Mining of Real-world Hypergraphs: Concepts, Patterns, and Generators

Part 3. Generative Models

Geon Lee

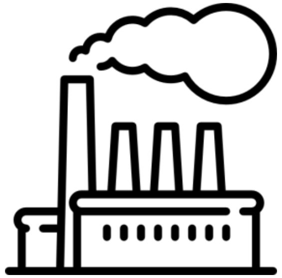
Jaemin Yoo

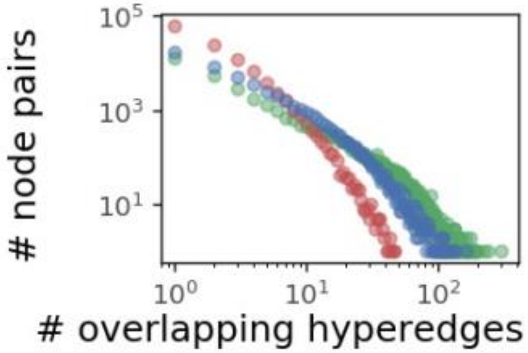
Kijung Shin

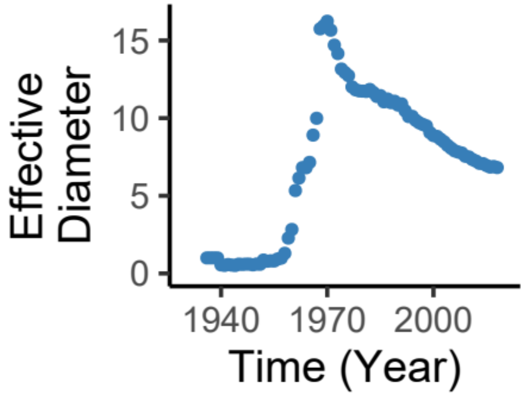
Part 3. Generative Models

*"How can we generate **realistic hypergraphs**?"*

*"What are **underlying mechanisms** that lead to the observed patterns?"*







Generative models

Static Graphs (Part 3-1)

Dynamic Graphs (Part 3-2)

Why Generative Models?

- **Explaining Patterns:** Shed a light on why patterns occur
- **Statistical Test:** Test the statistical significance of patterns
- **Benchmark Data Generation**
  - Create large realistic hypergraphs for evaluation of hypergraph algorithms
  - Useful when real hypergraphs are hard/impossible to collect
- **Anonymization**
  - Generate and publicize synthetic hypergraphs to structurally-similar real ones
  - Useful real hypergraphs cannot be publicized (due to sensitive information etc.)

Roadmap

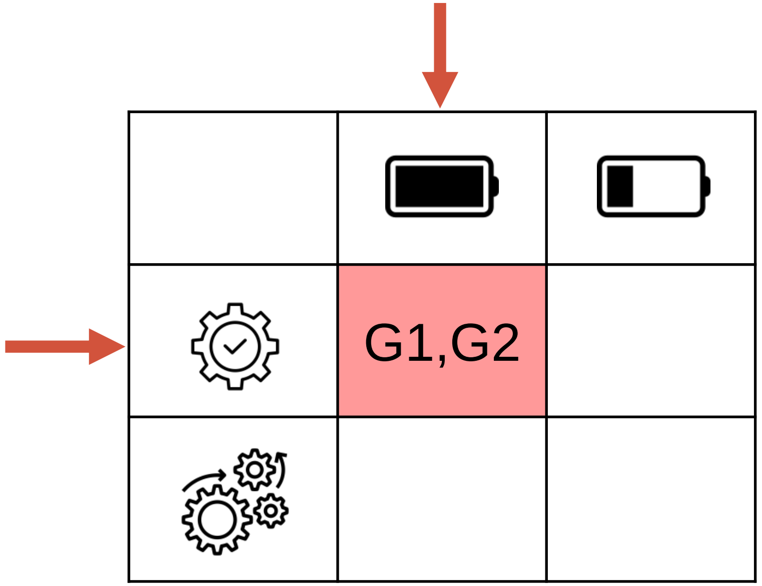
- **Part 1. Static Structural Patterns**
  - Basic Patterns
  - Advanced Patterns
- **Part 2. Dynamic Structural Patterns**
  - Basic Patterns
  - Advanced Patterns
- **Part 3. Generative Models**
  - Static Hypergraph Generator **<<**
  - Dynamic Hypergraph Generator

Part 3-1. Static Hypergraph Generative Models

		Part 3. Generative Models
Static Models	Full-Hypergraphs	C20, LCS21
	Sub-Hypergraphs	CYLBKS22
Dynamic Models	Full-Hypergraphs	DYHS20, KKS20, KBCYS23, GLLB23
	Sub-Hypergraphs	BKT18, CK21

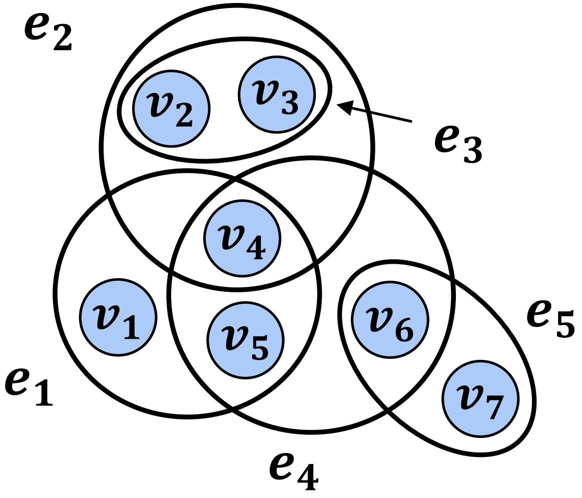
C20: Configuration Models

- **G1:** Pairwise reshuffling

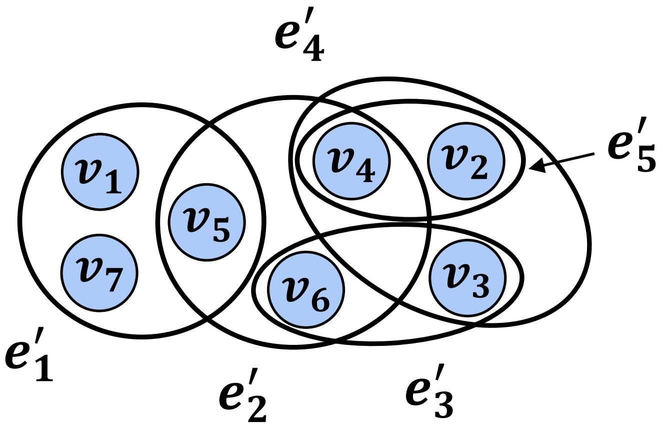


Configuration Models

- **Configuration models** generate random hypergraphs while preserving distributions of **node degrees** and **hyperedge sizes**.



