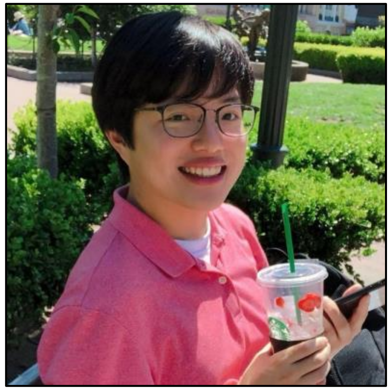
HashNWalk: Hash and Random Walk Based Anomaly Detection in Hyperedge Streams

Geon Lee

Minyoung Choe



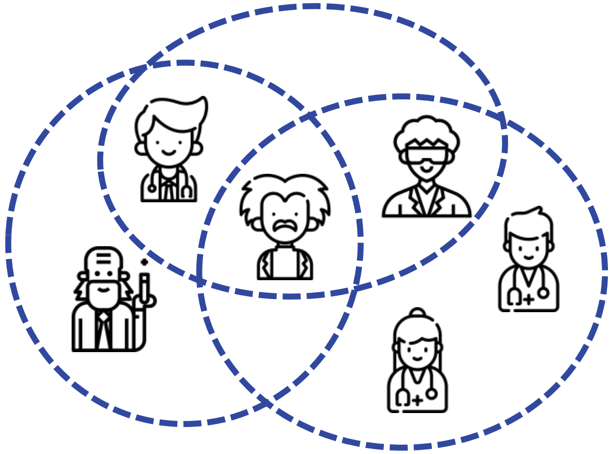
Kijung Shin

Overview

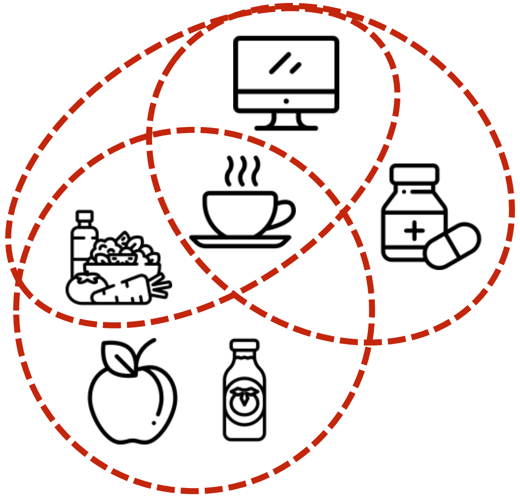
1. Introduction
2. Backgrounds
3. Algorithms
4. Experiments
5. Conclusion

Hypergraphs are Everywhere

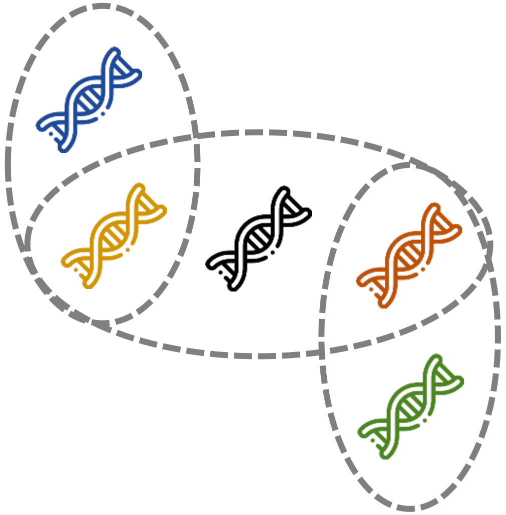
- **Hypergraphs** consist of nodes and hyperedges.
- Each **hyperedge** is a subset of any number of nodes.



