KDD2024
BARCELONA, SPAIN

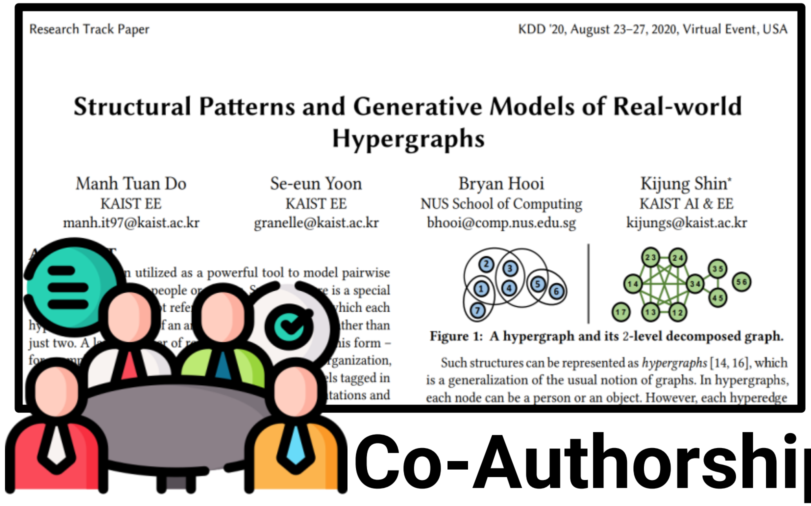
KAIST

Unsupervised Alignment of Hypergraphs with Different Scales

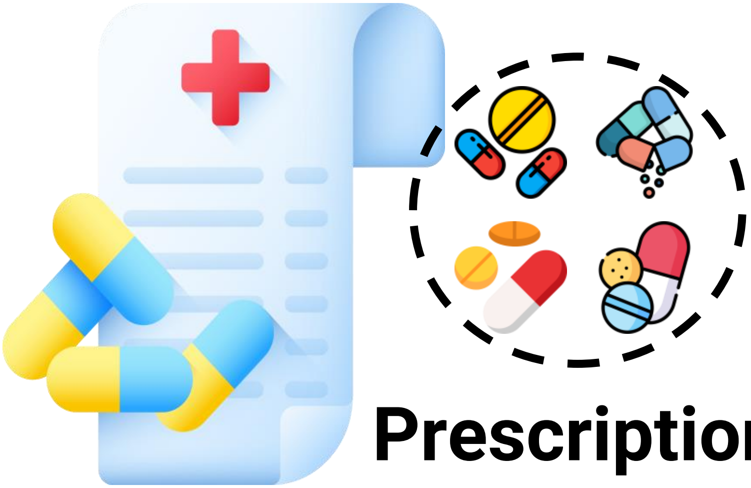
Manh Tuan Do

Kijung Shin

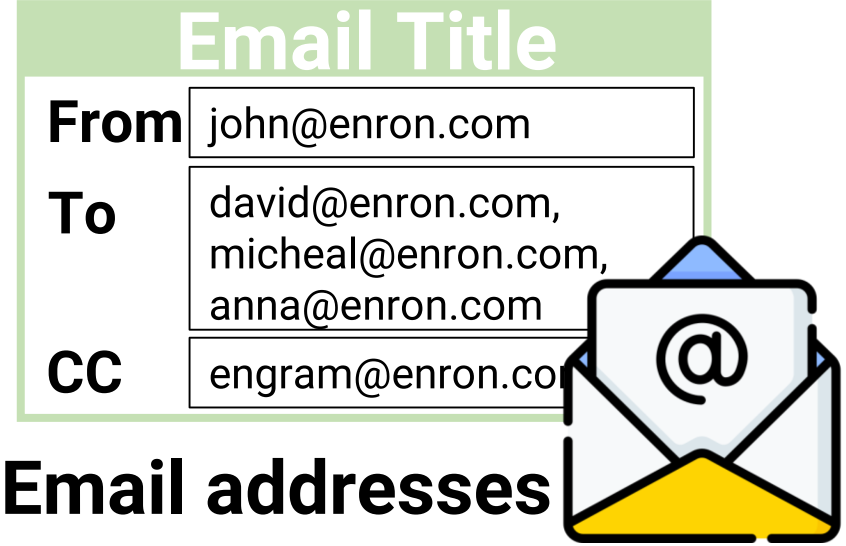
Motivations: Group Interactions are Everywhere!



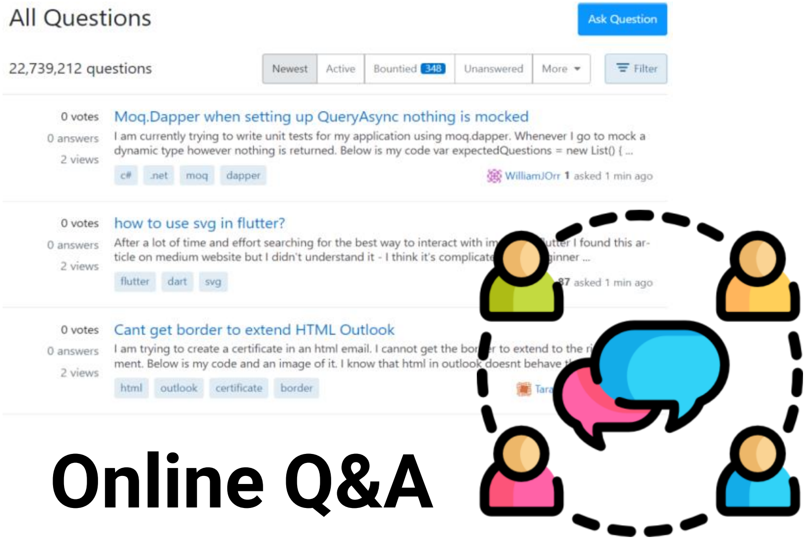
Co-Authorship



Prescription



Email addresses



Online Q&A

Hypergraph Models Group Interactions

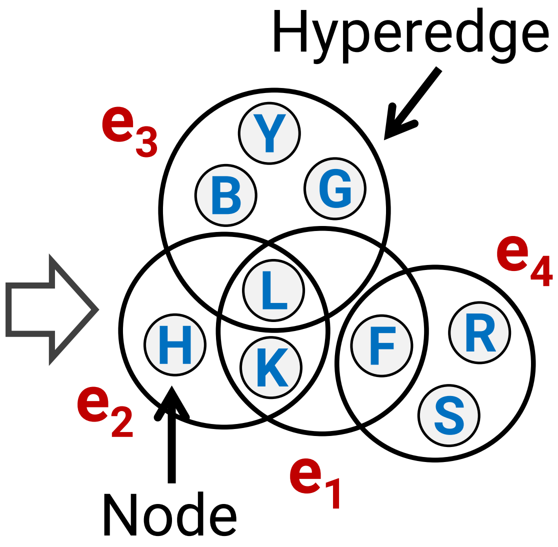
- A **hypergraph** $G = (V, E)$ has a node set V and a hyperedge set E
  - Each **hyperedge** consists of a subset of nodes of any size
- Hypergraphs represent **group interactions** among people/objects

Authors (Nodes)

Jure Leskovec (**L**)	Austin Benson (**B**)
Jon Kleinberg (**K**)	David Gleich (**G**)
Hao Yin (**Y**)	Timos Sellis (**S**)
Christos Faloutsos (**F**)	Nick Roussopoulos (**R**)
Daniel Huttenlocher (**H**)	

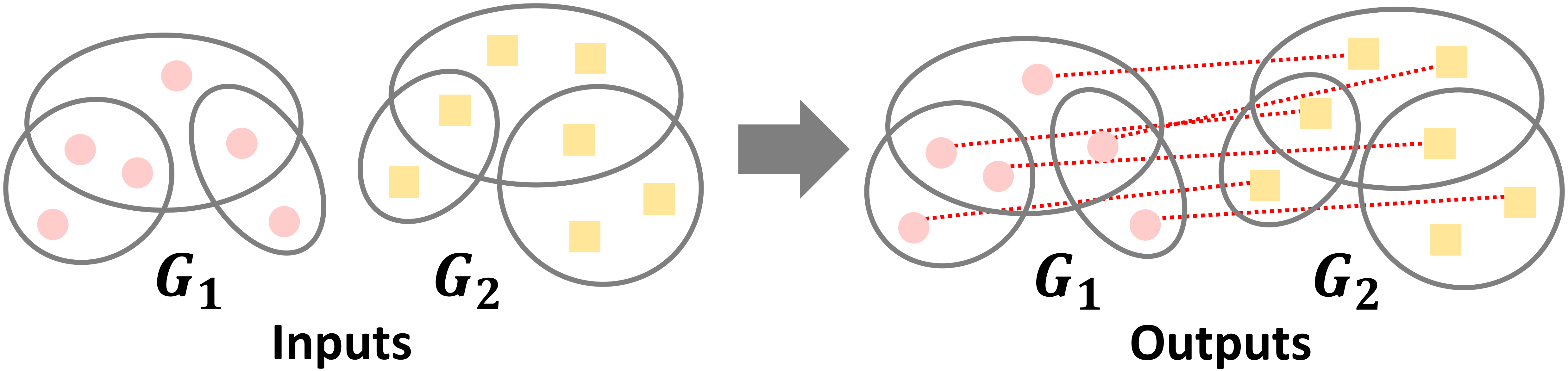
Publications (Hyperedges)

e_1: (L, K, F) KDD'05
e_2: (L, H, K) WWW'10
e_3: (Y, B, G, L) KDD'17
e_4: (S, R, F) VLDB'87



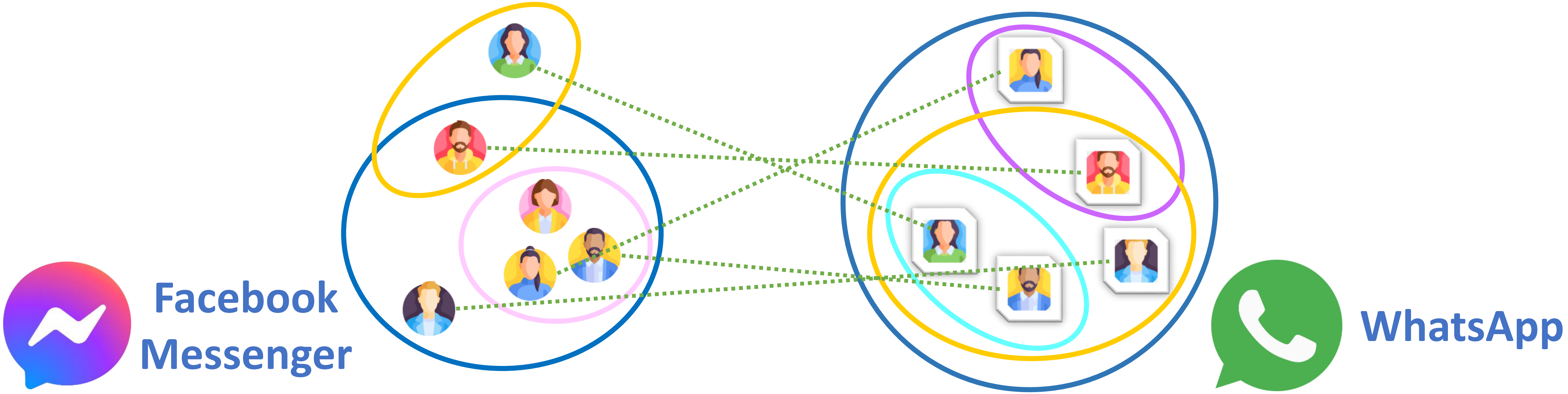
Hypergraph Alignment: Definition

- The focus of this work is **hypergraph alignment**
  - **Given**: two (or more) hypergraphs
  - **to Identify**: the "same nodes" across the hypergraphs