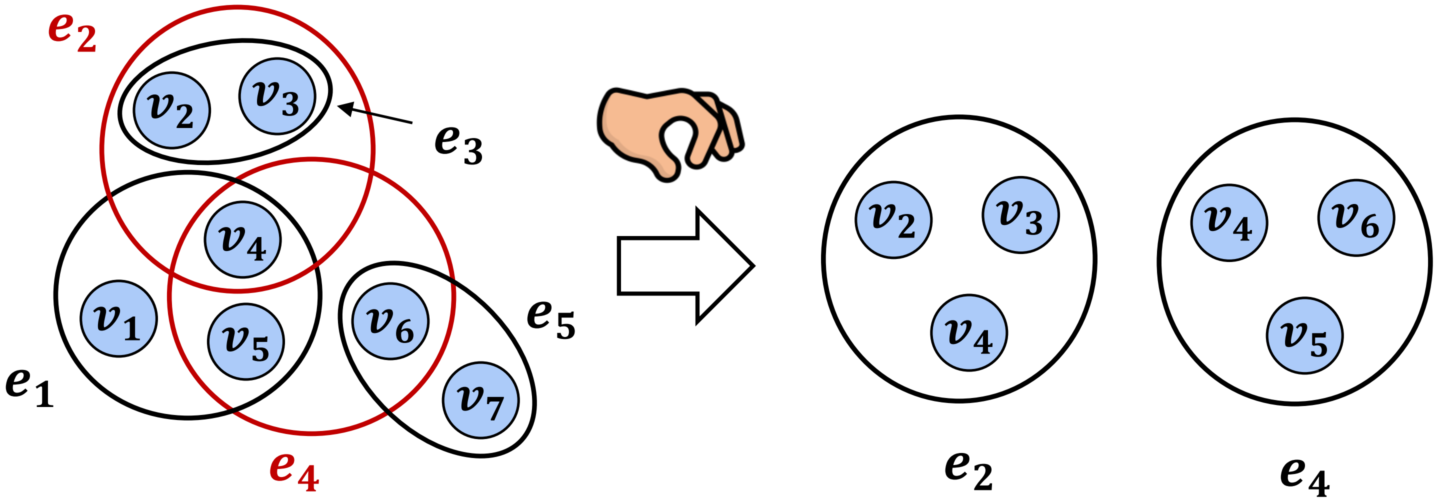
Real-world hypergraph



Randomized hypergraph

Pairwise Reshuffling

Step 1. Hyperedge Pair Sampling



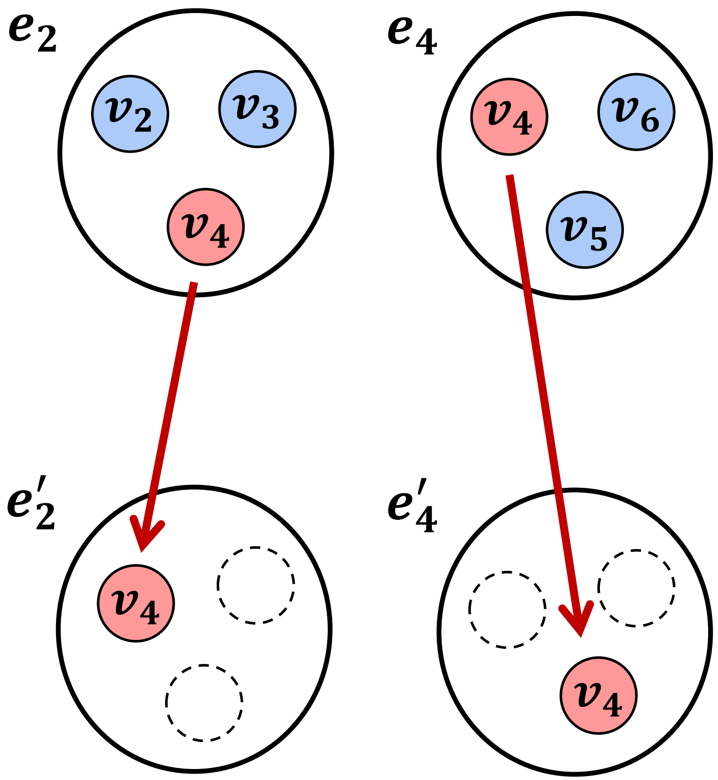
Step 1

Sample a pair of hyperedges uniformly at random.

$$(e_i, e_j) \in \binom{\boldsymbol{E}}{\boldsymbol{2}}$$

Pairwise Reshuffling (cont.)

Step 2. Shuffle Hyperedges

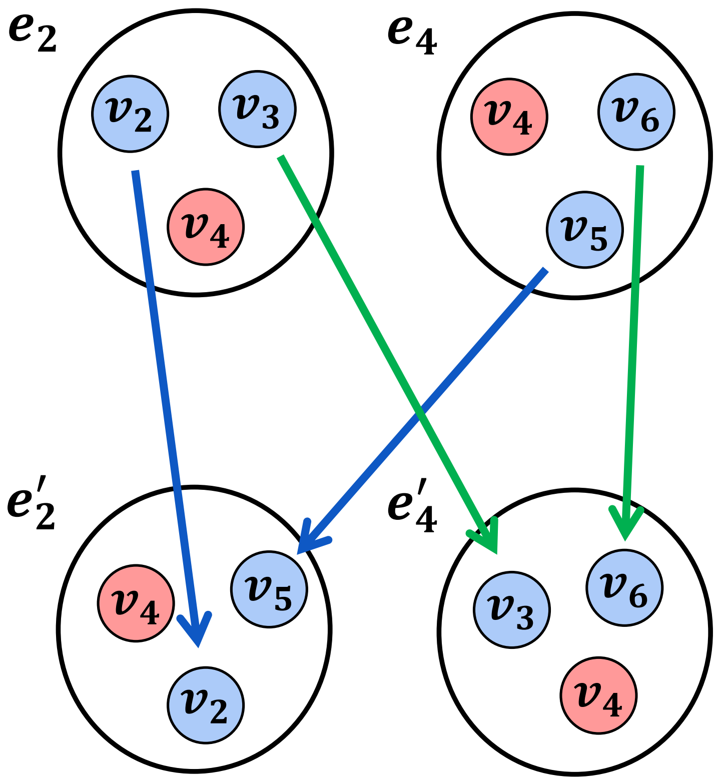


Step 2-1

For each node $v \in e_i \cap e_j$, add to both e'_i and e'_j.

Pairwise Reshuffling (cont.)

Step 2. Shuffle Hyperedges



Step 2-2

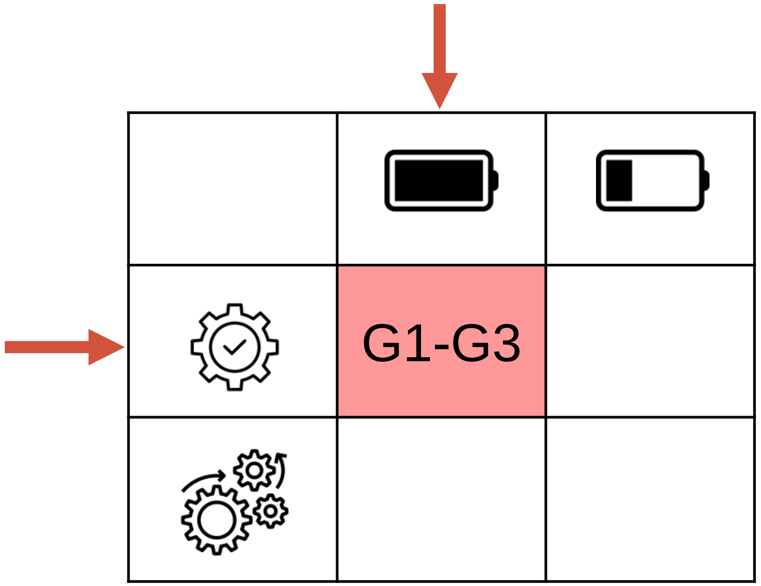
From $(e_i \cup e_j) - (e_i \cap e_j)$, sample $|e_i - e_j|$ nodes and add to e'_i.

Step 2-3

Add remaining nodes to e'_j.

