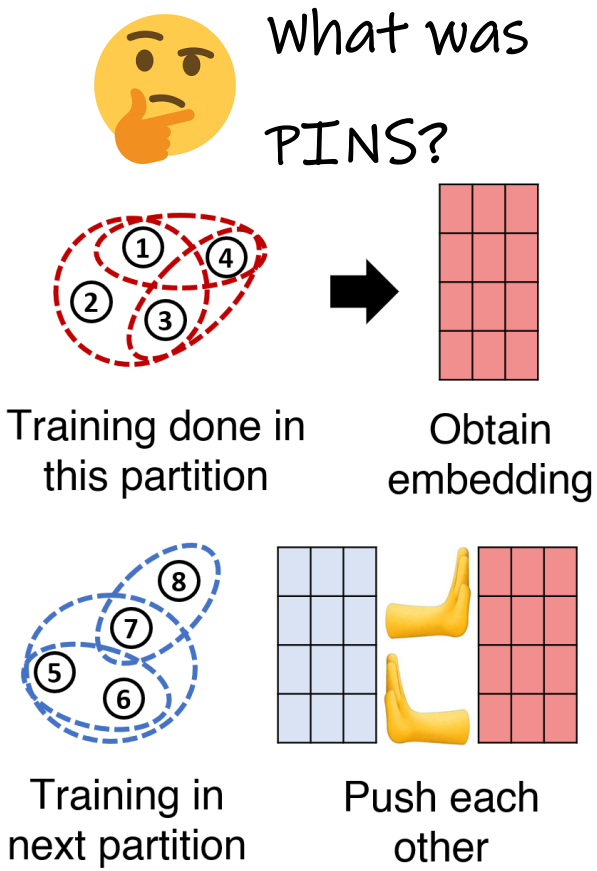
- RQ1) Accuracy of PCL on Task 2 (Local clustering).

Data Type	Methods	DBLP AP	DBLP AUROC	Trivago AP	Trivago AUROC	OGBN-MAG AP	OGBN-MAG AUROC	AMiner AP	AMiner AUROC	MAG AP	MAG AUROC	Avg. Rank
Only X	MLP	52.2 ± 2.3	51.3 ± 3.5	50.6 ± 0.4	50.9 ± 0.7	72.1 ± 1.0	73.9 ± 0.8	70.5 ± 0.9	72.9 ± 0.8	76.2 ± 1.3	79.4 ± 0.9	7.1
Full Graph	GCN	54.6 ± 4.0	54.1 ± 4.7	50.1 ± 0.2	50.2 ± 0.1	53.5 ± 0.3	53.8 ± 0.4	OOM	OOM	OOM	OOM	11.1
	GAT	55.1 ± 4.2	55.3 ± 5.9	50.1 ± 0.0	50.0 ± 0.0	OOM	OOM	OOM	OOM	OOM	OOM	11.6
	BGRL	55.1 ± 5.8	52.9 ± 3.1	50.3 ± 0.2	50.4 ± 0.2	53.2 ± 0.3	53.3 ± 0.3	OOM	OOM	OOM	OOM	10.9
	GGD	58.6 ± 9.5	56.0 ± 8.6	50.3 ± 0.2	50.3 ± 0.2	53.0 ± 0.3	53.1 ± 0.3	OOM	OOM	OOM	OOM	10.1
Full Hypergraph	HGNN	56.9 ± 5.3	54.9 ± 6.3	54.8 ± 3.7	54.8 ± 3.6	78.2 ± 1.0	80.1 ± 0.7	OOM	OOM	OOM	OOM	6.3
	UniGCNII	55.2 ± 4.6	52.9 ± 4.5	51.6 ± 1.0	51.1 ± 0.9	76.7 ± 1.2	79.0 ± 1.1	OOM	OOM	OOM	OOM	7.8
	ALLSET	54.2 ± 4.1	56.1 ± 5.4	51.0 ± 0.7	51.7 ± 0.7	60.0 ± 2.2	61.8 ± 2.5	OOM	OOM	OOM	OOM	8.6
	HCL	63.6 ± 6.9	61.4 ± 7.9	58.6 ± 2.6	58.8 ± 4.2	**78.5 ± 1.0**	**80.9 ± 0.7**	OOM	OOM	OOM	OOM	4.8
Partitioned Graph	GCN	51.9 ± 0.4	52.2 ± 0.4	50.1 ± 0.0	50.1 ± 0.0	51.6 ± 0.4	51.7 ± 0.4	54.9 ± 0.7	54.3 ± 0.7	OOM	OOM	12.7
	GAT	52.4 ± 0.6	53.1 ± 0.6	50.4 ± 0.7	50.5 ± 1.0	54.1 ± 0.7	54.3 ± 0.8	55.2 ± 0.4	54.6 ± 0.4	OOM	OOM	10.3
	BGRL	58.8 ± 7.7	56.8 ± 5.6	**61.2 ± 3.9**	**61.4 ± 3.9**	65.6 ± 1.0	66.4 ± 0.8	65.8 ± 0.5	67.1 ± 0.6	71.0 ± 1.0	72.7 ± 0.9	3.6
	GGD	**67.7 ± 12.7**	**67.2 ± 13.3**	59.4 ± 3.8	59.4 ± 3.5	64.8 ± 1.0	65.5 ± 0.6	62.8 ± 0.8	64.3 ± 0.8	68.5 ± 0.9	70.5 ± 0.9	3.8
Partitioned Hypergraph	HGNN	55.2 ± 6.5	55.2 ± 7.3	52.0 ± 1.0	52.4 ± 2.0	68.8 ± 1.3	69.9 ± 1.6	57.6 ± 1.6	57.5 ± 2.1	62.8 ± 3.0	62.5 ± 3.6	6.6
	UniGCNII	52.6 ± 1.7	52.3 ± 0.9	50.4 ± 0.3	50.1 ± 0.2	54.9 ± 0.6	54.6 ± 0.5	59.9 ± 1.2	59.6 ± 1.3	65.9 ± 2.2	66.0 ± 2.2	9.8
	ALLSET	53.2 ± 1.6	54.6 ± 2.1	51.5 ± 0.3	52.5 ± 0.5	57.5 ± 1.9	59.2 ± 2.8	58.1 ± 1.1	60.9 ± 1.4	OOM	OOM	8.2
Partitioned Hypergraph	**PCL+PINS (proposed)**	63.2 ± 10.6	64.0 ± 11.5	58.6 ± 1.0	59.2 ± 1.3	77.6 ± 0.8	79.8 ± 0.6	**80.2 ± 1.7**	**81.6 ± 0.8**	**83.1 ± 1.4**	**86.1 ± 1.5**	**2.1**