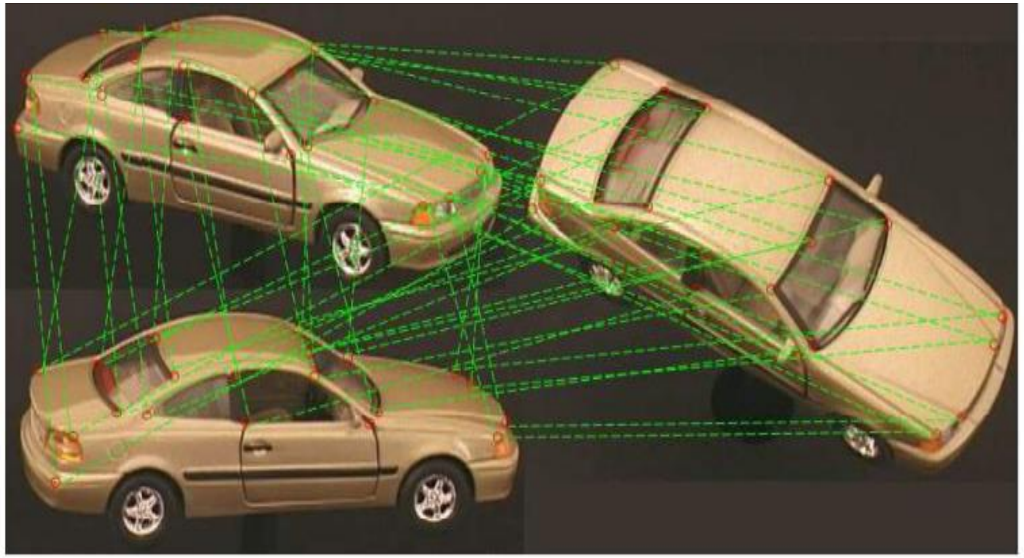
Hypergraph Alignment: Applications

- **User matching in social messaging platforms**
  - **Goal:** to identify the same users in different platforms
  - **Hypergraph:** group chats (hyperedges) among users (nodes)
  - **Applications:** cross-platform marketing and cybersecurity



Hypergraph Alignment: Applications

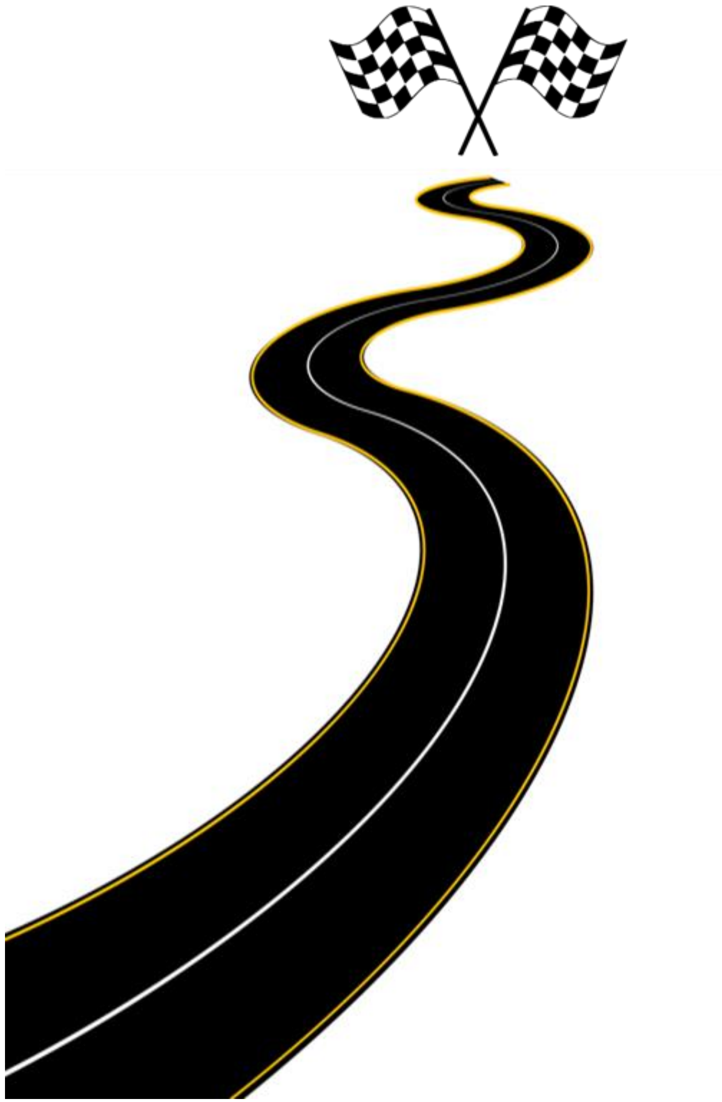
- **Object matching in images**
  - **Goal:** to match pixels (or features) corresponding to the same objects
  - **Hypergraph:** groups of similar (e.g., w.r.t. colors) pixels (or features)
  - **Applications:** medical imaging, image reconstruction, & surveillance



The image is from Yan et al., "*Discrete Hyper-Graph Matching*", CVPR 2015.

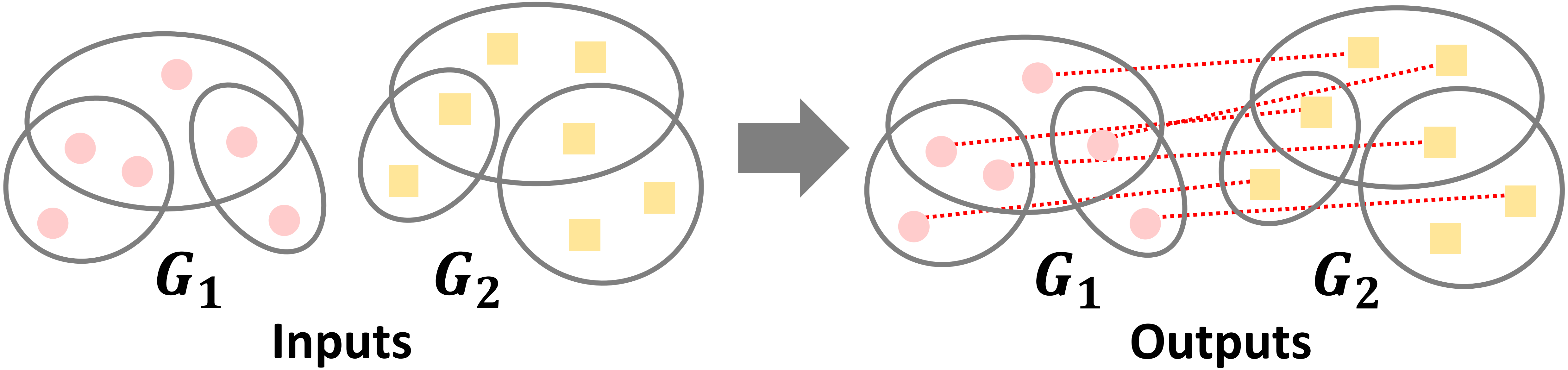
Roadmap

1. Introduction
2. **Challenges <<**
3. Proposed method
4. Experiments
5. Conclusion



Unsupervised Hypergraph Alignment: Definition

- We address the **unsupervised** hypergraph alignment
  - Given: two hypergraphs G_1 and G_2, potentially with **different scales**
  - to Identify: correct node correspondences across G_1 and G_2
  - **No ground-truth node correspondences or node attributes are given**

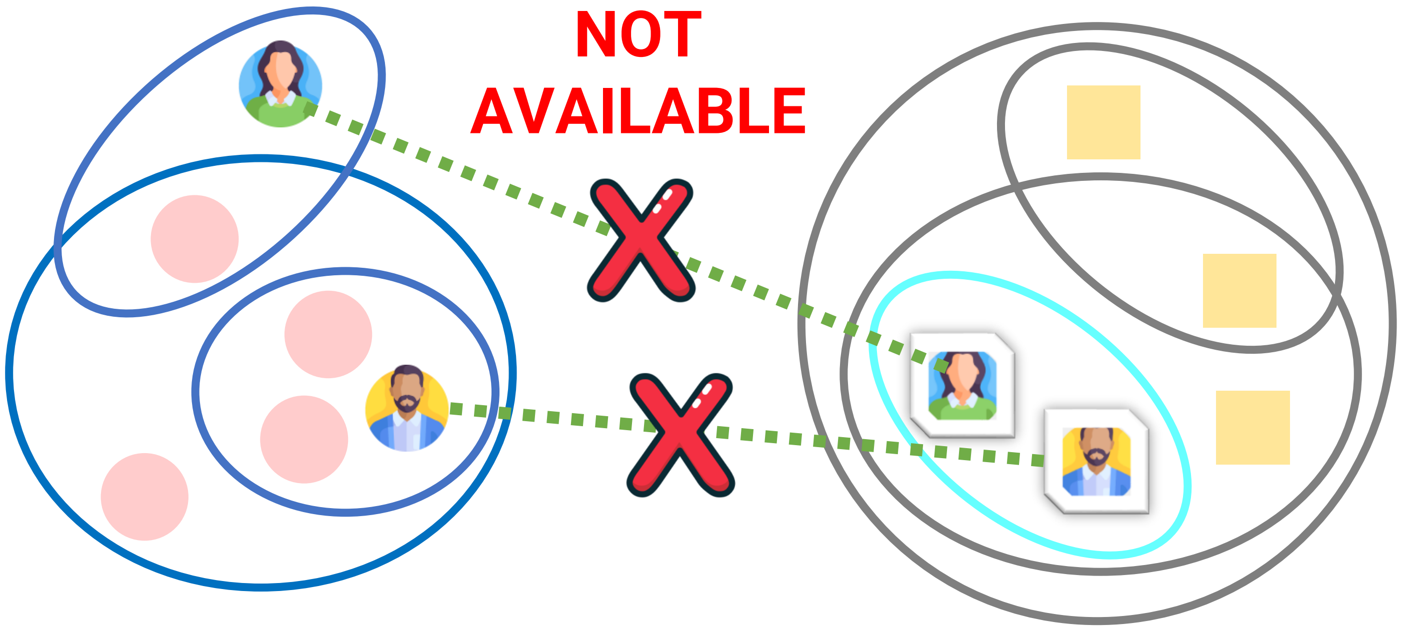


Challenge 1: Absence of Node Attributes

- **Node attributes may not be available** in real-world hypergraphs
  - For example, for messaging platforms, privacy-protection regulations may prevent the disclosure of user information
- It can be desirable to avoid relying on attributes for alignment