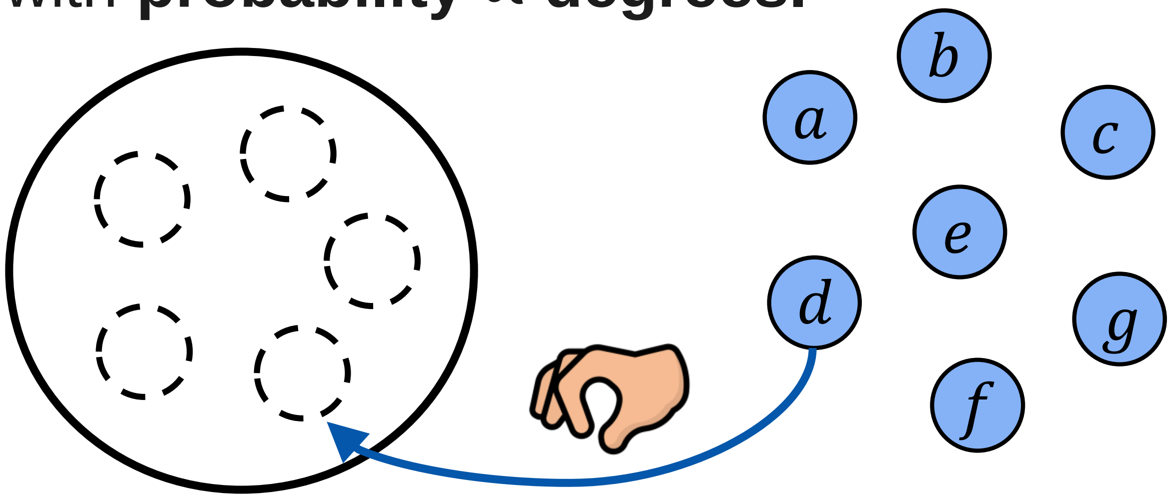
LCS21: Static Full-Hypergraph Generator

- **G1.** HyperCL: **Hyper**grpaph **C**hung-**L**u (Basic model)
- **G2.** HyperLap: **Hyper**graph Over**Lap** (Multilevel HyperCL)



HyperCL: Basic Model

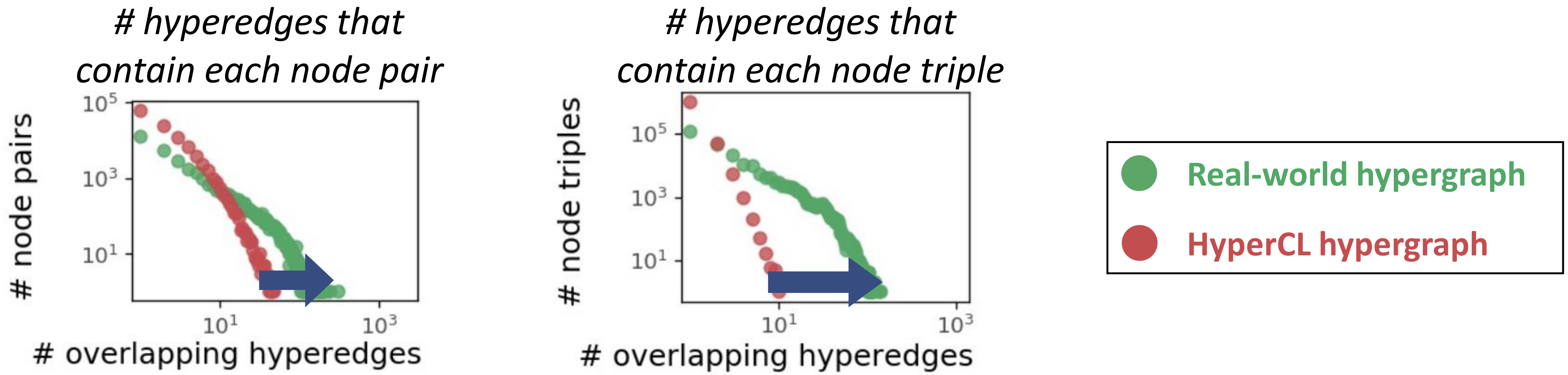
- An approximate configuration model.
- **HyperCL** fills each hyperedge with sampled nodes.
  - Samples nodes with **probability** ∝ **degrees.**



Random sampling
prob. ∝ d's degree

Evaluation of Configuration Models

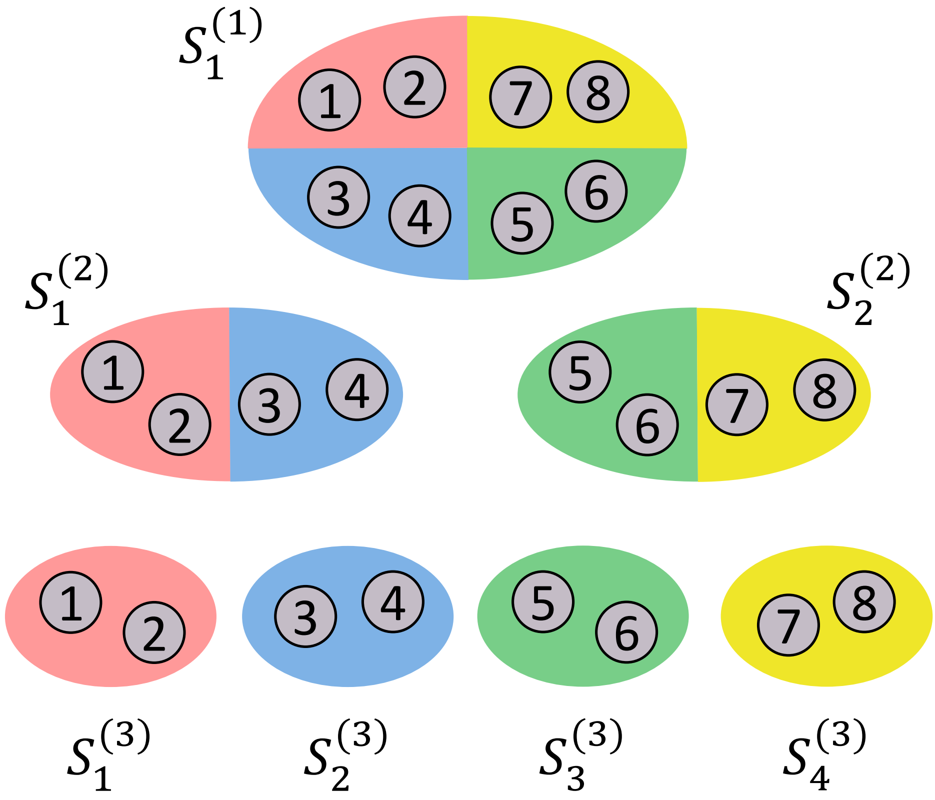
- **Configuration models** preserve node degrees and hyperedge sizes.
- They are limited in reproducing realistic overlapping patterns.
- Especially, they **fail to produce highly-overlapping hyperedges**.