Collaborations of Researchers



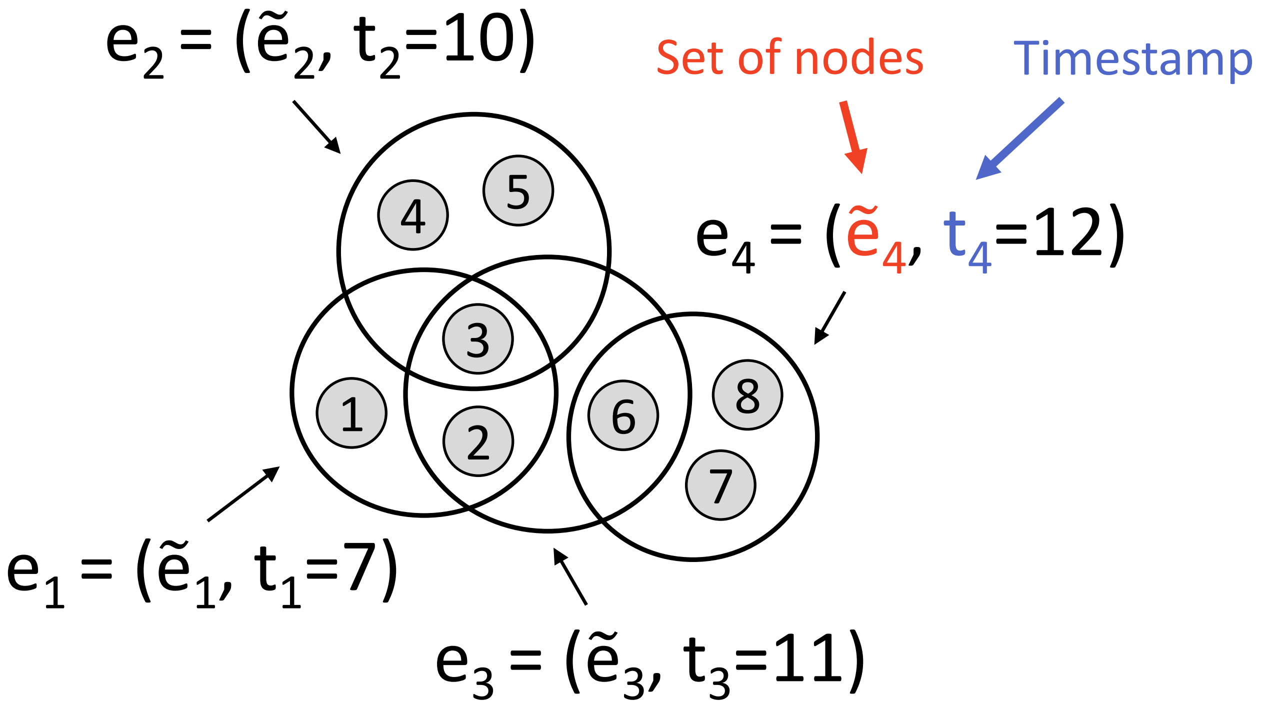
Co-purchases of Items



Joint Interactions of Proteins

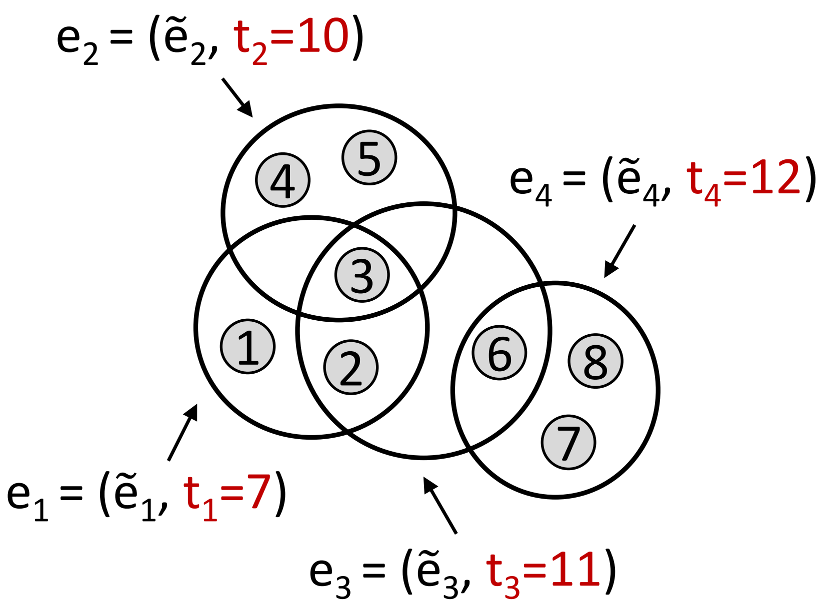
Hypergraphs Evolve Over Time

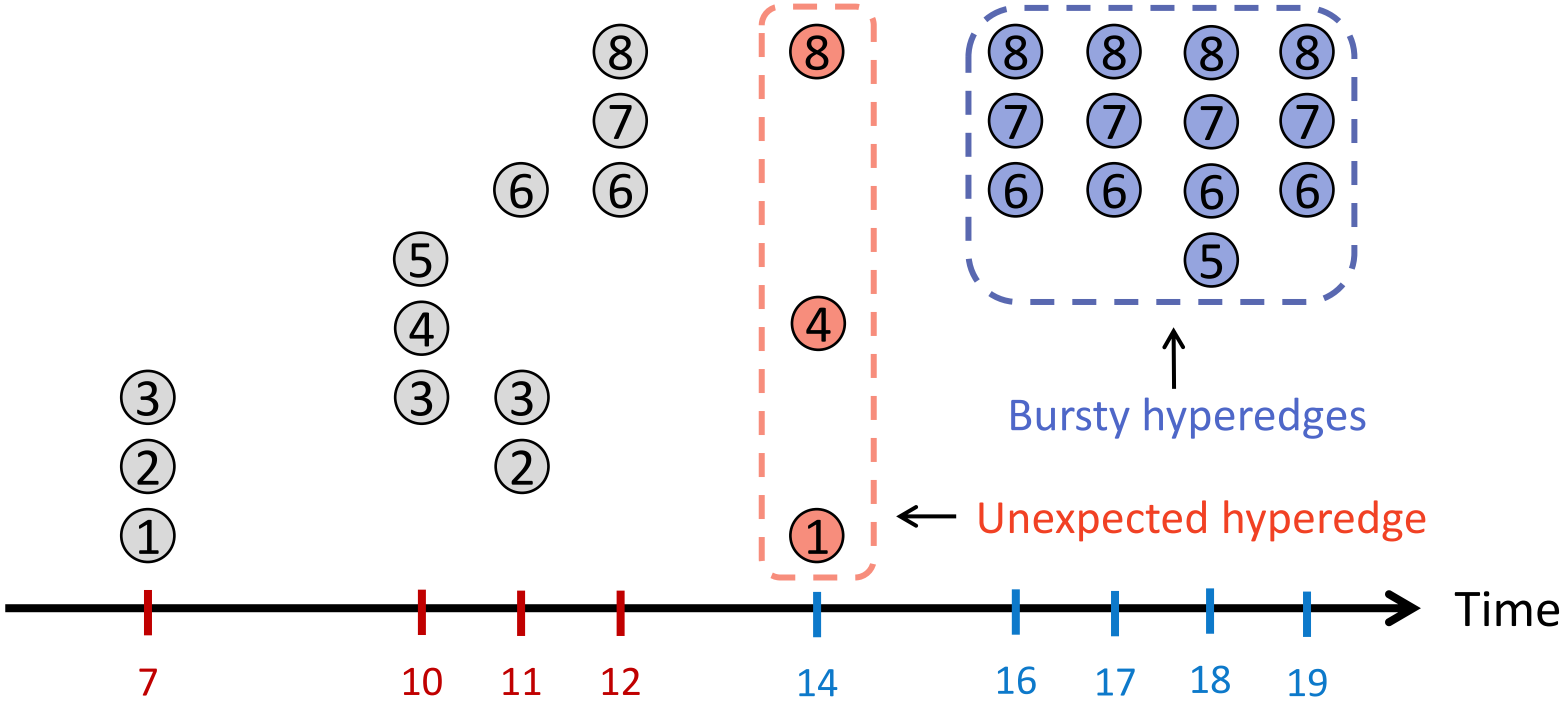
- In many real-world scenarios, hypergraphs **evolve over time**.
- A **hyperedge stream** $\{(e_i, t_i)\}_{i=1}^{\infty}$ is a sequence of hyperedges.



Anomalies in Hypergraphs

- We focus on two intuitive aspects: **unexpectedness** and **burstiness**.
  - **Unexpected hyperedges** consist of unnatural combinations of nodes.
  - **Bursty hyperedges** repeat in a short period of time.





Problem Definition

- We formalize the **hyperedge anomaly detection** problem as follow:

Given a stream $\mathcal{E} = \{(e_i, t_i)\}_{i=1}^{\infty}$ of hyperedges, detect anomalous hyperedges, whose **structural** or **temporal** properties deviate from general patterns, in **near real-time** using **constant space**.