Challenge 2: Absence of Supervision

- **Supervision may not be available** in real-world hypergraphs
  - Ground-truth node correspondence may not be available
- It can be desirable to avoid relying on supervision for alignment



Challenge 3: Scale Disparity of Hypergraphs

- Two hypergraphs may be **substantially different in sizes**
  - One may have (much) more nodes or hyperedges than the other

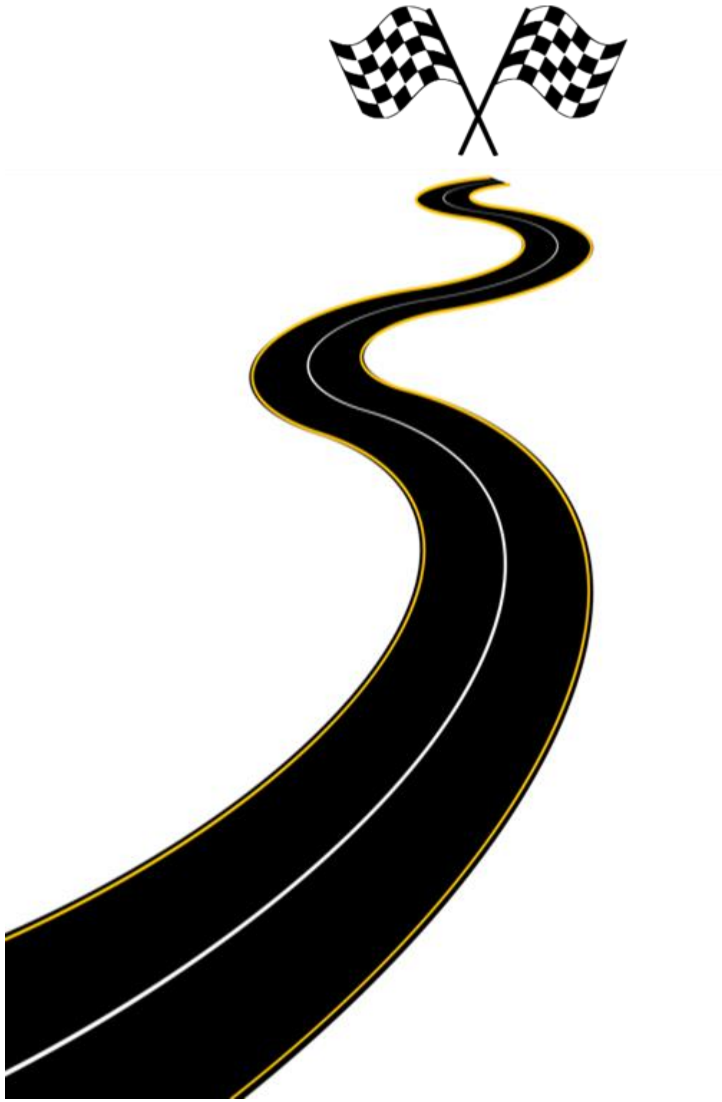
Facebook Messenger
(1 billon active users)

<<

WhatsApp
(2 billion active users)

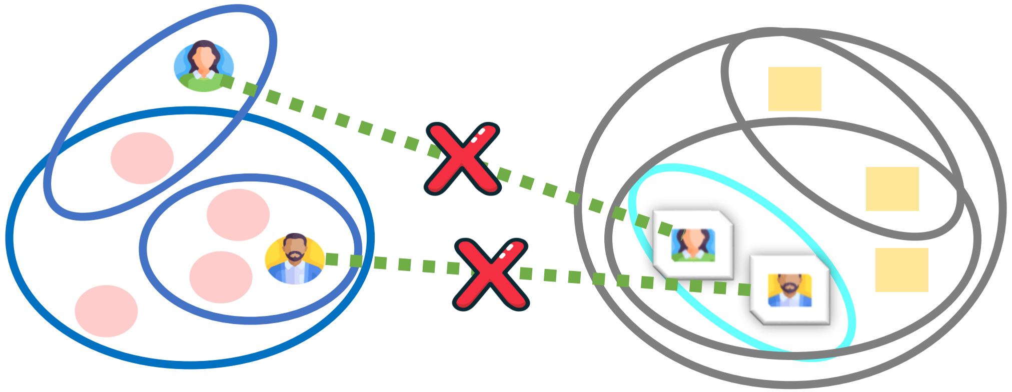
Roadmap

1. Introduction
2. Challenges
3. **Proposed Method <<**
4. Experiments
5. Conclusion



Our Contributions

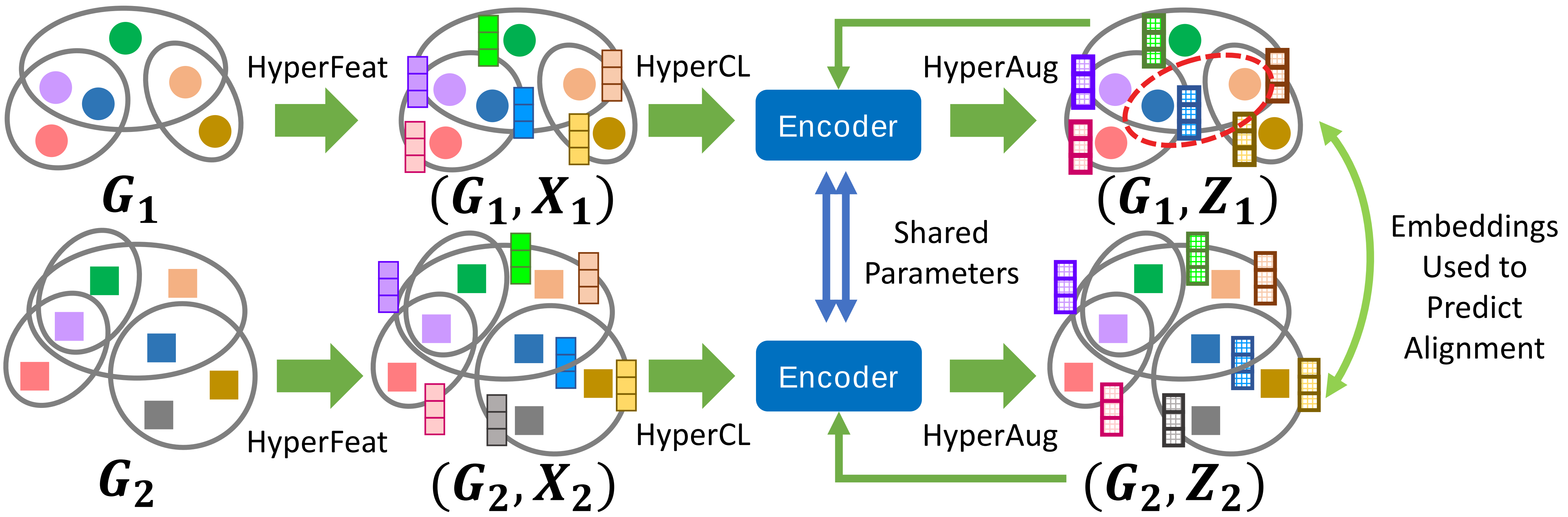
- We propose **HyperAlign** for hypergraph alignment
- It directly addresses the aforementioned challenges
  - Challenge 1: Absence of node attribute
  - Challenge 2: Absence of supervision
  - Challenge 3: Scale disparity of two hypergraphs



1 billon << **2 billons**

Proposed Method: Overview

- HyperAlign learns node embeddings for both hypergraphs
- The embeddings are then used to infer node correspondences

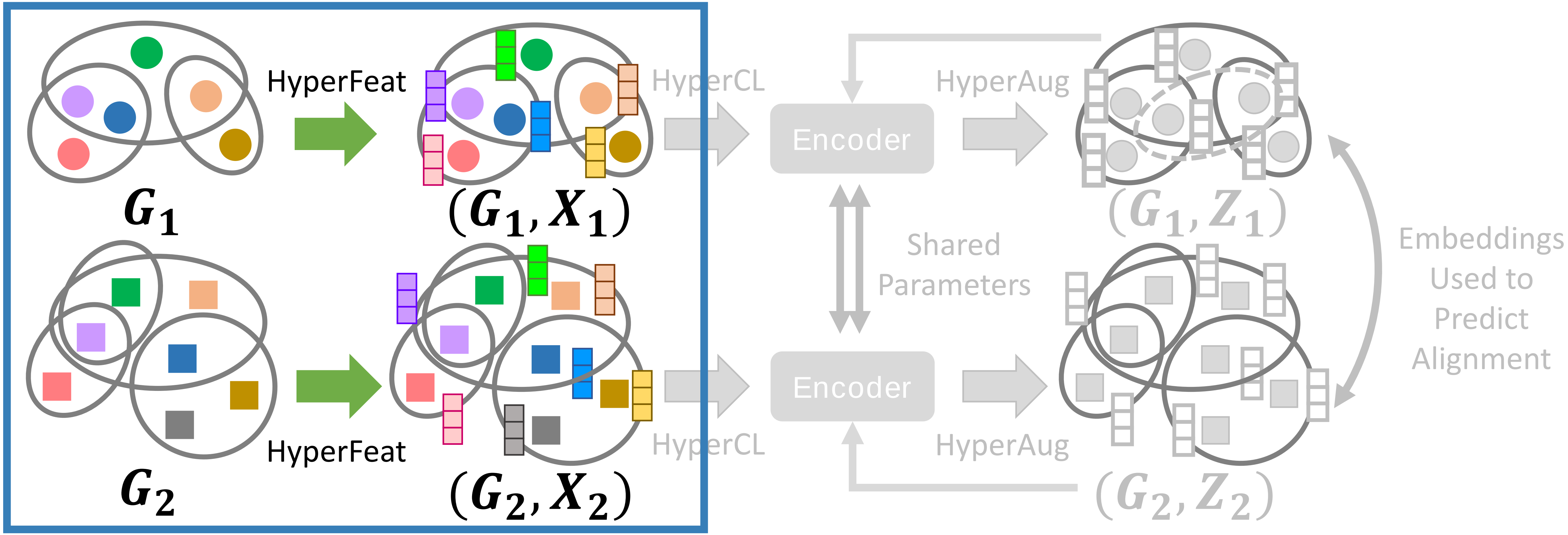


Proposed Method: Overview

- Each step of **HyperAlign** specifically addresses a key challenge:
- **Step 1. HyperFeat: node feature extraction** from hypergraph topology
  - Addressing Challenge 1, absence of node attributes
- **Step 2. HyperCL:** contrastive learning as **"pseudo" supervised alignment**
  - Addressing Challenge 2, absence of supervision
- **Step 3. HyperAug:** adversarial learning with **topological augmentation**
  - Addressing Challenge 3, scale disparity of hypergraphs