HyperLap: Multilevel HyperCL

Step 1. Random Hierarchical Grouping of Nodes

Given number of levels L (L=3 for example)



At the lowest **level 1**, group $\boldsymbol{S_1^{(1)}}$ is the entire node set:

$$\boldsymbol{S_1^{(1)} = V}$$

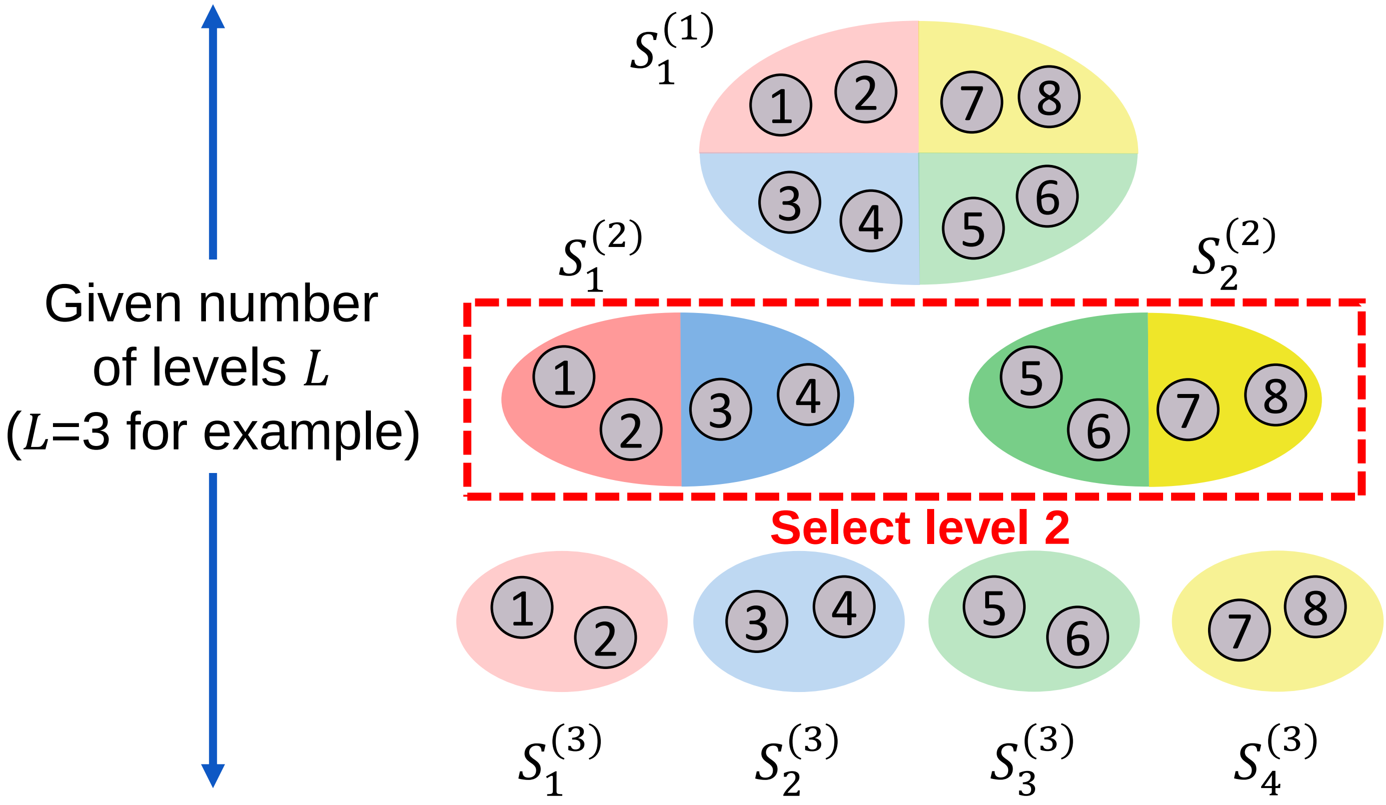
At **level** ℓ, group $\boldsymbol{S_i^{(\ell)}}$ is the union of the lower level's ones:

$$\boldsymbol{S_i^{(\ell)} = S_{2i-1}^{(\ell+1)} \cup S_{2i}^{(\ell+1)}}$$

Nodes are partitioned into $\boldsymbol{2^{L-1}}$ disjoint groups.

HyperLap: Multilevel HyperCL (cont.)

Step 2. Hyperedge Generation



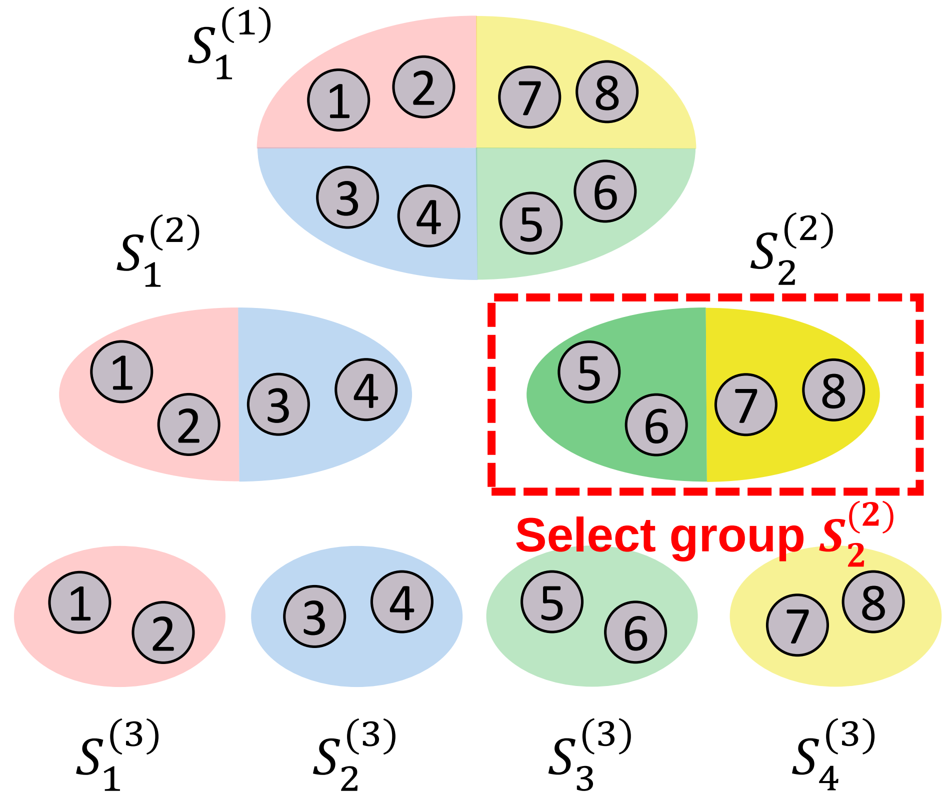
Step 2-1

Select a level with probability proportional to the given weight of each level $\{w_1, \ldots, w_L\}$.

HyperLap: Multilevel HyperCL (cont.)

Step 2. Hyperedge Generation

Given number of levels L (L=3 for example)



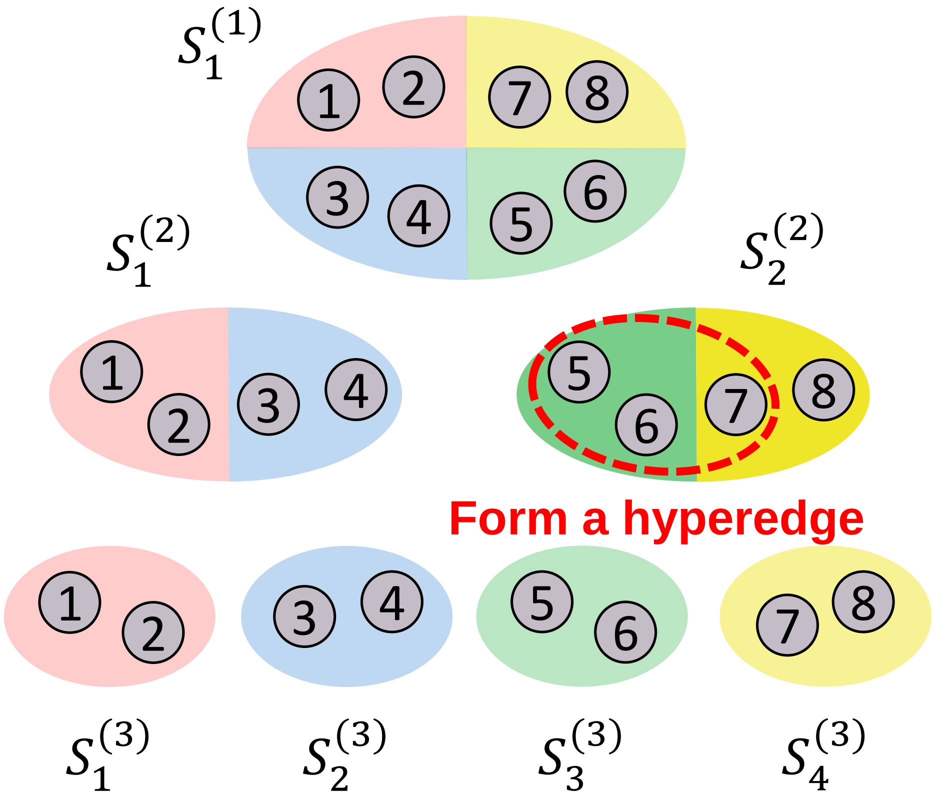
Step 2-2

Select a group uniformly at random.

