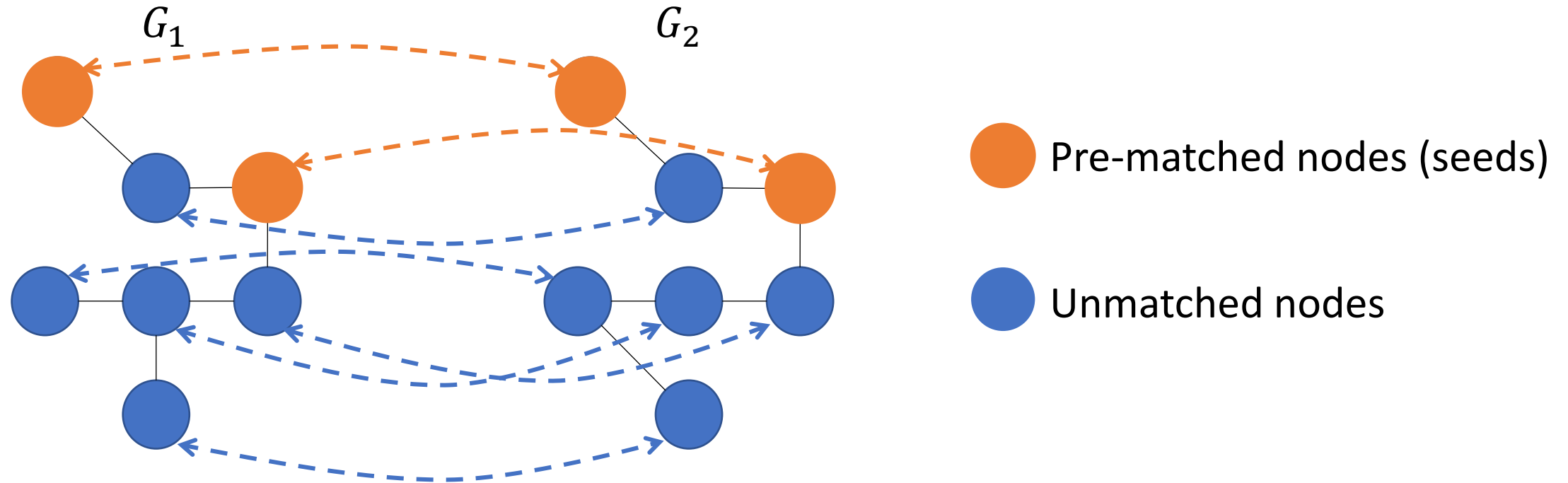


SeedGNN: Graph Neural Network for Supervised Seeded Graph Matching

Liren Yu, Purdue University

Joint work with Prof. Jiaming Xu (Duke University) and Prof. Xiaojun Lin (Purdue University)