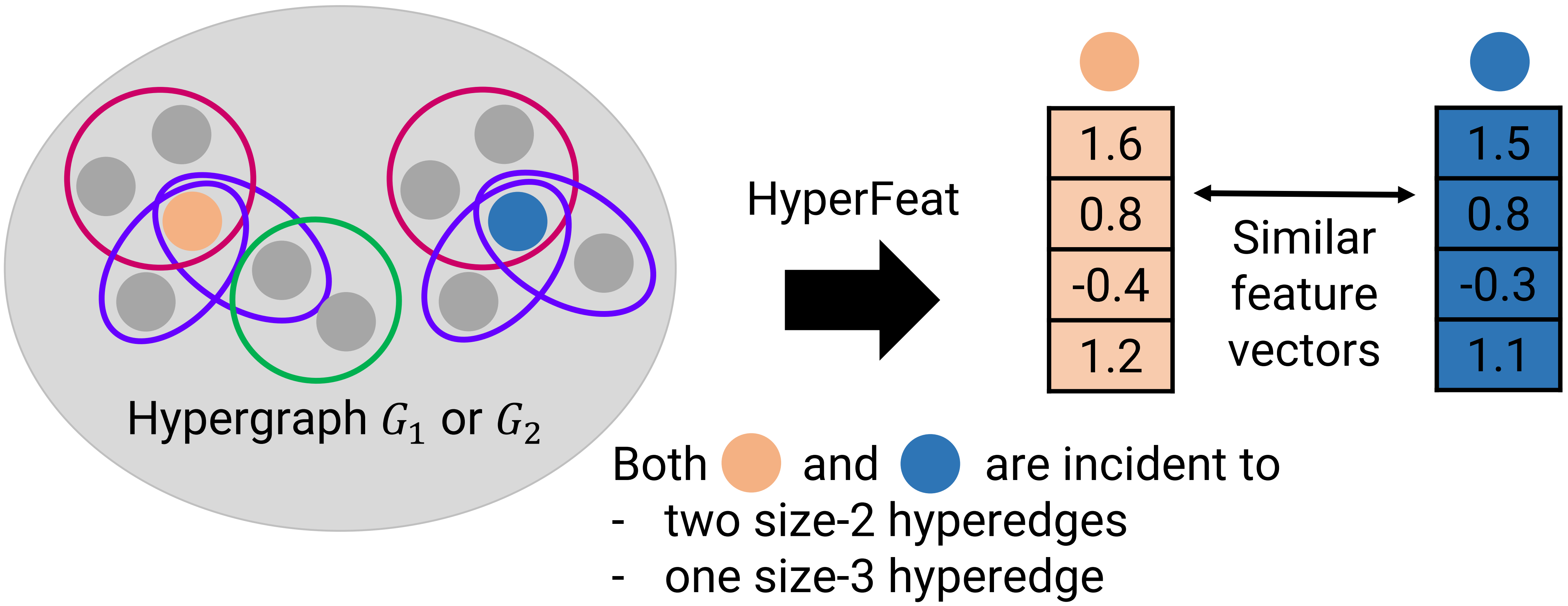
Step 1. HyperFeat: Overview

- Step 1. HyperFeat: **node feature extraction** from hypergraph topology
  - Addressing Challenge 1, absence of node attributes



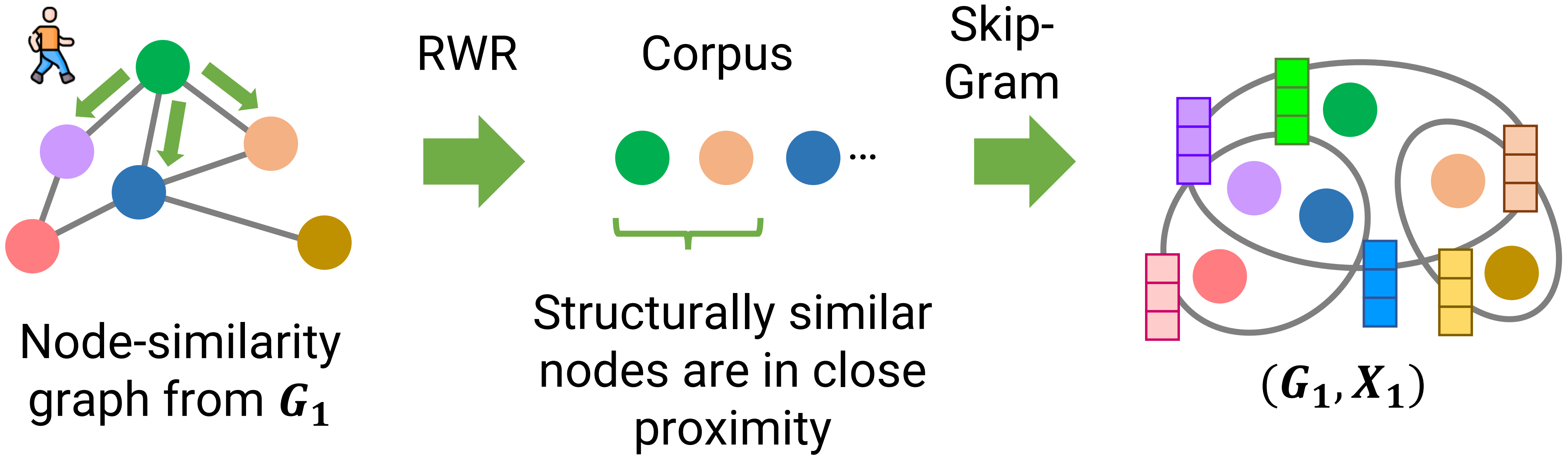
Step 1. HyperFeat: Overview (cont.)

- HyperFeat aims to preserve **structural similarities** within each hypergraph
- Structural similarity reflects the count of incident hyperedges of each size



Step 1. HyperFeat: Details

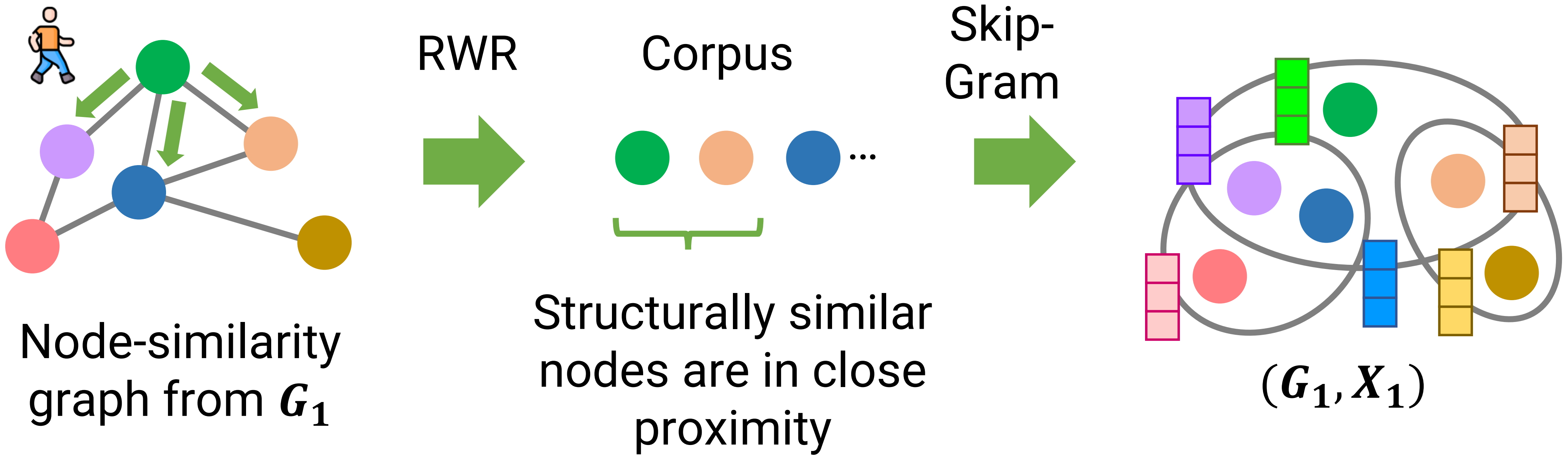
- Node-similarity graph to connect structurally similar nodes
- Random walk with restart (RWR) to obtain a corpus
- Skip-Gram with negative sampling to learn node embeddings from corpus