Hypergraph Random Walk

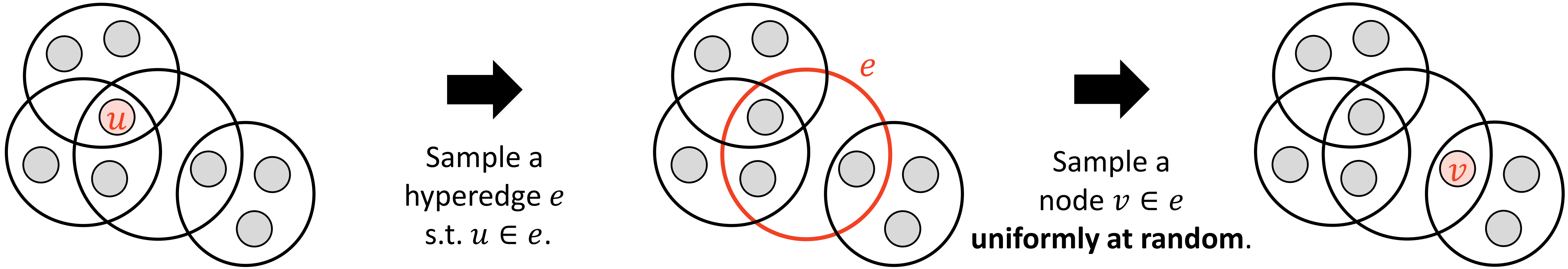
- Typically, a **random walk on a hypergraph G** is formulated as:

If the current node is u,

① Select a hyperedge e that contains node u (i.e., $u \in e$) with probability proportional to the weight $\omega(e)$.

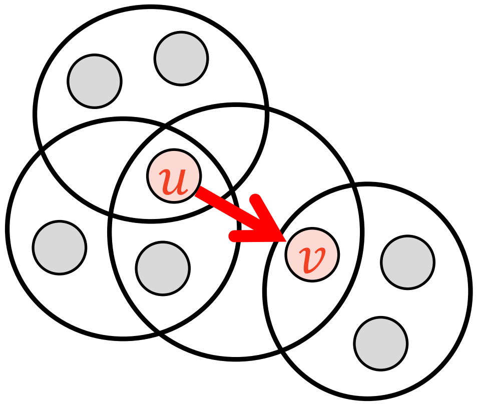
② Select a node $v \in e$ with probability **uniformly at random**.

③ Walk to node v.

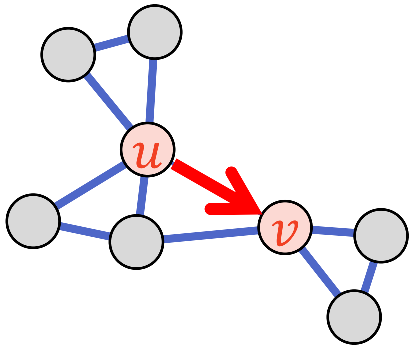


Hypergraph Random Walk (cont.)

- However, this is **equivalent** to the random walks on **clique expansion**.
- Clique expansion suffers from the **loss of information** on high-order interactions.



=



Random walk on a **hypergraph**

Random walk on a **clique expansion**

Hypergraph Random Walk (cont.)

- **Edge-dependent vertex weight**-based random walk is designed as:

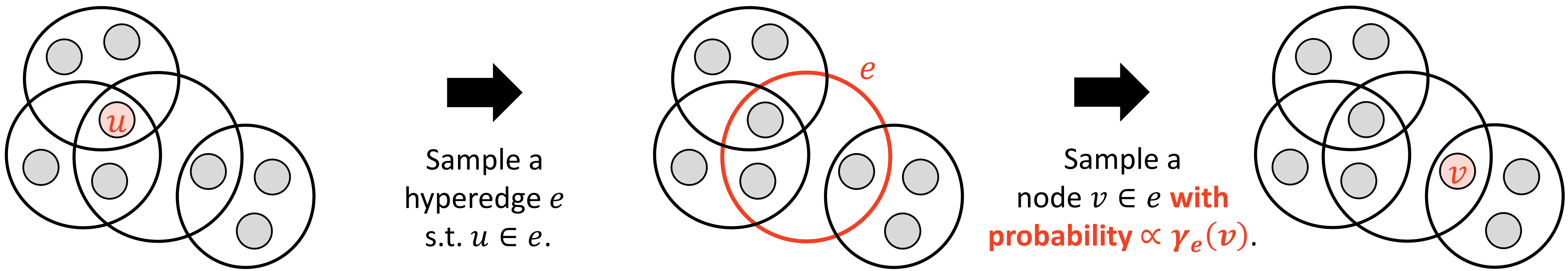
If the current node is u,

① Select a hyperedge e that contains node u (i.e., $u \in e$) with probability proportional to the weight $\omega(e)$.