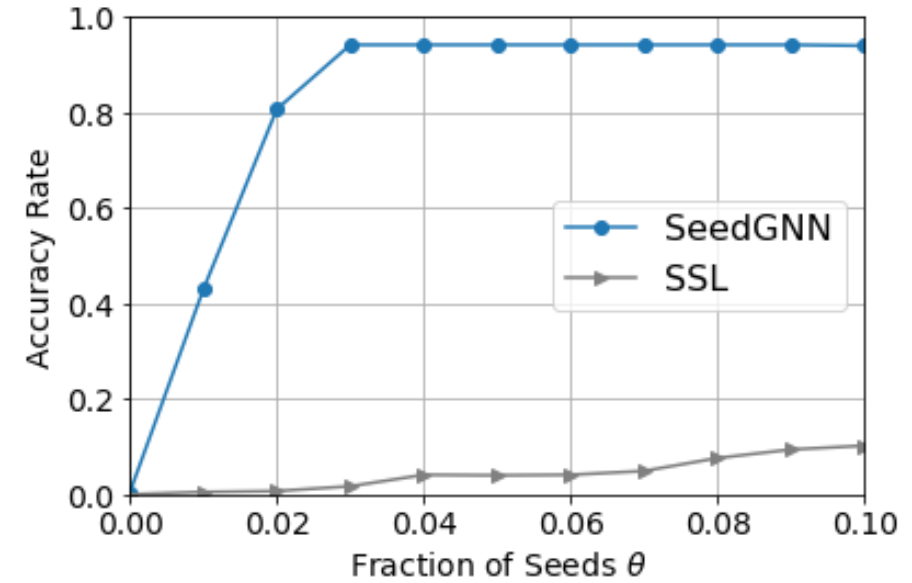
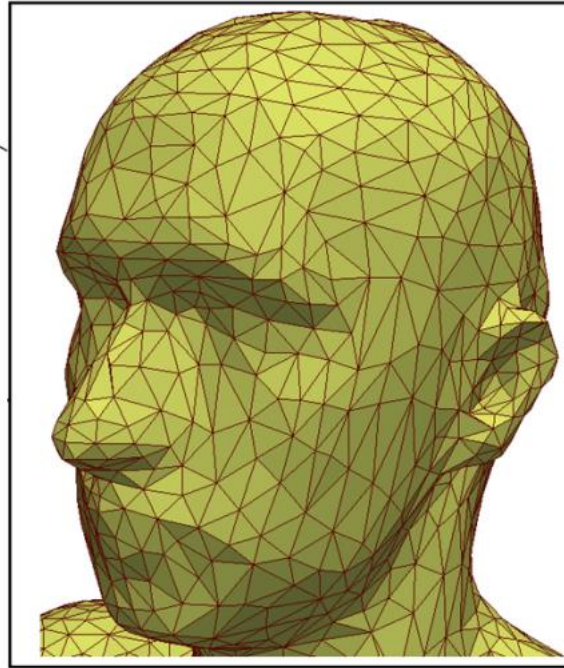
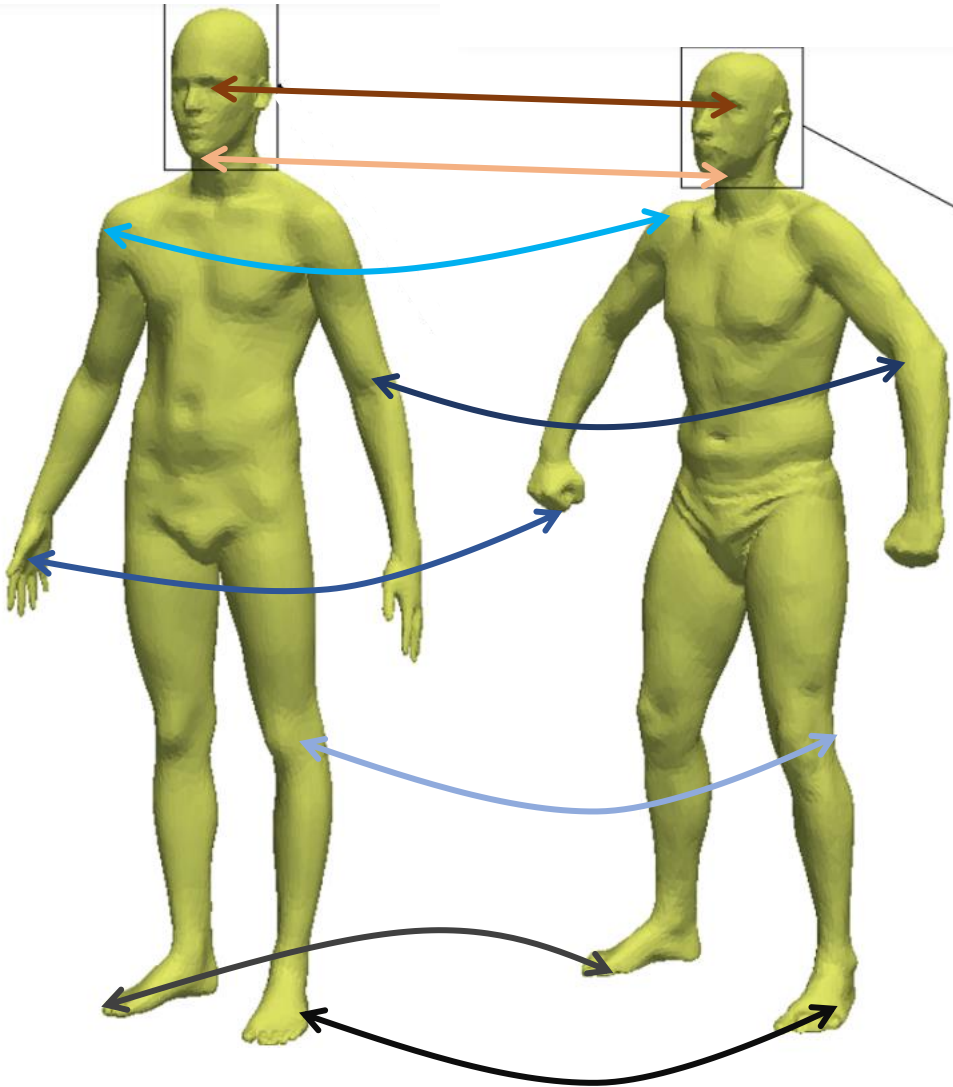
Problem: seeded graph matching