*bold: best **underline: second-best

Experimental Results

- RQ2) Effectiveness of PINS on Task 2.



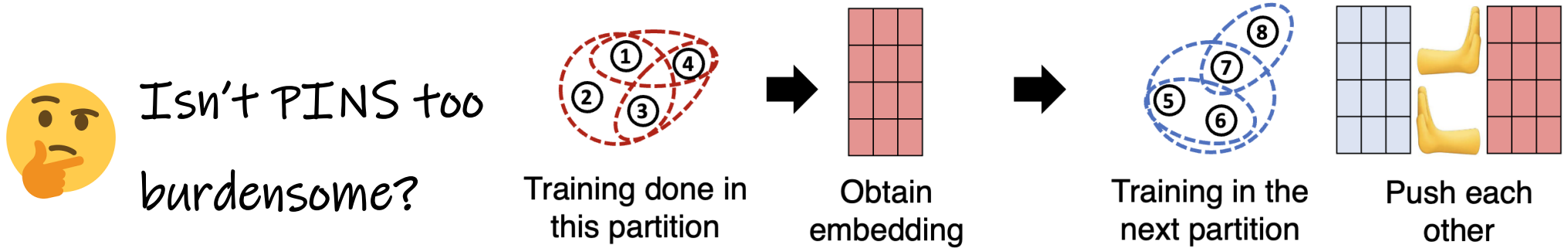
Data	Metric	PCL w/o PINS	PCL+PINS
DBLP	AP	62.4 ± 11.5	**63.2 ± 10.6**
	AUROC	61.1 ± 9.9	**64.0 ± 11.5**
Trivago	AP	**58.6 ± 1.0**	**58.6 ± 1.0**
	AUROC	58.7 ± 1.0	**59.2 ± 1.3**
OGBN-MAG	AP	77.3 ± 0.7	**77.6 ± 0.8**
	AUROC	79.6 ± 0.3	**79.8 ± 0.6**
AMiner	AP	78.6 ± 1.3	**80.2 ± 1.7**
	AUROC	81.3 ± 1.2	**81.6 ± 0.8**
MAG	AP	**83.3 ± 1.1**	83.1 ± 1.4
	AUROC	**86.3 ± 0.7**	86.1 ± 1.5

*bold: best

**underline: second-best

Experimental Results

- RQ3) Memory consumption of PINS.