Details

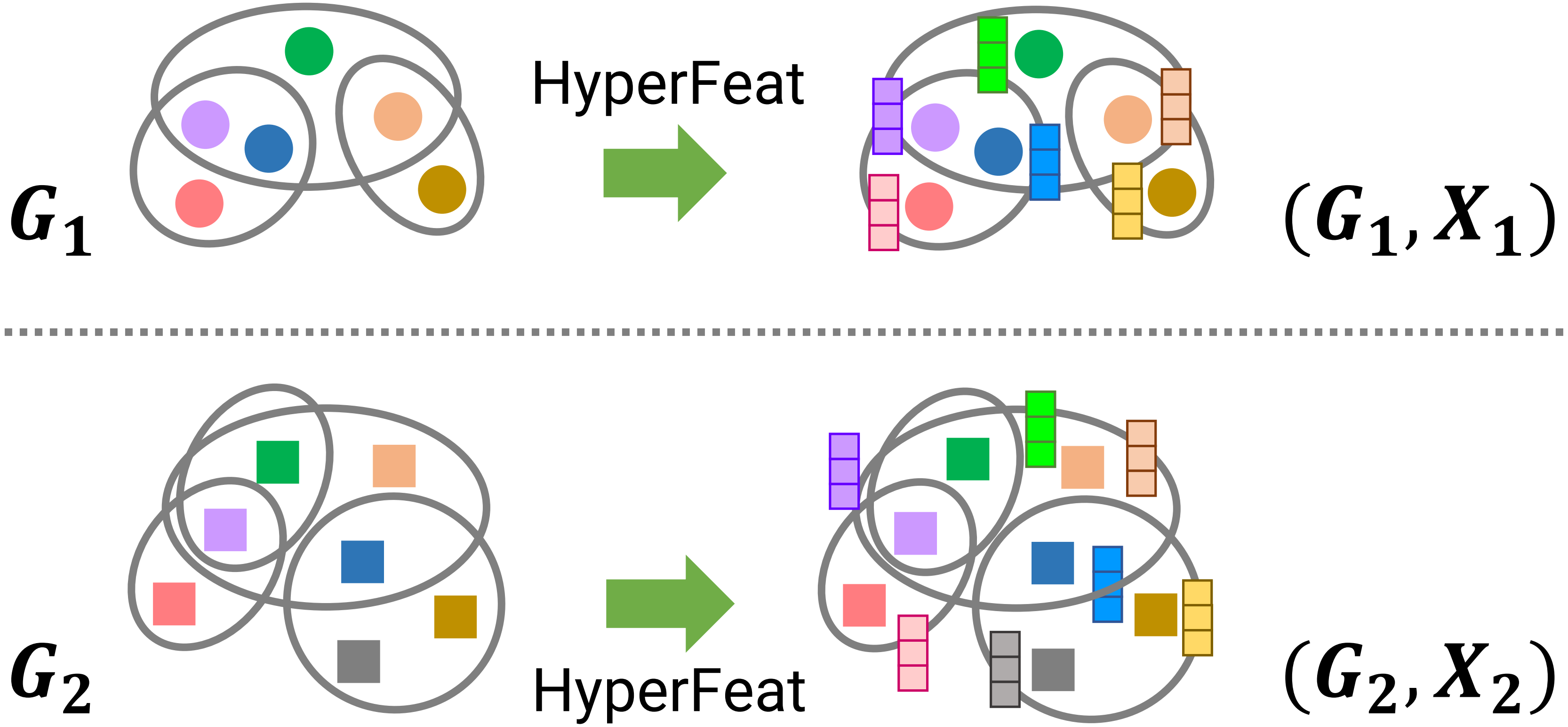
Step 1. HyperFeat: Theoretical Properties

- HyperFeat has **desirable properties** with a sufficiently large corpus:
  - **Equivalence to implicit matrix factorization**
  - **Invariance to node permutation**
  - **Distinguishability of non-isomorphic hypergraphs**



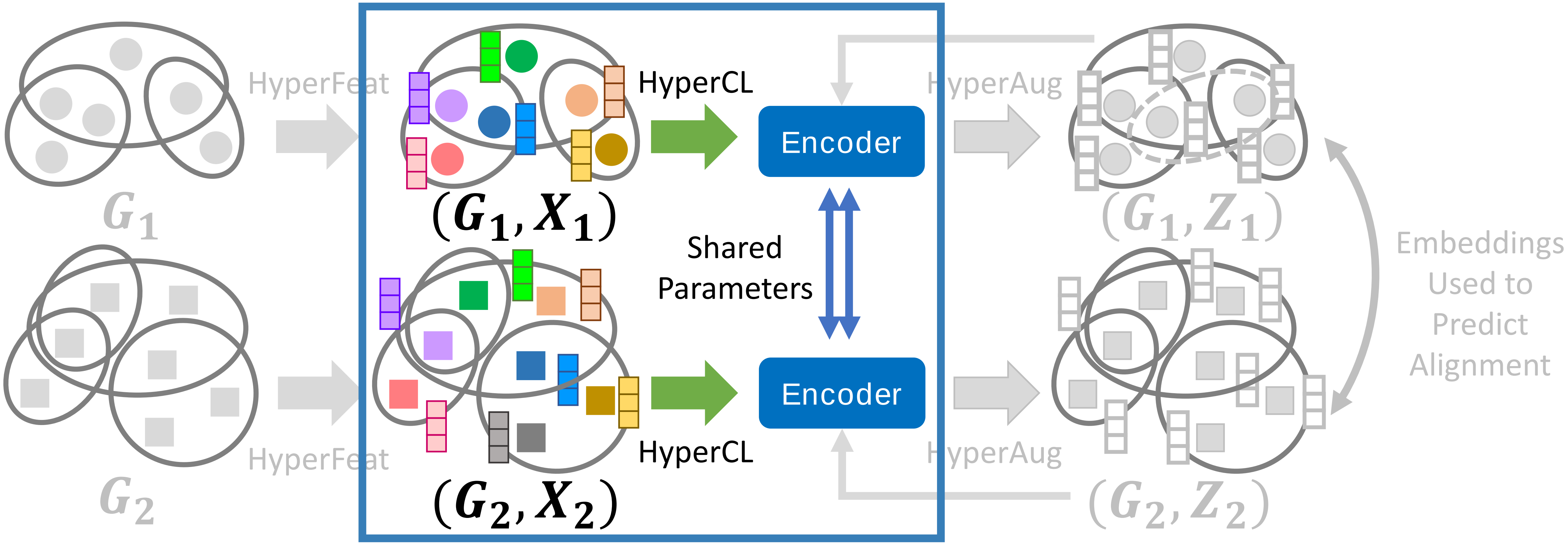
Step 1. HyperFeat: Summary

- HyperFeat is applied to **each hypergraph** to obtain node features
  - Addressing Challenge 1, absence of node attributes



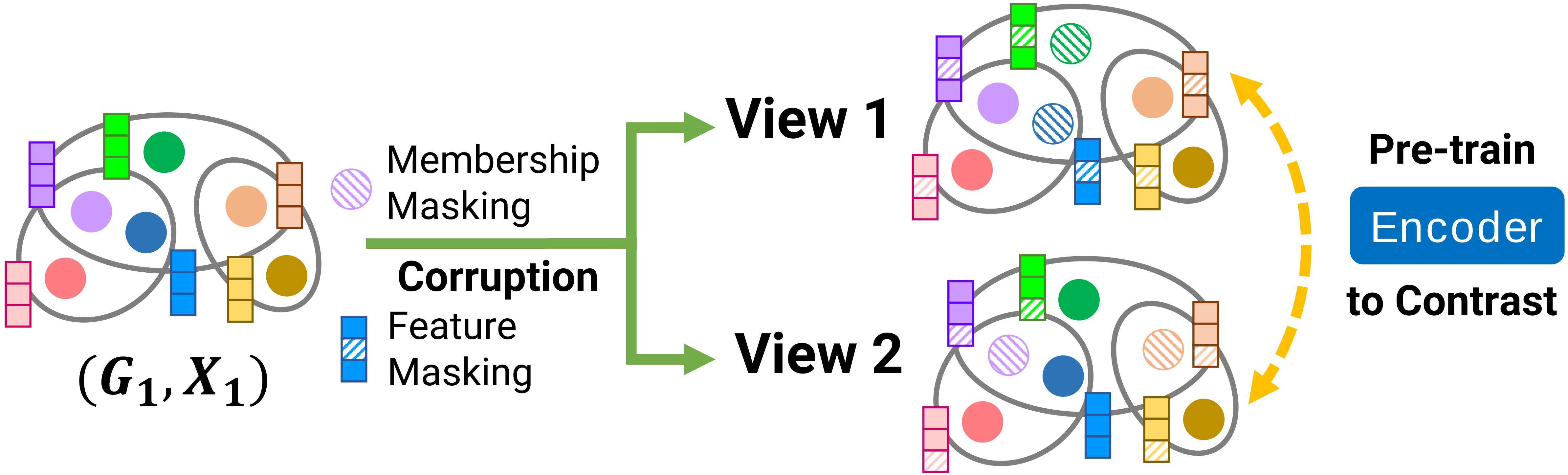
Step 2. HyperCL: Overview

- Step 2. HyperCL: contrastive learning as **"pseudo" supervised alignment**
  - Addressing Challenge 2, absence of supervision



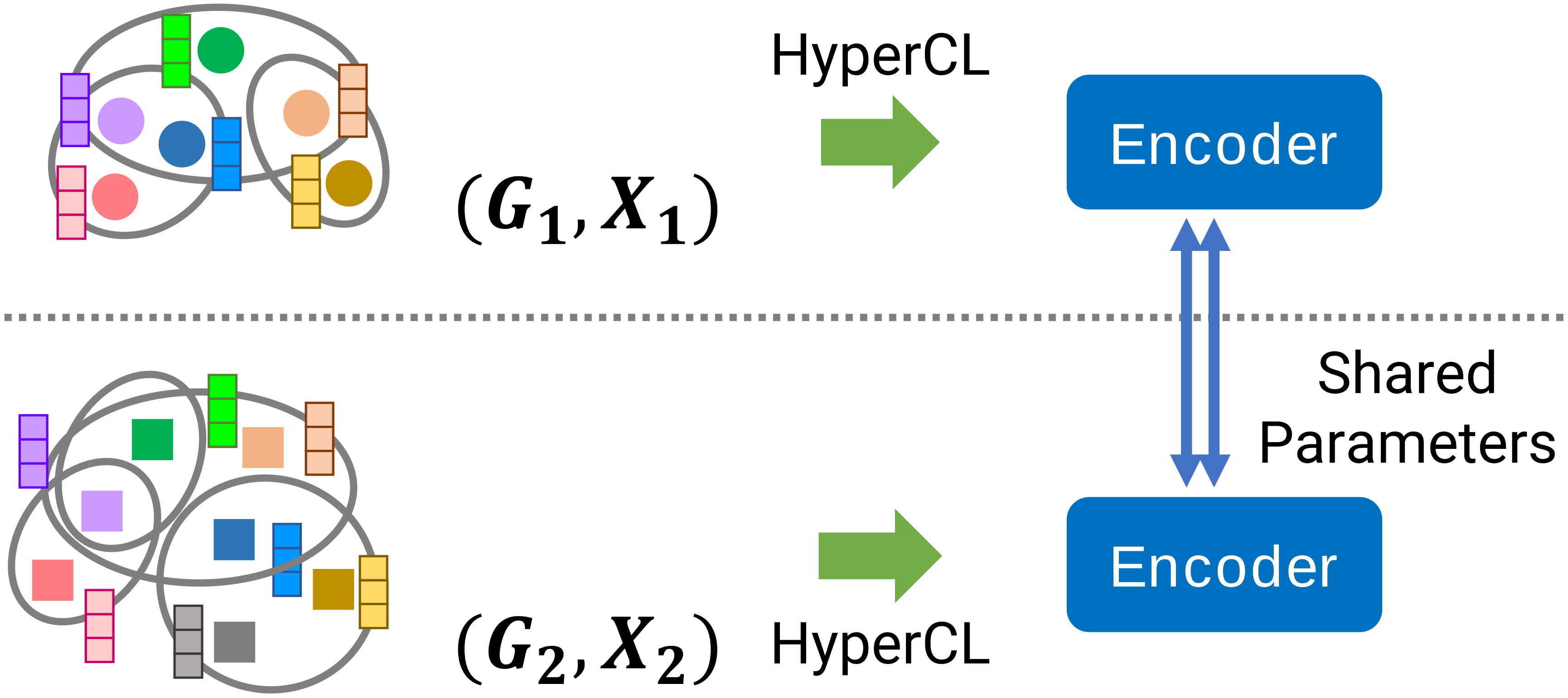
Step 2. HyperCL: Procedures

- HyperCL creates **two views from each hyperedge** through "corruption"
- For the two views, we know the **ground-truth node correspondences**
- HyperCL **pretrains the encoder** to learn the correspondence