Goal: find the mapping between unmatched nodes using seeds

Applications: computer vision, social network de-anonymization, computational biology, and natural language processing ...

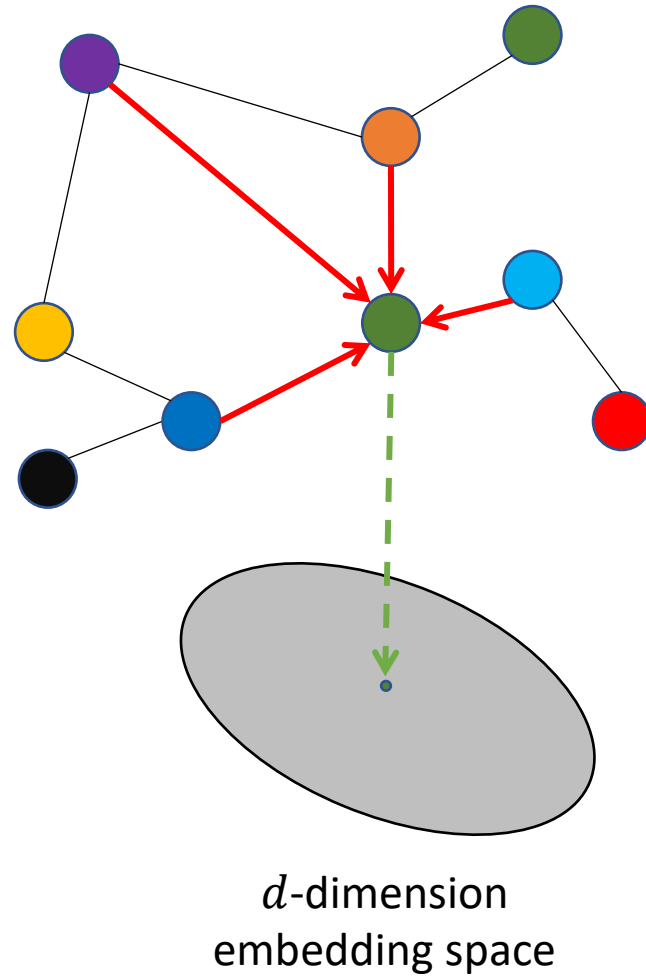
Application: matching 3D scans of object surface