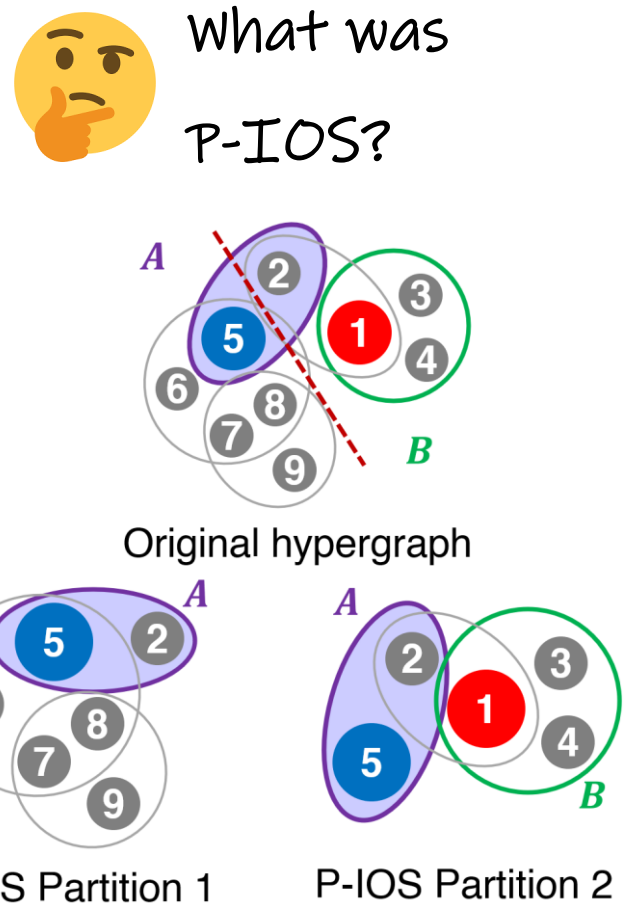


	Method	OGBN-MAG	AMiner	MAG
Average GPU Memory Usage (GB)	PCL w/o PINS	2.344	10.573	23.942
	PCL+PINS	2.347	10.575	23.945

0.1% More

Experimental Results

- RQ4) Effectiveness of P-IOS on Task 1 and Task 2.



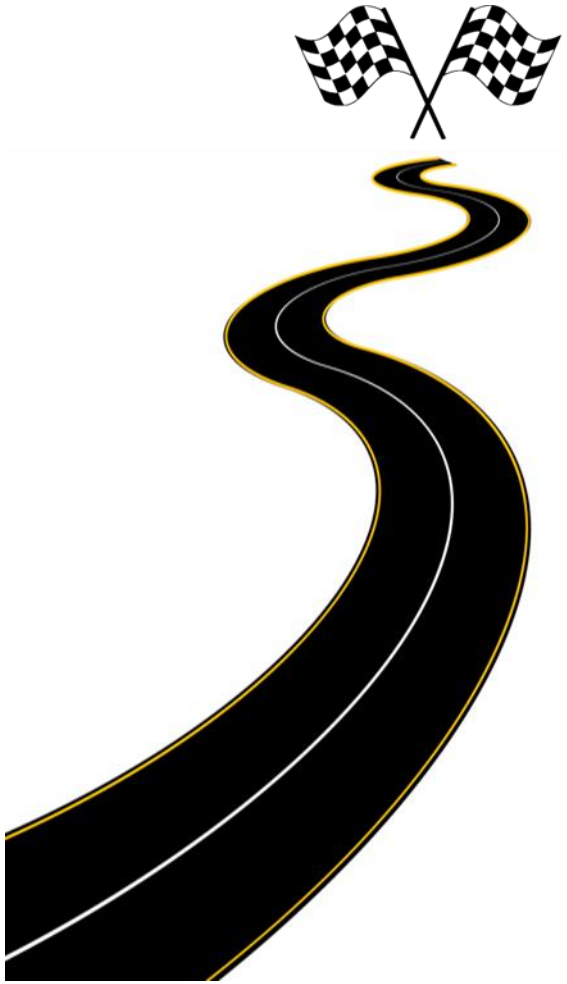
Task	Data	Metric	PCL w/o P-IOS	PCL+P-IOS
Task-I	DBLP	AP	87.1 ± 1.9	**90.8 ± 1.8**
		AUROC	88.3 ± 2.0	**92.9 ± 1.2**
	Trivago	AP	88.2 ± 0.9	**89.4 ± 1.2**
		AUROC	88.9 ± 0.7	**89.5 ± 0.9**
	OGBN-MAG	AP	94.1 ± 0.4	**94.8 ± 0.4**
		AUROC	95.1 ± 0.2	**96.2 ± 0.2**
Task-II	DBLP	AP	62.4 ± 11.5	**64.7 ± 11.5**
		AUROC	61.1 ± 9.9	**62.3 ± 12.3**
	Trivago	AP	58.6 ± 1.0	**59.2 ± 1.0**
		AUROC	58.7 ± 1.0	**59.3 ± 1.0**
	OGBN-MAG	AP	77.3 ± 0.7	**78.8 ± 0.9**
		AUROC	79.6 ± 0.3	**80.9 ± 0.7**

*bold: best

**underline: second-best

Roadmap

- Overview
- Proposed Tasks
- Proposed Datasets
- Proposed Training Method
- Experimental Results
- Conclusions



Conclusions

In this work, we present…

- Tasks: Two pair–level tasks on hypergraphs,

  (Hyperedge disambiguation and Local clustering).

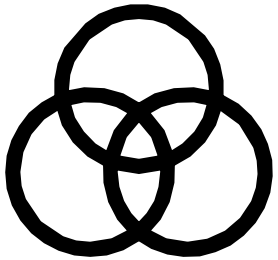
- Datasets: Two large-scale benchmark hypergraph datasets,

  (AMiner and MAG).

- Training methods: Partitioning & Contrastive learning-based scalable training strategy and its additional tools,

  (PCL, PINS, and P-IOS).

Materials



Hypergraph Datasets

Code

Thank You For Listening!

Sunwoo Kim*, Dongjin Lee*, Yul Kim,

Jungho Park, Taeho Hwang, Kijung Shin