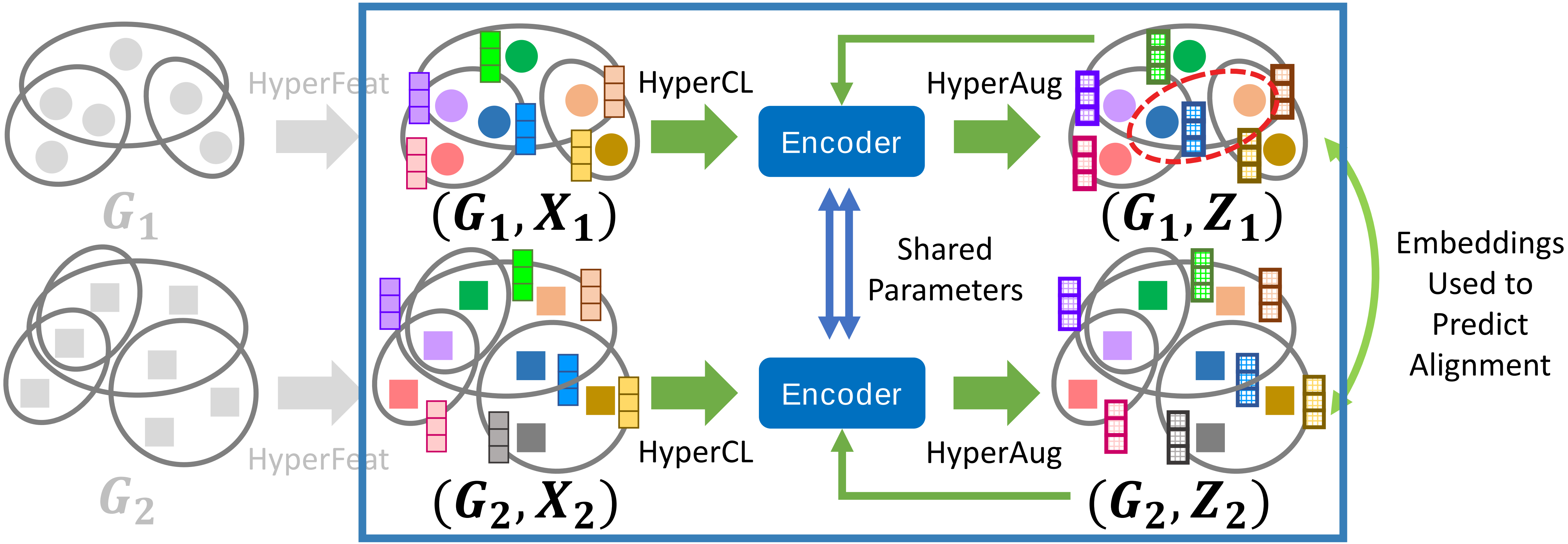
Step 2. HyperCL: Summary

- Contrastive learning serves as a “**pseudo” supervised alignment task**
- Note HyperCL is applied to **each hypergraph** to pretrain the same encoder



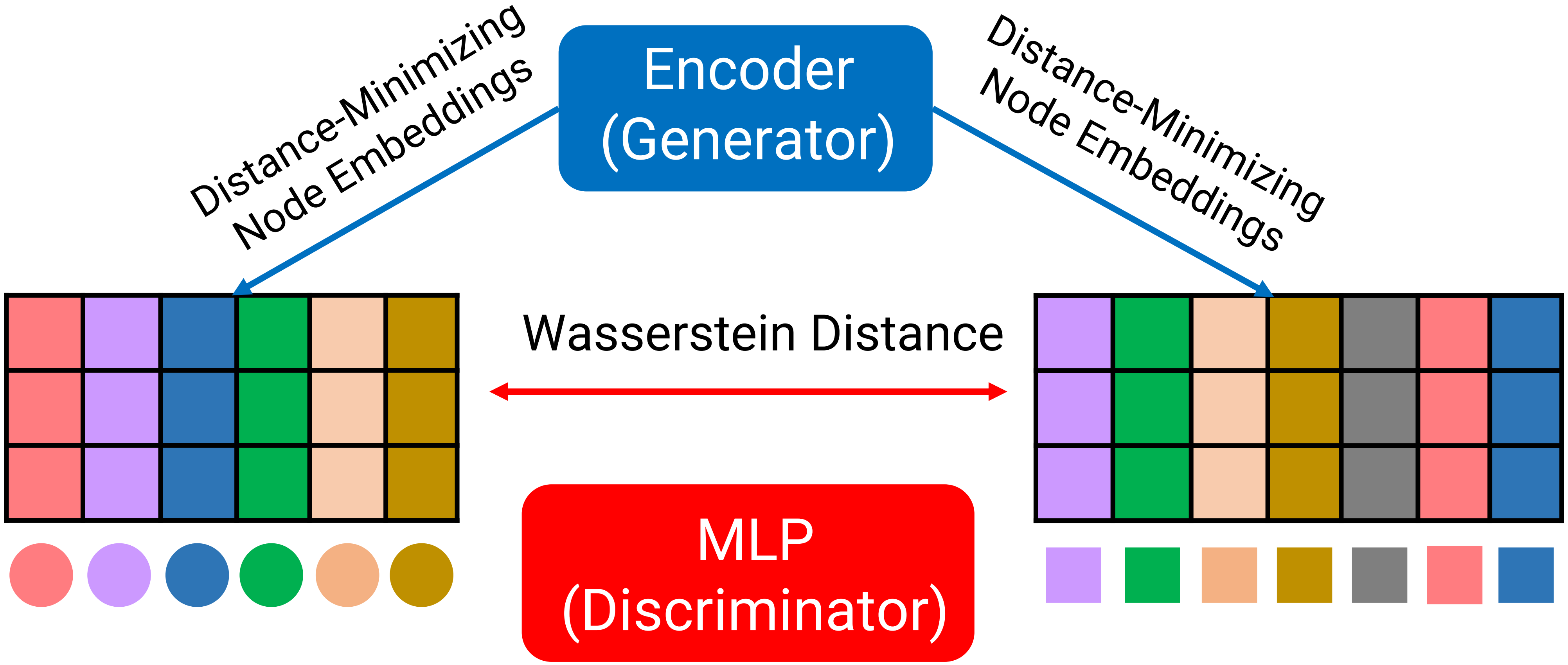
Step 3. HyperAug: Overview

- **Step 3. HyperAug:** adversarial learning with **topological augmentation**
  - Addressing Challenge 3, scale disparity of hypergraphs



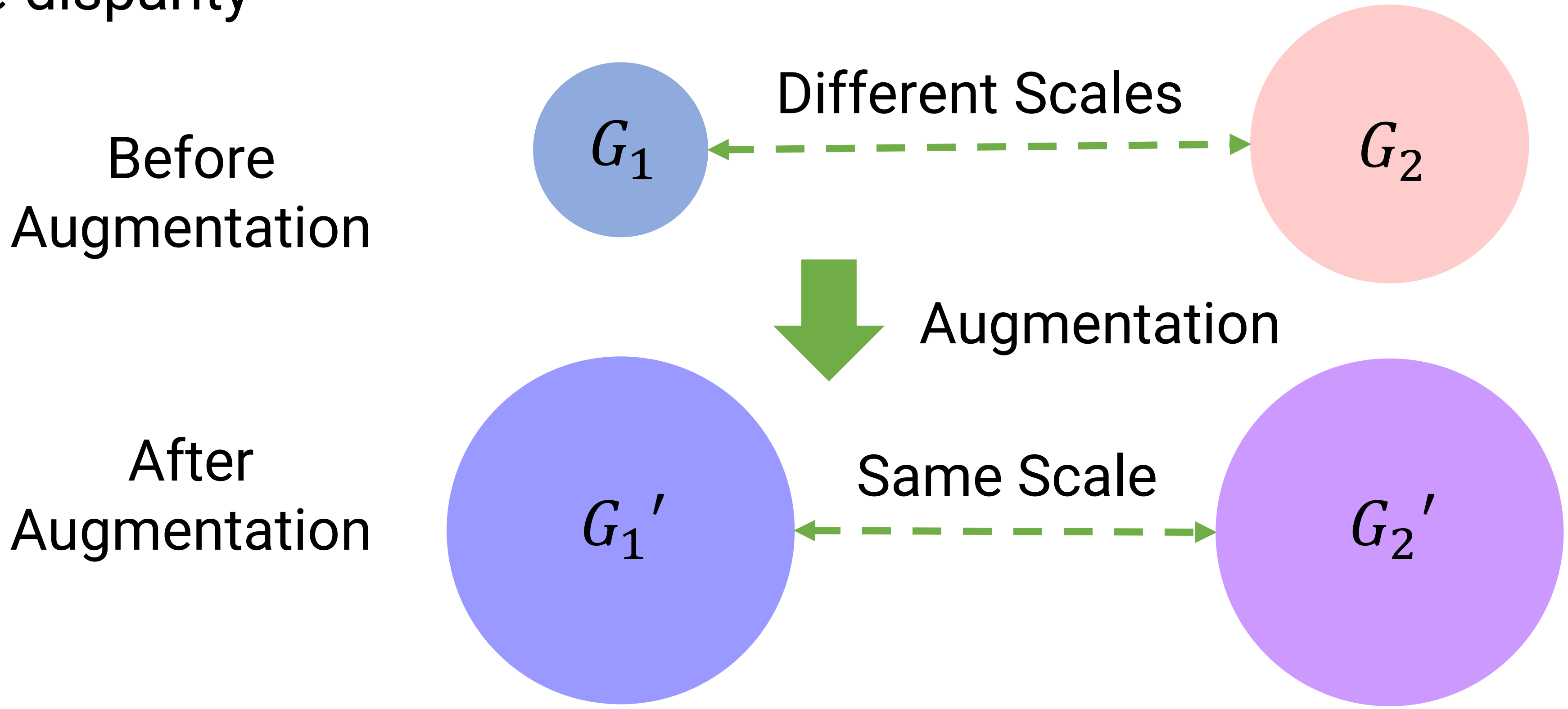
Step 3. HyperAug: (1) GAN Framework

- HyperAug employs generative adversarial networks (GAN) to **align two node embedding spaces**