With a small fraction of seeds,

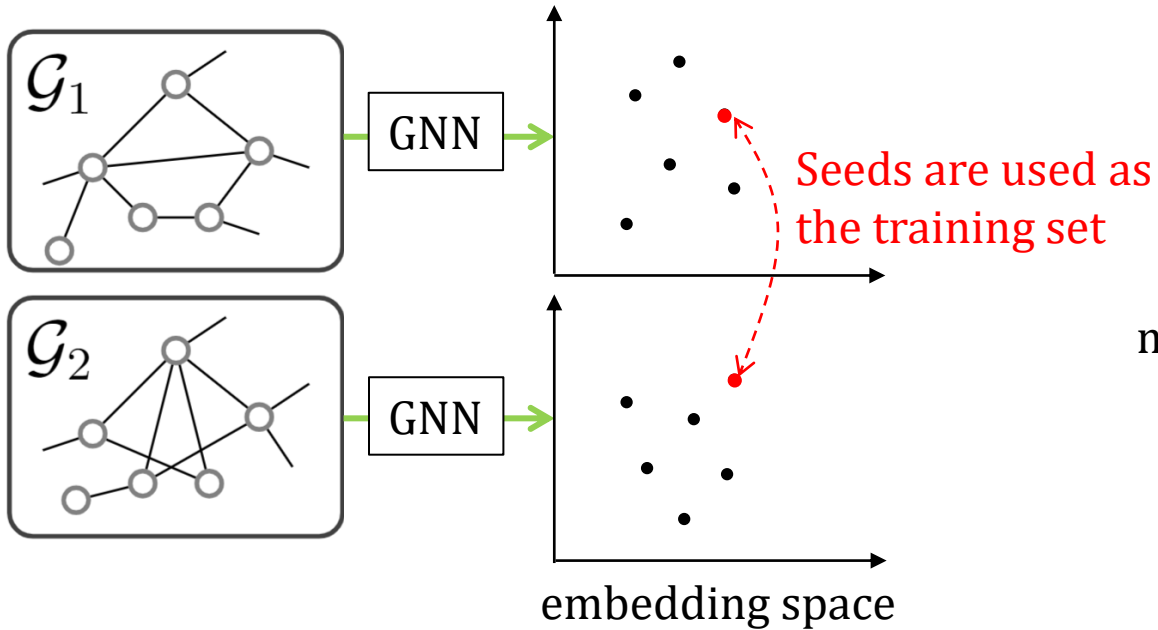
- Our new SeedGNN can match nearly all nodes
- The existing SSL (semi-supervised learning) GNN almost completely fails

Graph Neural Networks (GNNs)



GNNs for seeded graph matching

Semi-supervised learning (SSL)



- Require many seeds

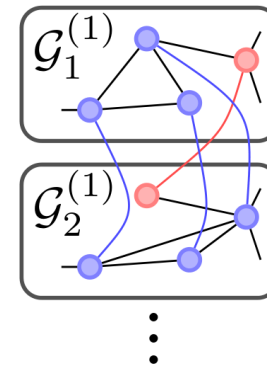
Supervised learning (SL)

- Seedless ✓
- Seeded ✗

Our Method: SeedGNN

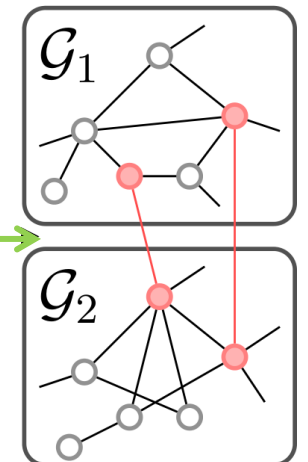
- Supervised learning
- Learn to use seeds

Training set:
matched graphs



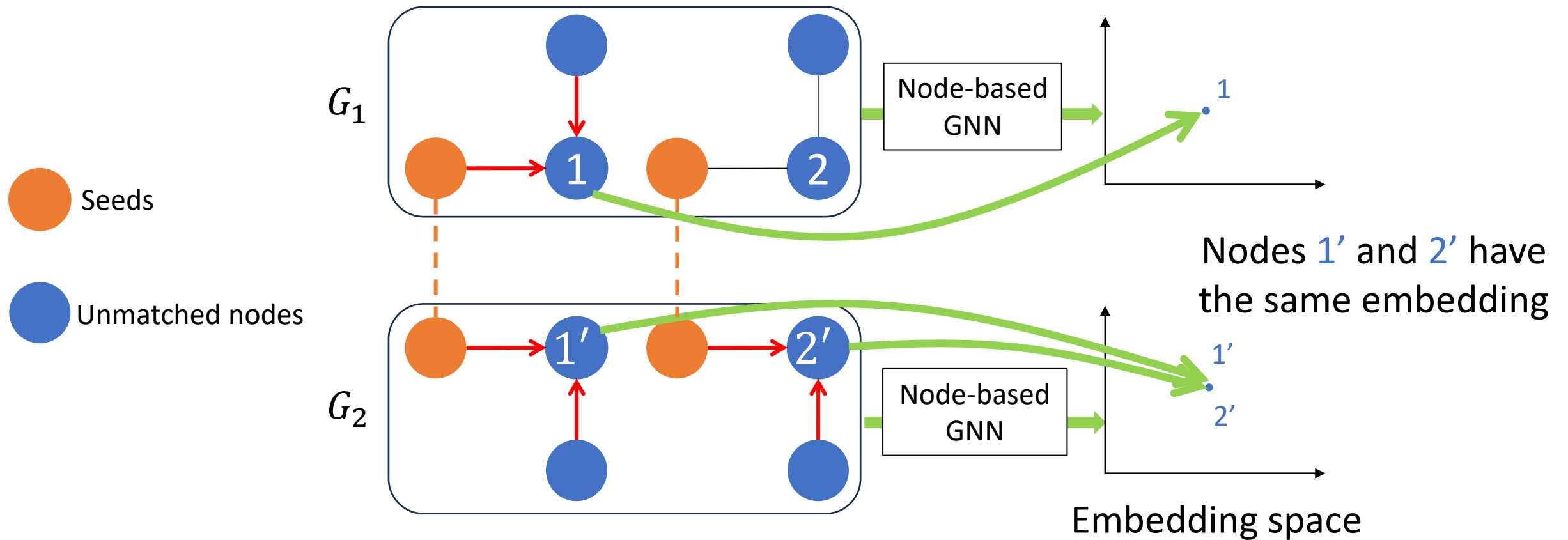
SeedGNN (with
transferable
knowledge)