② Select a node $v \in e$ with probability ~~uniformly at random~~.

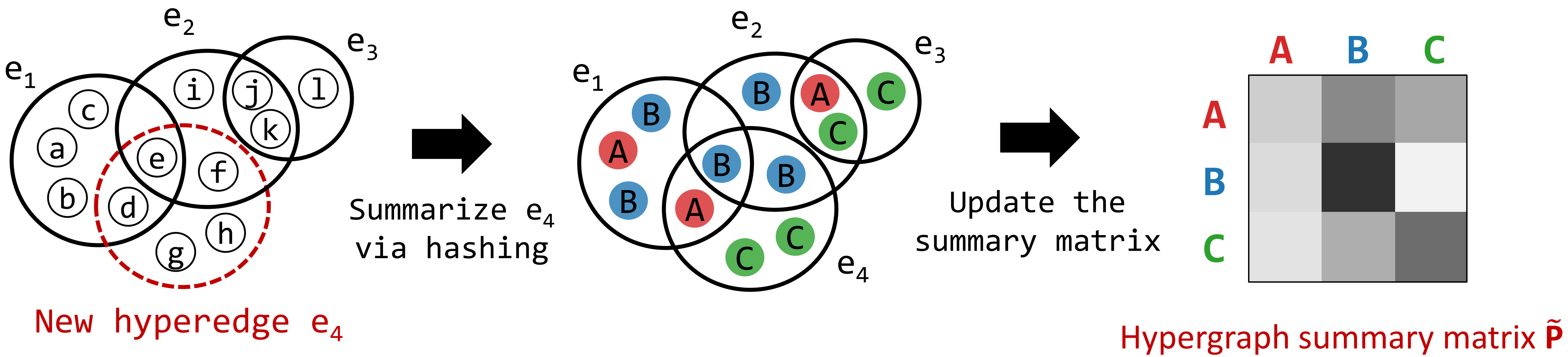
Proportional to the **edge-dependent vertex weight** $\gamma_e(v)$

③ Walk to node v.



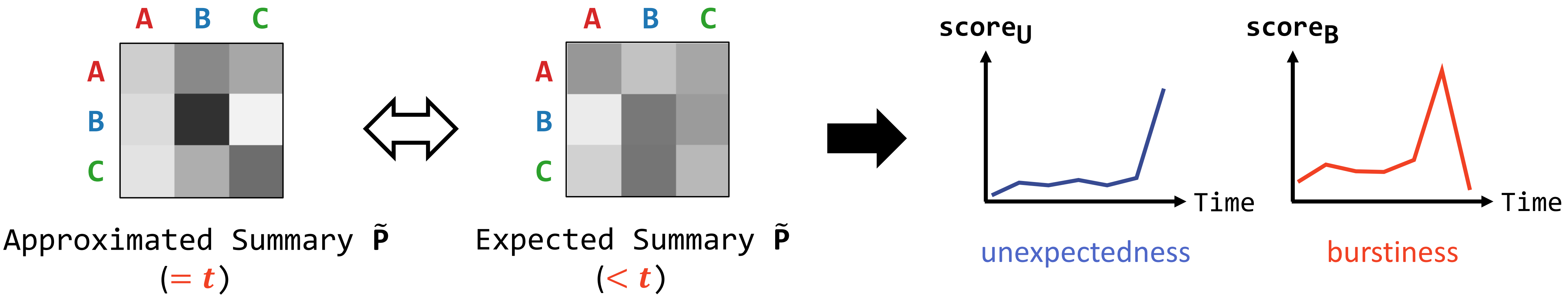
HashNWalk

- We propose **HashNWalk**, a **fast** and **space-efficient** algorithm for detecting anomalies in a hyperedge stream.
- We maintain a hypergraph summary matrix $\tilde{\mathbf{P}}$ where $\tilde{\mathbf{P}}_{A,B}$ is the random walk transition probability from supernode A to supernode B.