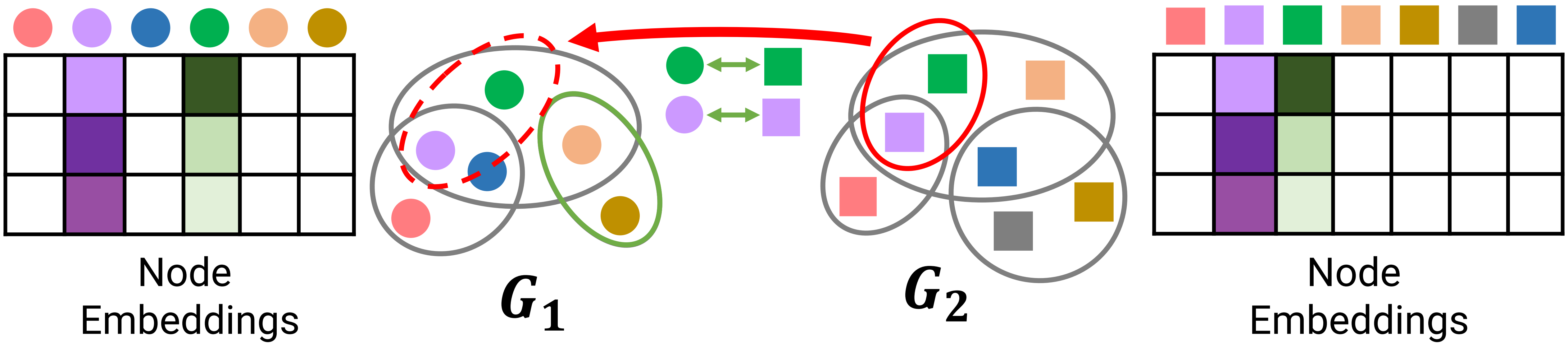
Step 3. HyperAug: (2) Hyperedge Augmentation

- **Given:** node embeddings in the current iteration of GAN training,
- **Augment:** hyperedges in both hypergraphs
- **to Resolve**: scale disparity



Step 3. HyperAug: Details

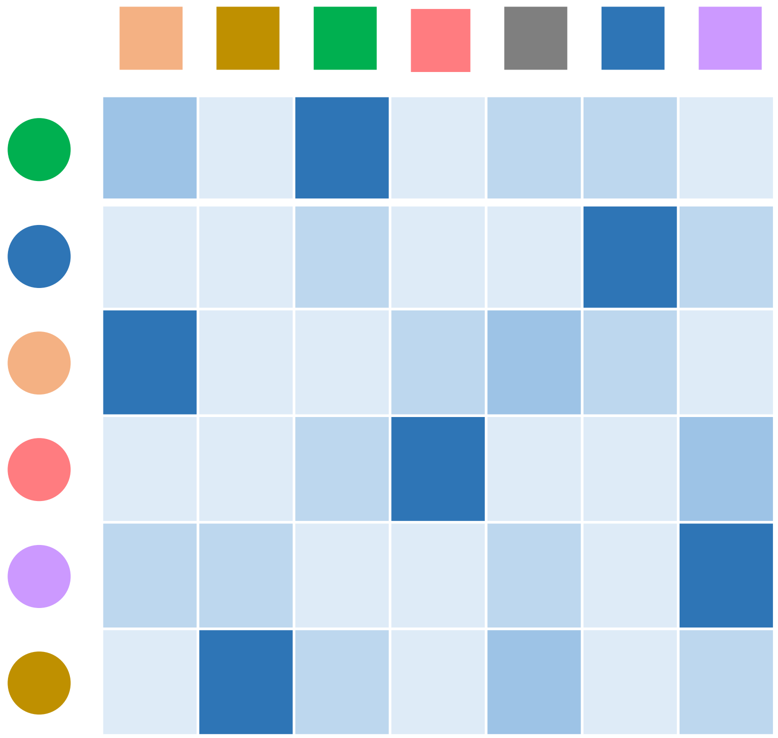
- For each hyperedge in one input, create a virtual hyperedge in the other
  - For each member, **find the most similar node** in the other hypergraph
    - We use the **node embeddings** in the current iteration
  - Construct **virtual hyperedges** containing the counter-part nodes



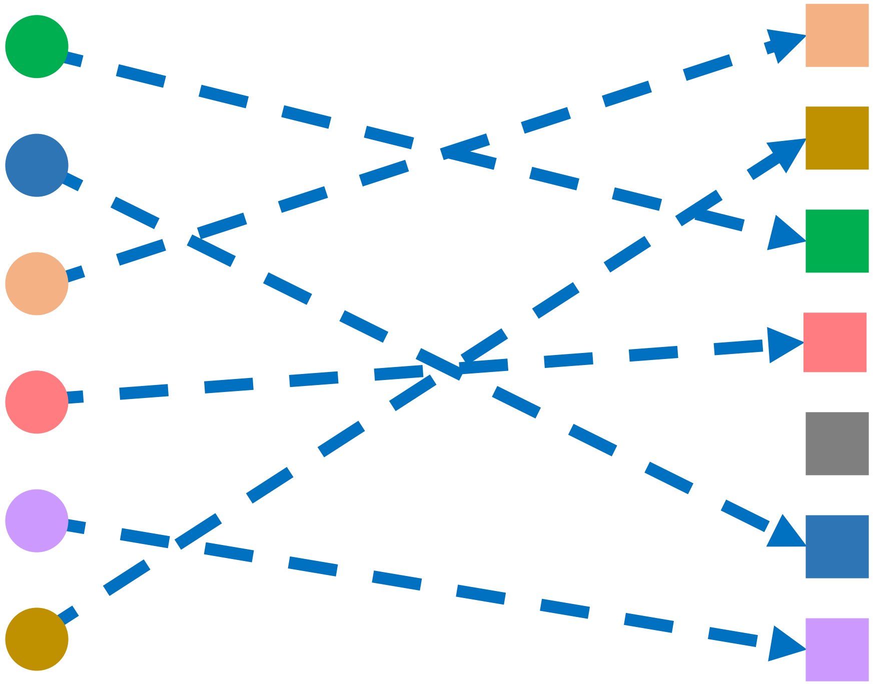
Last Step: Inferring Alignment based on Embeddings

- First, we **measure node similarity** based on embeddings
- Then, we **greedily match most similar nodes**

Node Similarity Based on Embeddings



Greedy Matching Based on Similarity



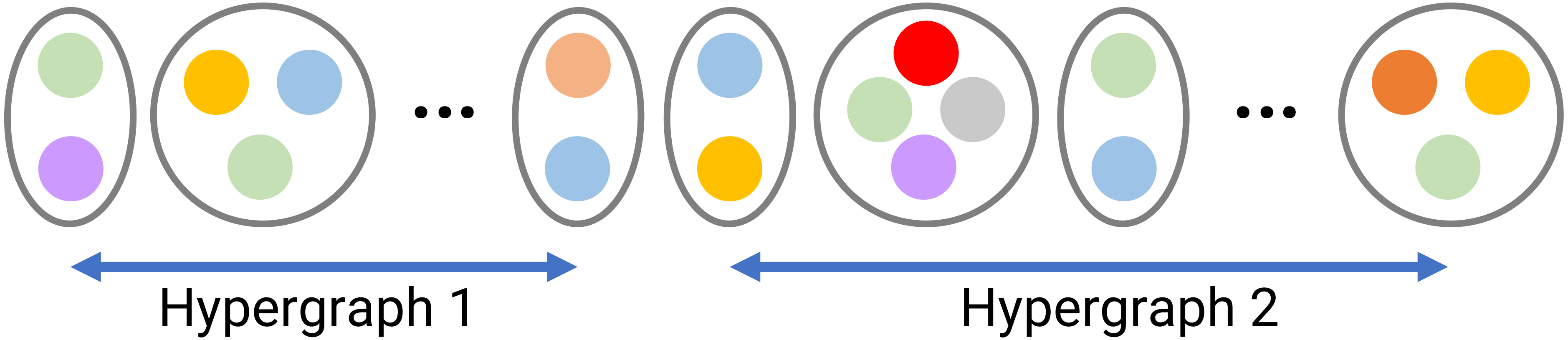
Roadmap

1. Introduction
2. Challenges
3. Proposed Method
4. **Experiments <<**
5. Conclusion

Experimental Settings

- **Datasets**: 12 real-worlds hypergraphs

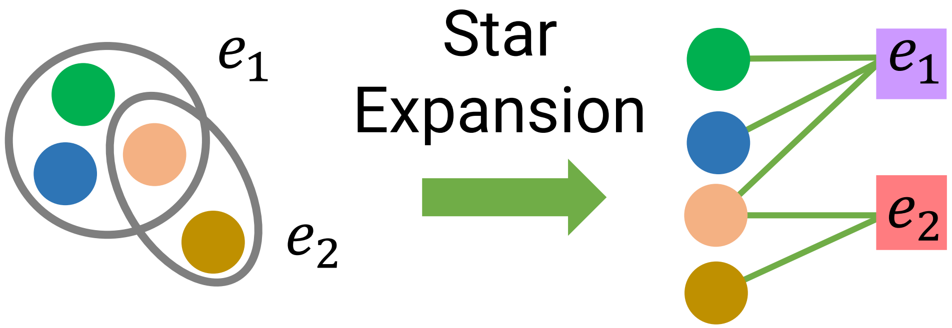
- **Preprocessing of each dataset:**
  - Two hypergraphs are from non-overlapping intervals of timestamps
  - One interval can be (much) longer than the other



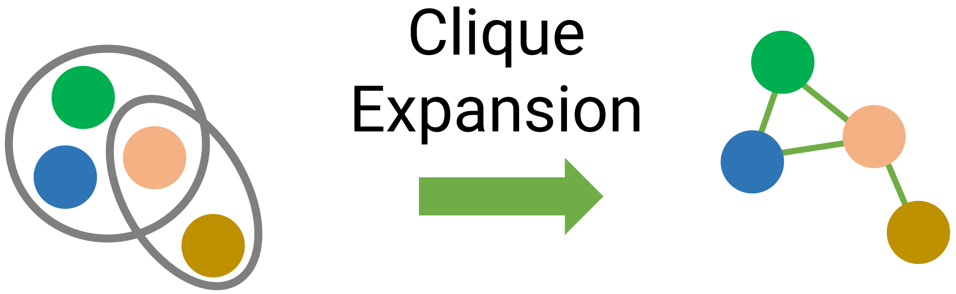
- **Performance**: % of correctly estimated pairs among the ground-truth ones