Test set:
unseen graphs



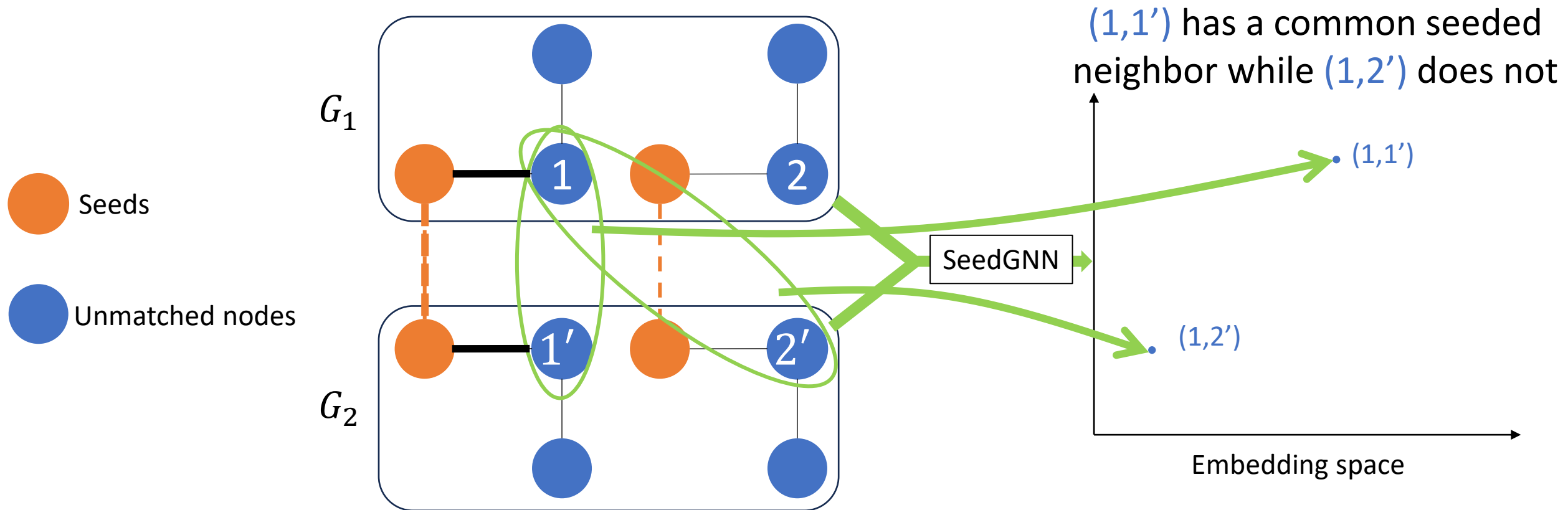
Key Idea 1: pair-wise GNN instead of node-based GNN