HyperLap: Multilevel HyperCL (cont.)

Step 2. Hyperedge Generation

Given number of levels L (L=3 for example)

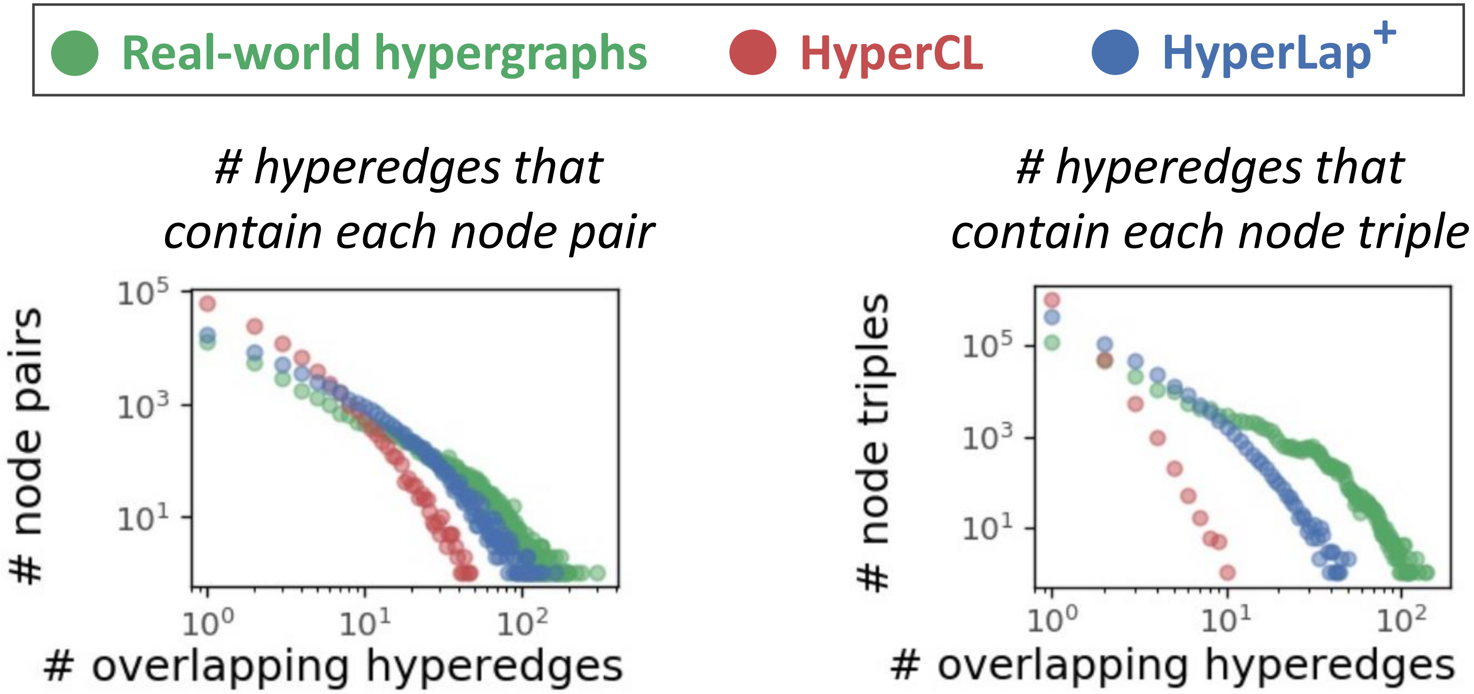


Step 2-3

Sample nodes independently with probability proportional to the degree of each node to form a hyperedge.

Evaluation of HyperLap

- **HyperLap** produces realistic overlaps of hypereges in **many** aspects.
- For example, **HyperLap** yields highly-overlapping hyperedges.

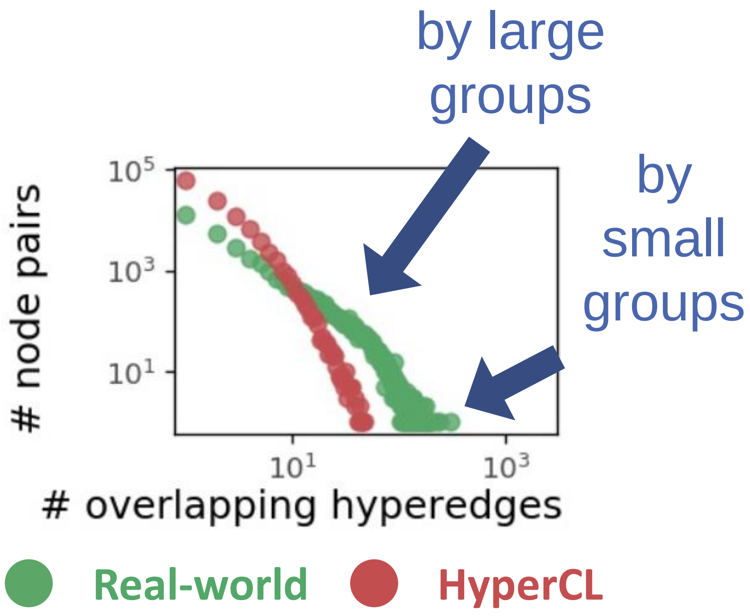


Details

Evaluation of HyperLap (cont.)

?

Question: How can **HyperLap** yield realistic **overlapping patterns** of hyperedges?



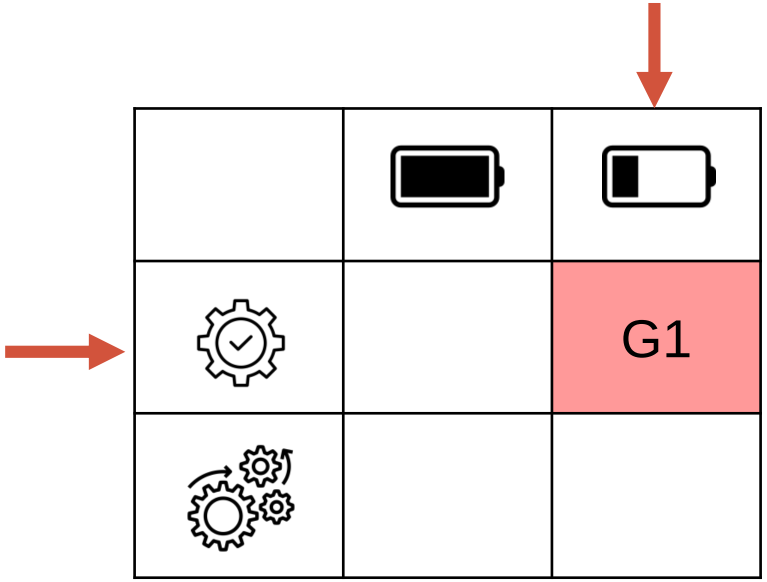
Answer:

- **HyperLap** generates hyperedges from groups of various sizes.
- Hyperedges from **small groups highly overlap** with each other.
- Hyperedges from **large groups less overlap** with each other.