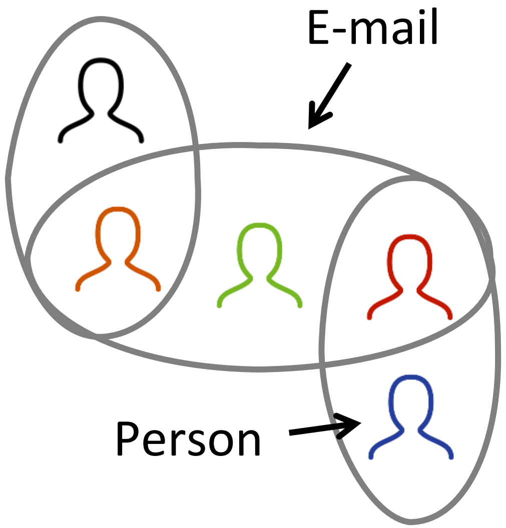
HashNWalk (cont.)

- Once the hypergraph summary $\tilde{\mathbf{P}}$ is updated at time t, it is compared with the previous summary ($< t$).
- We define scoring functions $\mathbf{score_U}$ and $\mathbf{score_B}$ to detect unexpected and bursty hyperedges, respectively.

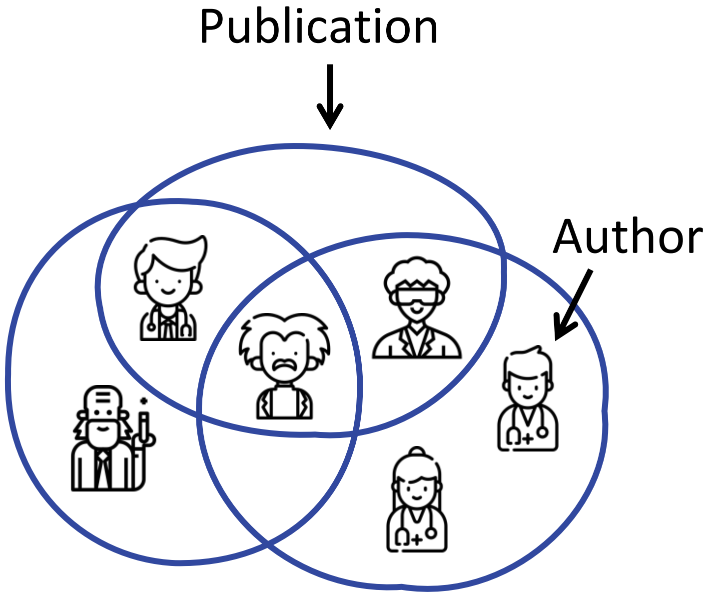


Experimental Settings

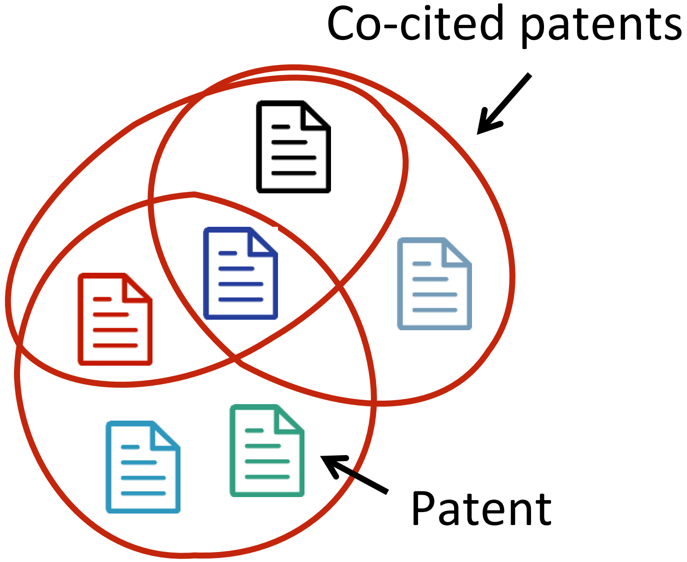
- We use various real-world and semi-real hypergraphs to evaluate **HashNWalk**.