- Node-based GNNs fail to use seed information to distinguish nodes



Key Idea 1: pair-wise GNN instead of node-based GNN

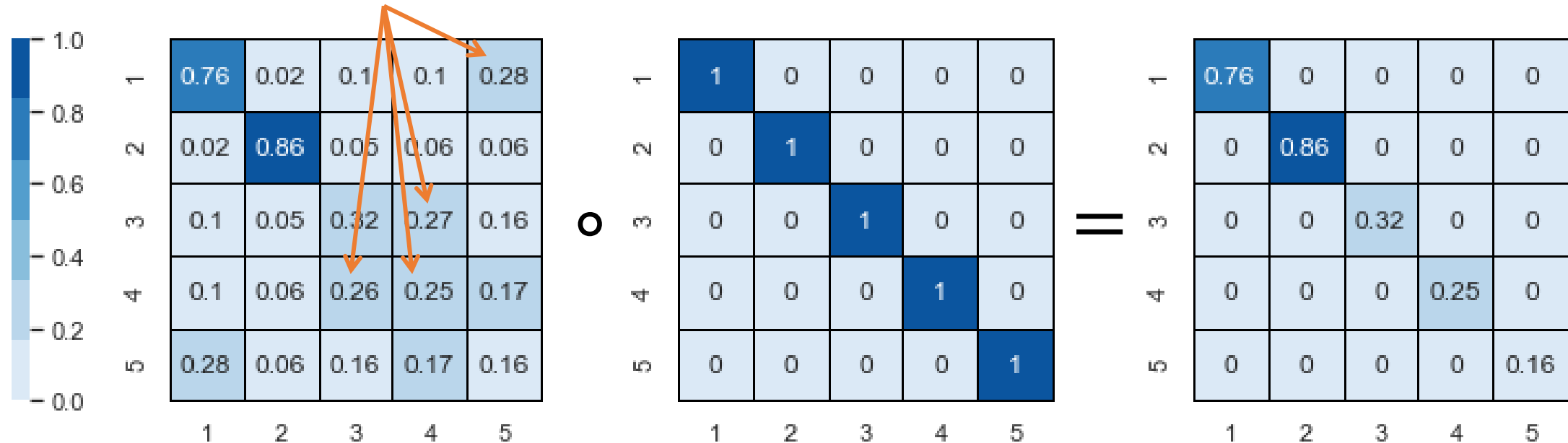
- Node-based GNNs fail to use seed information to distinguish nodes
- SeedGNN applied on node-pairs across graphs can easily utilize the seeds



Key Idea 2: masking to enable successful percolation

- Percolation (i.e., using newly matched pairs as new seeds) is crucial for seeded graph matching
- However, the new seeds found may be noisy (fake pairs)

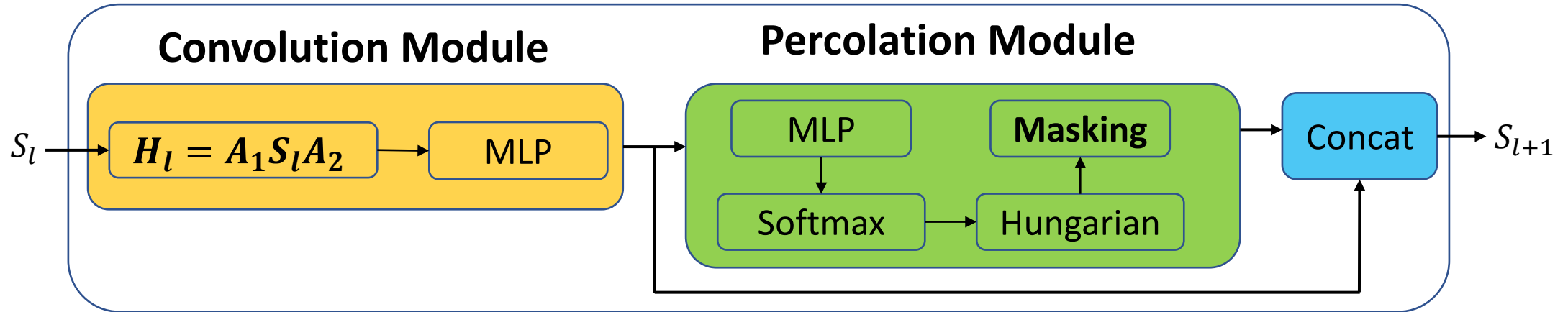
Fake pairs (off diagonal) may have comparable similarity with true pairs (on diagonal)



- Use the Hungarian algorithm to filter out the noisy information

Architecture

The l -layer of SeedGNN



- Convolution module aggregates neighboring information near the node pair
- Percolation module filters out noisy information
- Combining these two modules allows adaptive feature choosing