!

CYLBKS22: Static Sub-Hypergraph Generator

- **G1.** MiDaS: **Mi**nimum **D**egree Bi**a**sed **S**ampling of Hyperedges



Hypergraph Representative Sampling

?

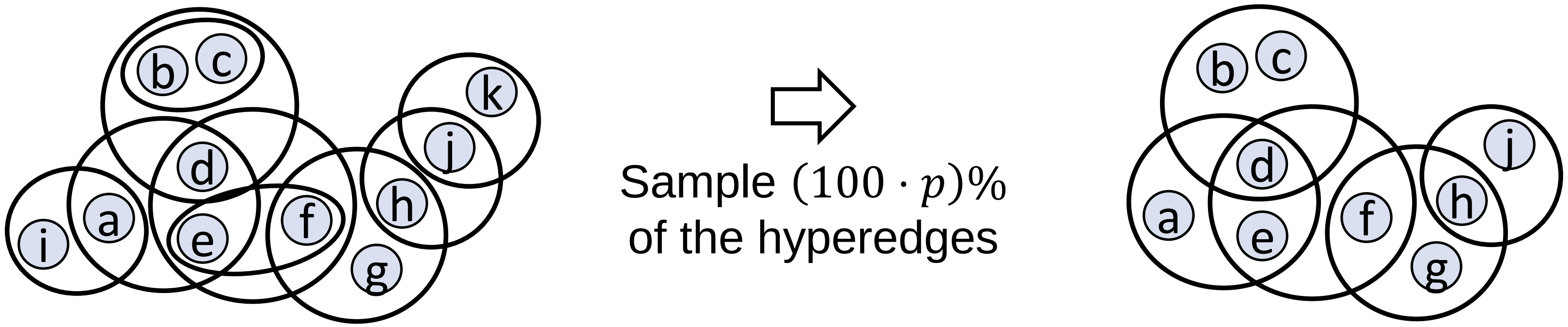
Question:

From a hypergraph $\mathcal{G}$, how can we generate a **smaller hypergraph** $\widehat{\mathcal{G}}$ that preserves the structural properties?

Answer:

We sample a **representative sub-hypergraph** $\widehat{\mathcal{G}}$ from $\mathcal{G}$

!



Hypergraph Representative Sampling (cont.)

?

Question:

What is a **representative** sub-hypergraph?

Answer:

We consider **10 structural properties**.

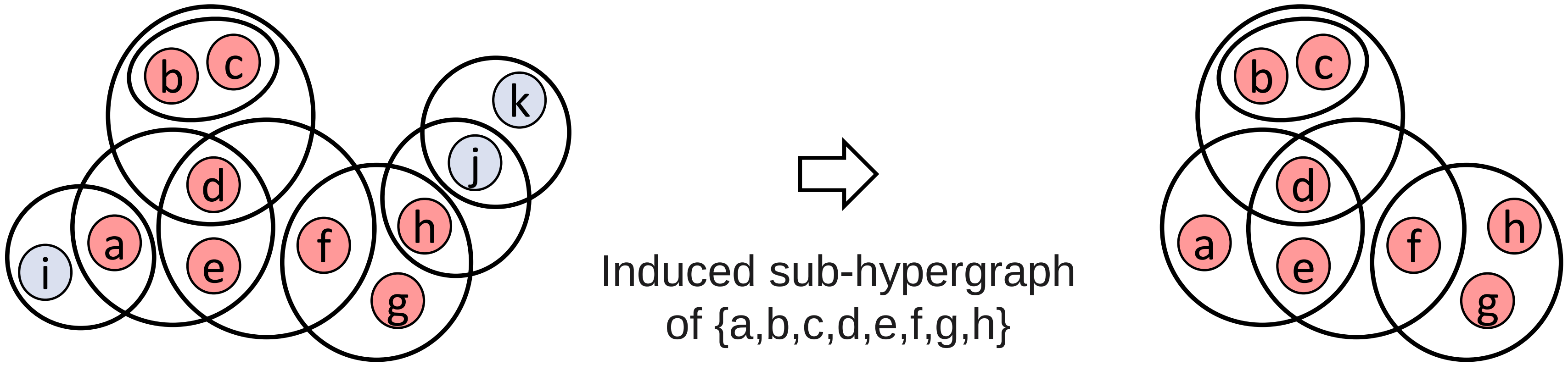
P1. Degree	P5. Singular Values	P8. Density
P2. Pair Degree	P6. Connected Component Size	P9. Overlapness
P3. Size	P7. Global Clustering Coefficient	P10. Effective Diameter
P4. Intersection Size		

Node-level, hyperedge-level, and hypergraph-level structural properties

!

Simple and Intuitive Approaches

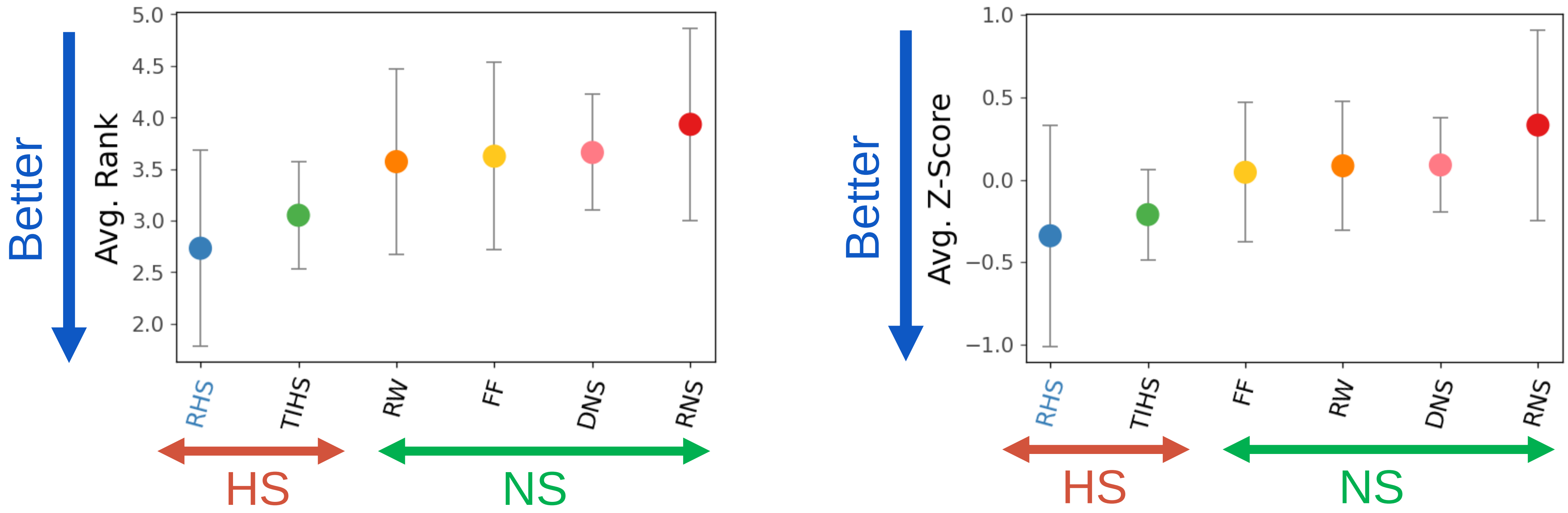
- **Node selection (NS)** chooses a subset of nodes and returns the sub-hypergraph induced by the nodes.