email-Enron dataset



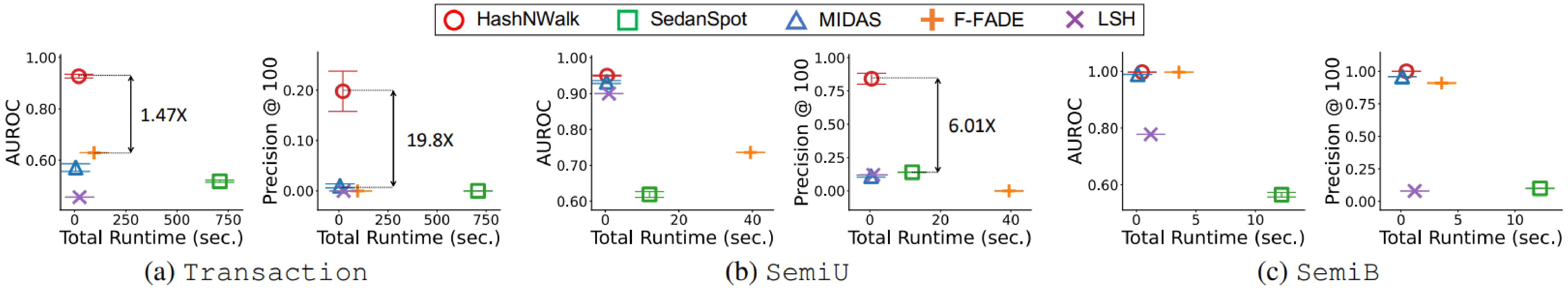
DBLP dataset



cite-patent dataset

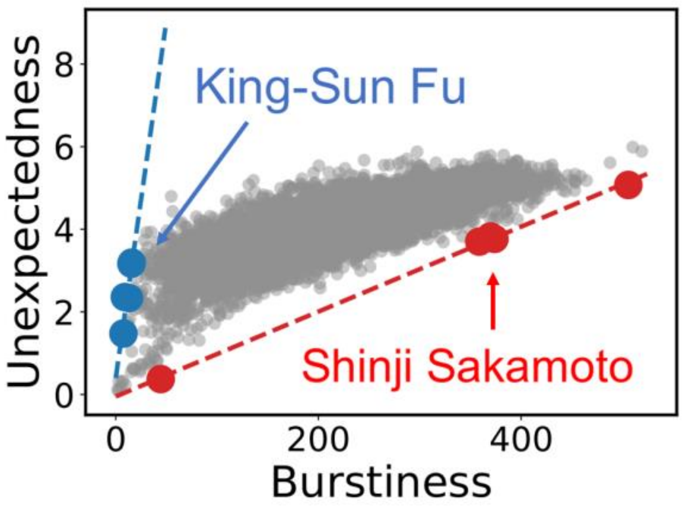
Experimental Results

- **HashNWalk** is **accurate** and **fast**.
  - 3 datasets: Transaction (real-world), SemiU (semi-real), and SemiB (semi-real)
  - 4 competitors: SedanSpot, MIDAS, F-FADE, and LSH

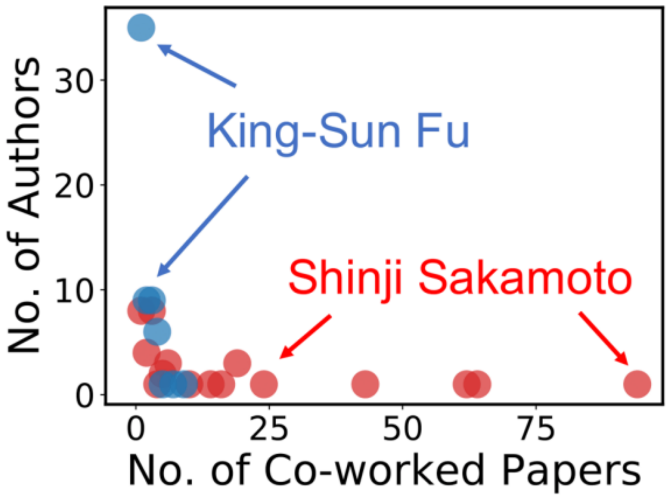


Experimental Results (cont.)

- Case study in **DBLP hypergraph**
  - **HashNWalk** captures different co-working styles of researchers.