Baselines & Competitors

- **Bipartite graph** based methods:
  - Big-Align [Koutra et al. 2013]

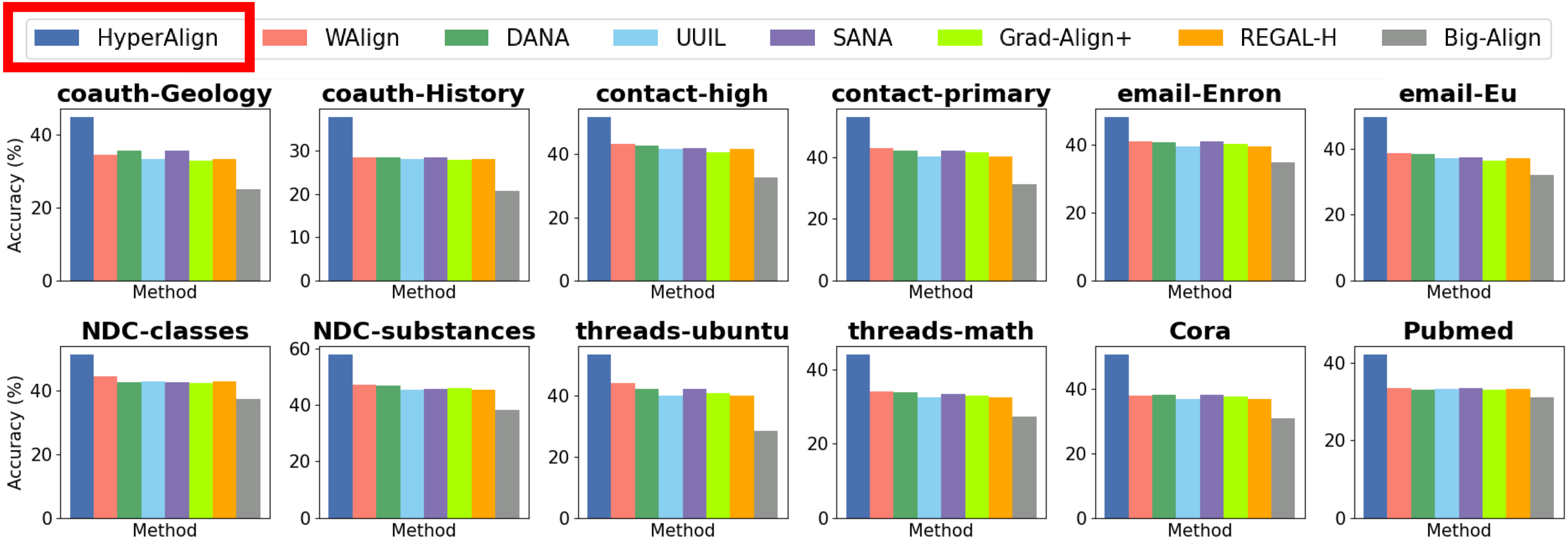


- **Unipartite graph** based methods:
  - Node embedding based: REGAL [Heimann et al. 2018]
  - Learning based: SANA [Peng et al. 2023] & Grad-Align+ [Park et al. 2022]
  - GAN based: UUIL [Li et al. 2018], DANA [Derr et al. 2021], WAlign [Gao et al. 2021]



Q1. Alignment Performance

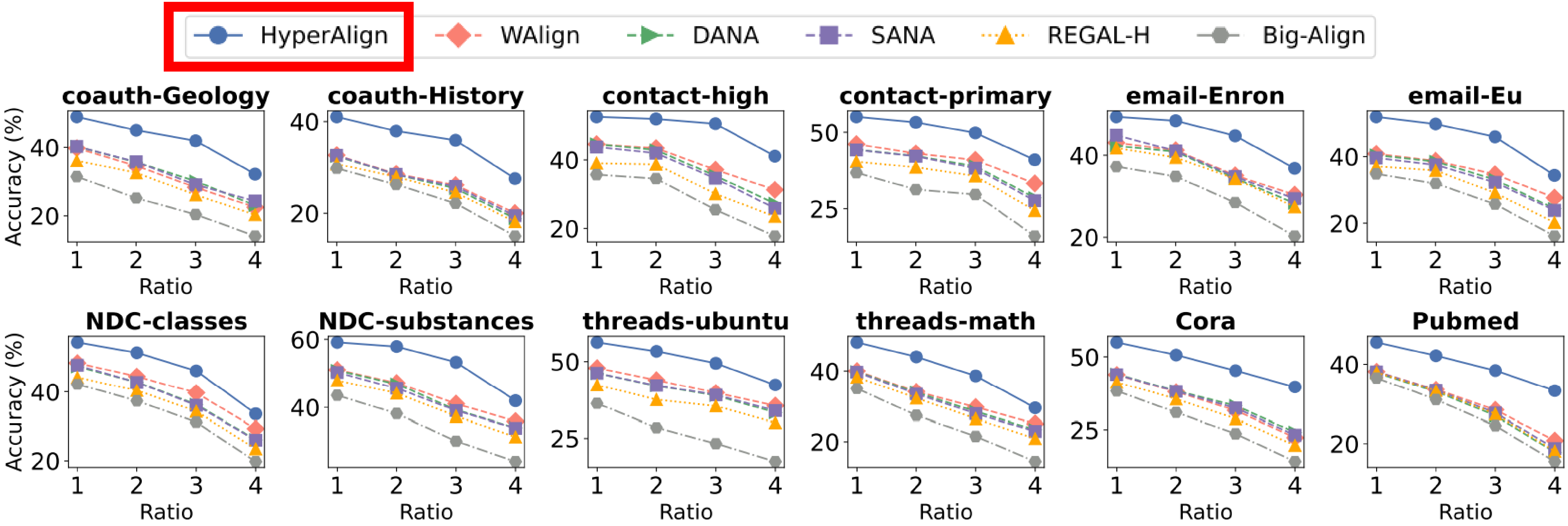
- **Q1.** How **accurate** is **HYPERALIGN**?
- **A1. HYPERALIGN** consistently outperforms all competitors in all datasets



Y-axis: Alignment Accuracy (the higher the better; scale disparity ratio = 0.5)

Q2. Scale Disparity Ratio of Two Input Hypergraphs

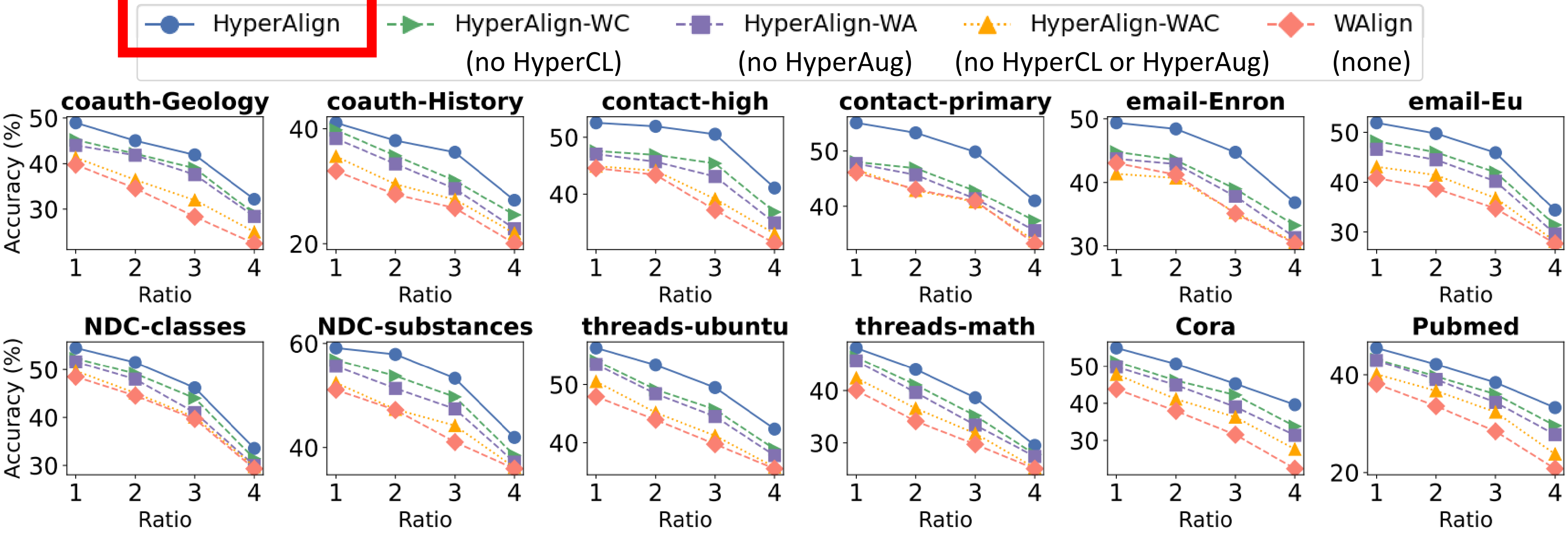
- **Q2.** How does the **scale disparity ratio** affect **HYPERALIGN**'s superiority?
- **A2. HYPERALIGN** is consistently superior across all disparity ratios



X-axis: Size Disparity Ratio & Y-axis: Alignment Accuracy

Q3. Ablation Studies

- **Q3.** Does each **component** of **HyperAlign** contribute to its performance?
- **A3.** Yes!



X-axis: Size Disparity Ratio & Y-axis: Alignment Accuracy

Roadmap

1. Introduction
2. Challenges
3. Proposed Method
4. Results
5. **Conclusion <<**

Conclusions: Our Contributions

- **New Problem:** unsupervised alignment of hypergraphs with scale disparity
- **Novel Method:** HyperAlign, addressing three challenges:
  - Challenge 1: Absence of node attribute
  - Challenge 2: Absence of supervision
  - Challenge 3: Scale disparity of two hypergraphs
- **Extensive Experiments:** we demonstrate the superiority HyperAlign

Code & Dataset: https://github.com/manhtuando97/HyperAlign

Acknowledgements

- **HyperAlign** stands on "the shoulders of giants"
- **Feature Extraction in Graph:**
  - Struc2Vec [Ribeiro et al. 2017]
- **GAN-Based Alignment of Graphs**
  - DANA [Derr et al. 2021], WAlign [Gao et al. 2021]
- **Contrastive Learning on Graphs**
  - GraphCL [You et al. 2020]