- **Hyperedge selection (HS)** directly chooses a subset of hyperedges.

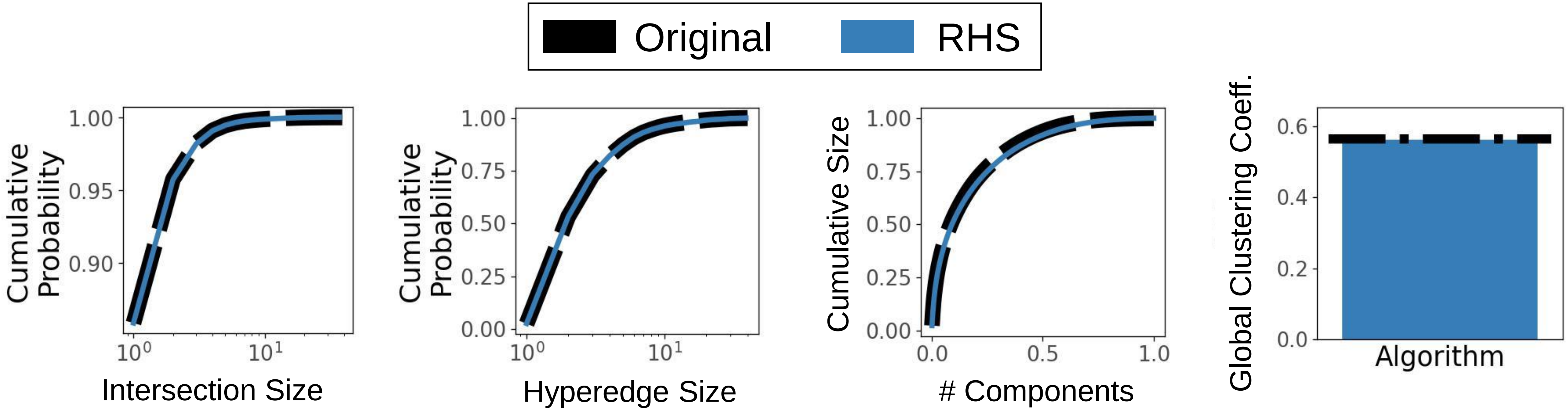
Simple and Intuitive Approaches (cont.)

- **RHS (Random Hyperedge Sampling)** performs best overall
  - **RHS** chooses hyperedges **uniformly at random**



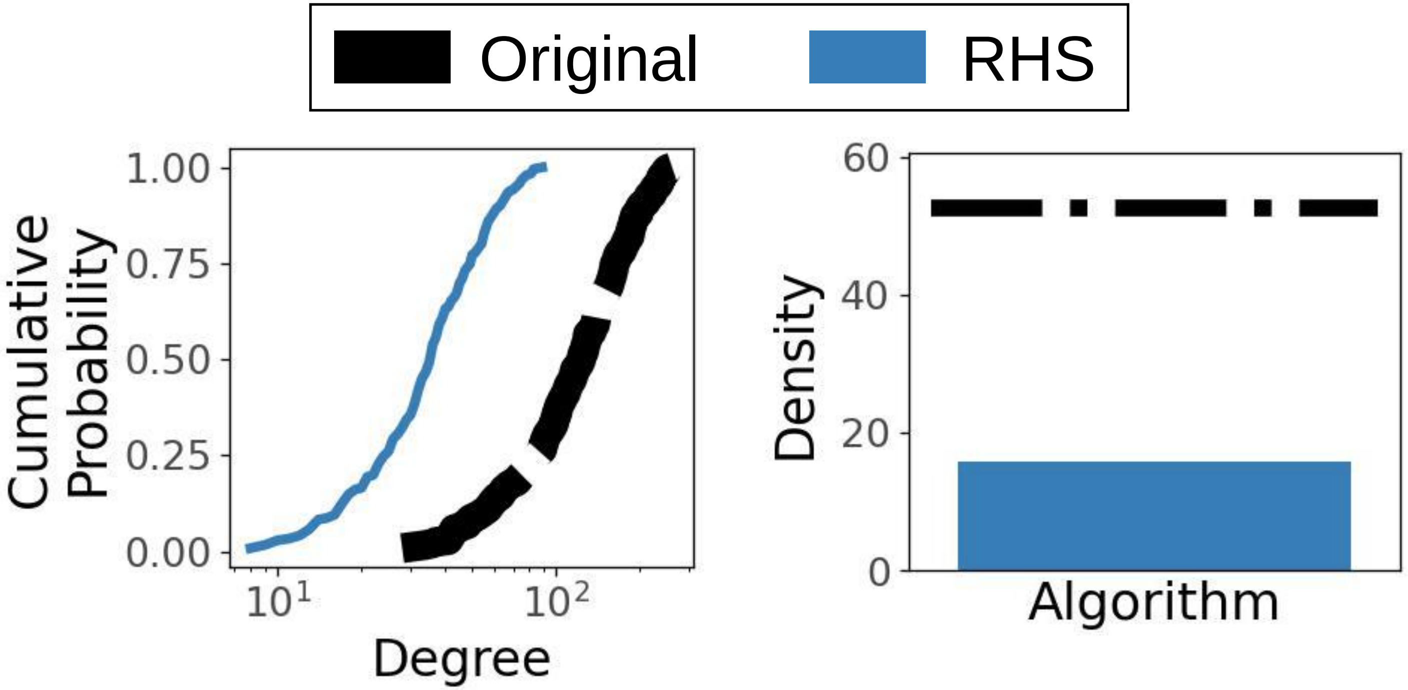
Random Hyperedge Sampling: Pros

- **RHS** preserves many structural properties surprisingly well.



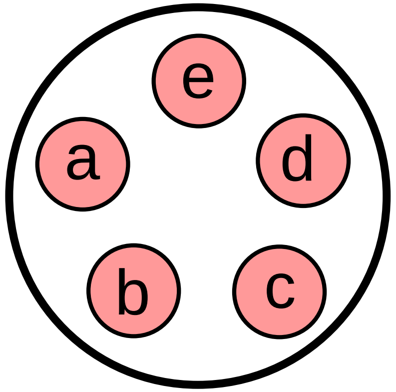
Random Hyperedge Sampling: Cons

- **RHS** gives a weakly connected sub-hypergraph.
- Especially, RHS suffers from a **lack of high-degree nodes.**



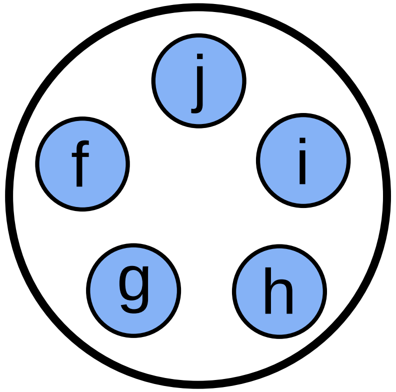
MiDaS-Basic: Main Idea

- To increase the fraction of high-degree nodes, **prioritize** hyperedges composed of **high-degree nodes**.



Node (v)	Degree (d_v)
a	4
b	2
c	3
d	8
e	6

$<$

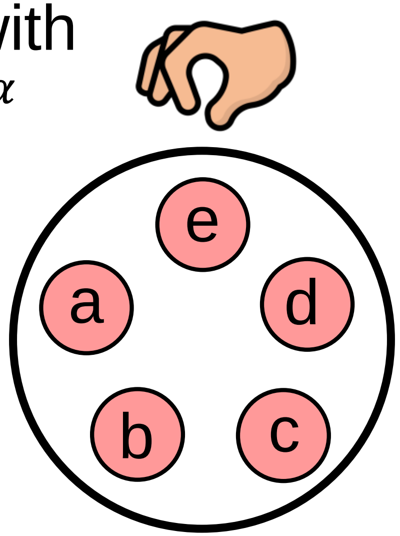


Node (v)	Degree (d_v)
f	10
g	12
h	8
i	10
j	8

MiDaS-Basic: Main Idea (cont.)