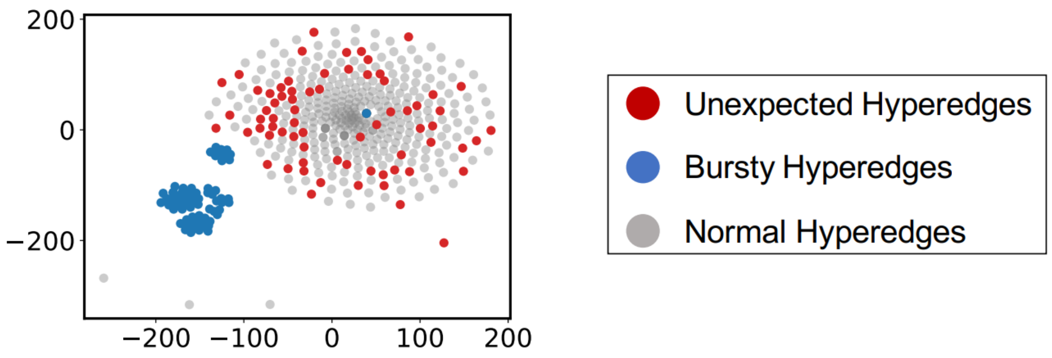
Some authors deviate from the general pattern.



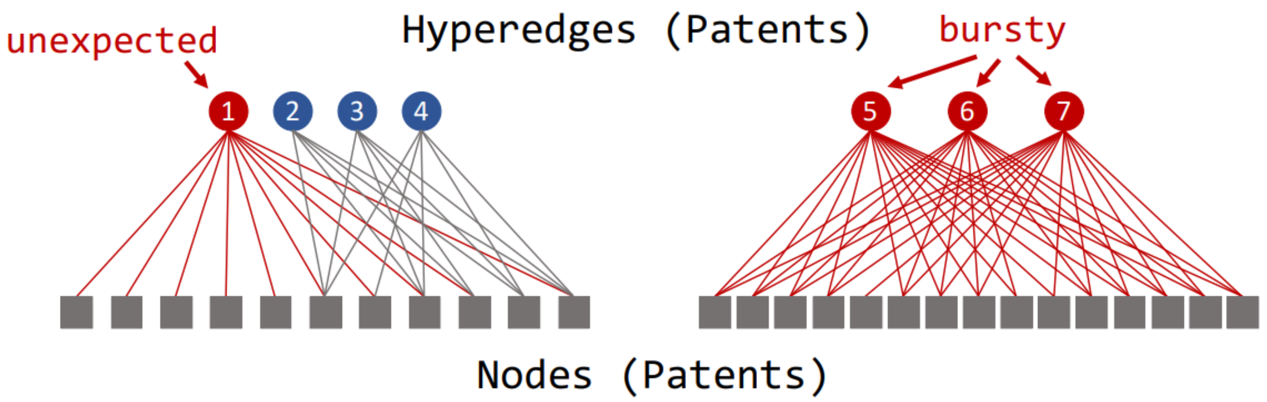
Dr. Fu and Dr. Sakamoto differ in their co-working patterns.

Experimental Results (cont.)

- Case study in **cite-patent hypergraph**
  - **HashNWalk** captures anomalous patents.



Unexpected & bursty hyperedges have different properties.



Patent 1 cited multiple patents that have not been cited together before.
Patents 5 – 7 cited almost the same set of patents.

Conclusion

- We propose **HashNWalk** an online anomaly detector for hyperedge streams.

HashNWalk is:

✓ **Fast**: It takes near real-time to process each new hyperedge.

✓ **Space Efficient**: The size of the hypergraph summary is a predefined constant.

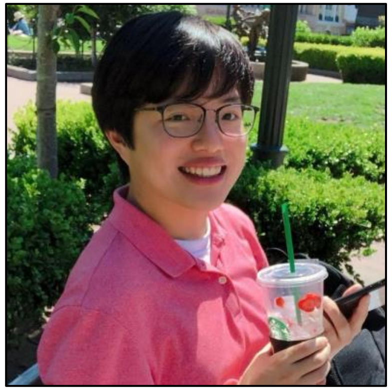
✓ **Accurate**: It successfully detects anomalous hyperedges in real-world hypergraphs.

Code & datasets: https://github.com/geonlee0325/HashNWalk

HashNWalk: Hash and Random Walk Based Anomaly Detection in Hyperedge Streams

Geon Lee

Minyoung Choe



Kijung Shin