- Sampling a target number of hyperedges with probability proportional to the **minimum degree of nodes** in each hyperedge to the power of α

Sampling this hyperedge with probability $\propto (\min_{v \in e} d_v)^{\alpha}$

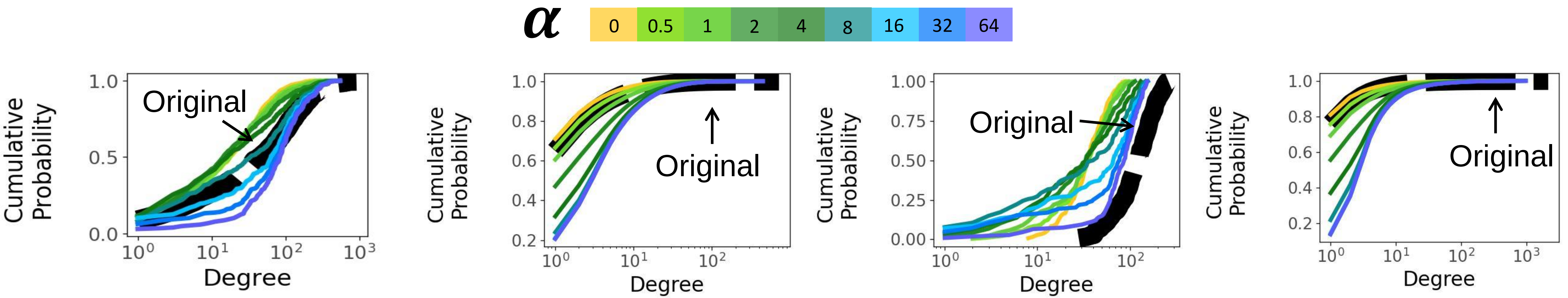


Node	Degree
a	4
b	2
c	3
d	8
e	6

Details

MiDaS-Basic: Empirical Properties

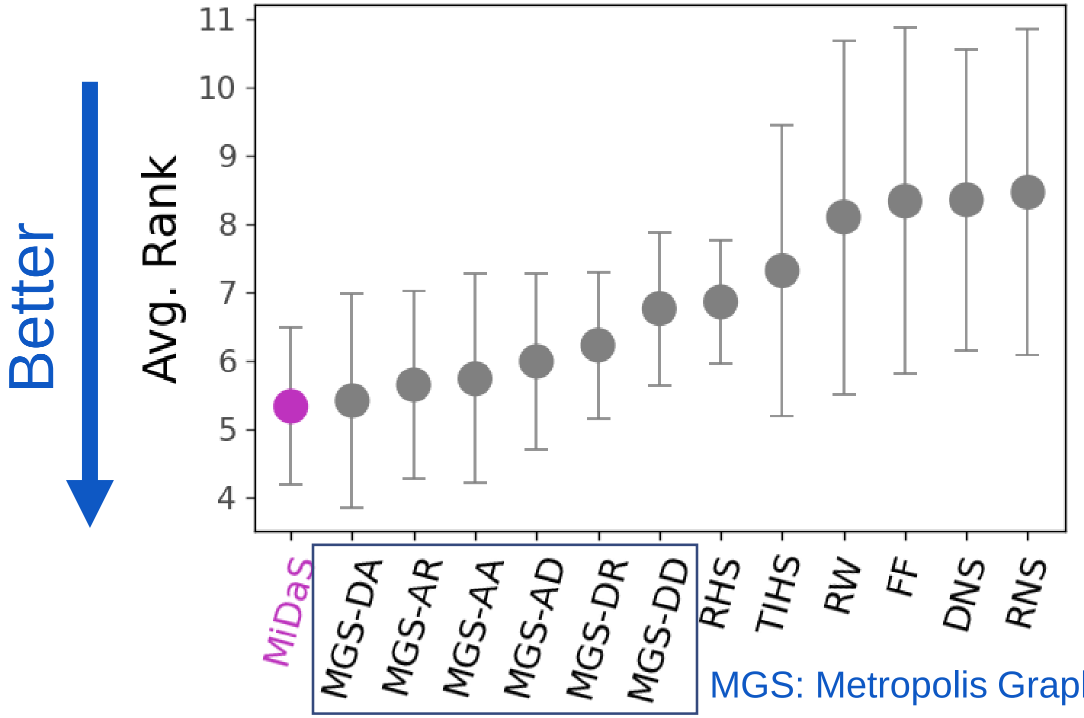
- With a proper α **value, degree distributions are well preserved.**
- Motivates (full-fledged) MiDaS with a hill-climbing search of α



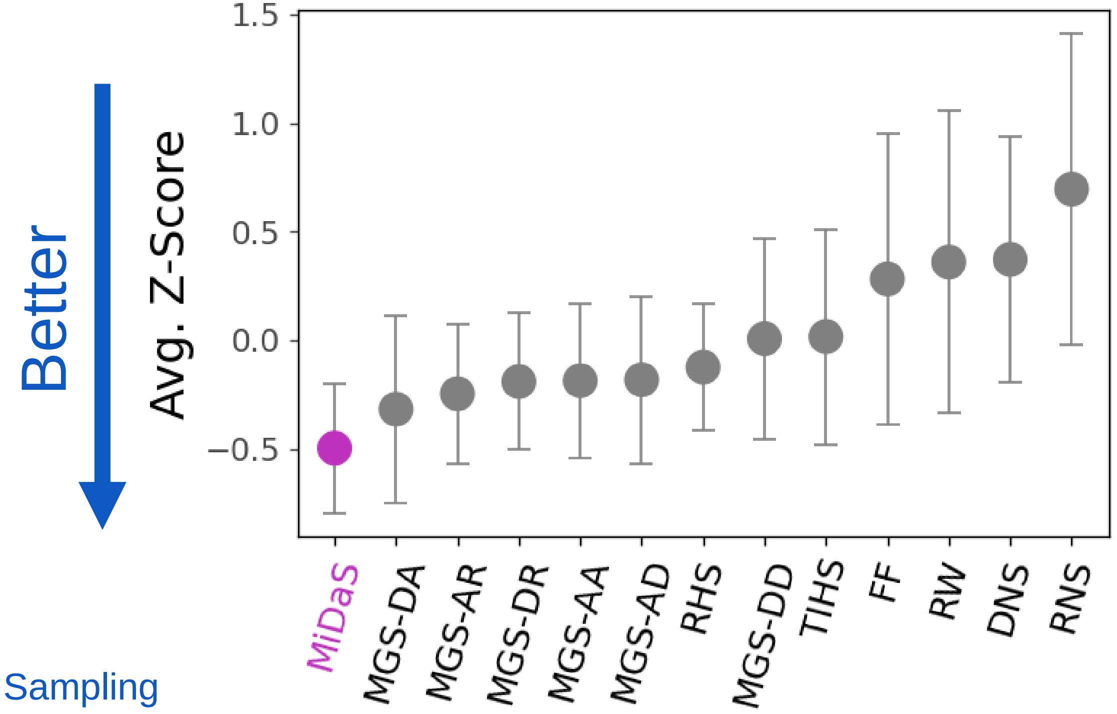
Email-Eu **Coauth Geology** **Contact Primary** **Threads Ubuntu**

MiDaS: Evaluation

- **MiDaS** provides overall the most representative samples in terms of both average rankings and Z-scores.



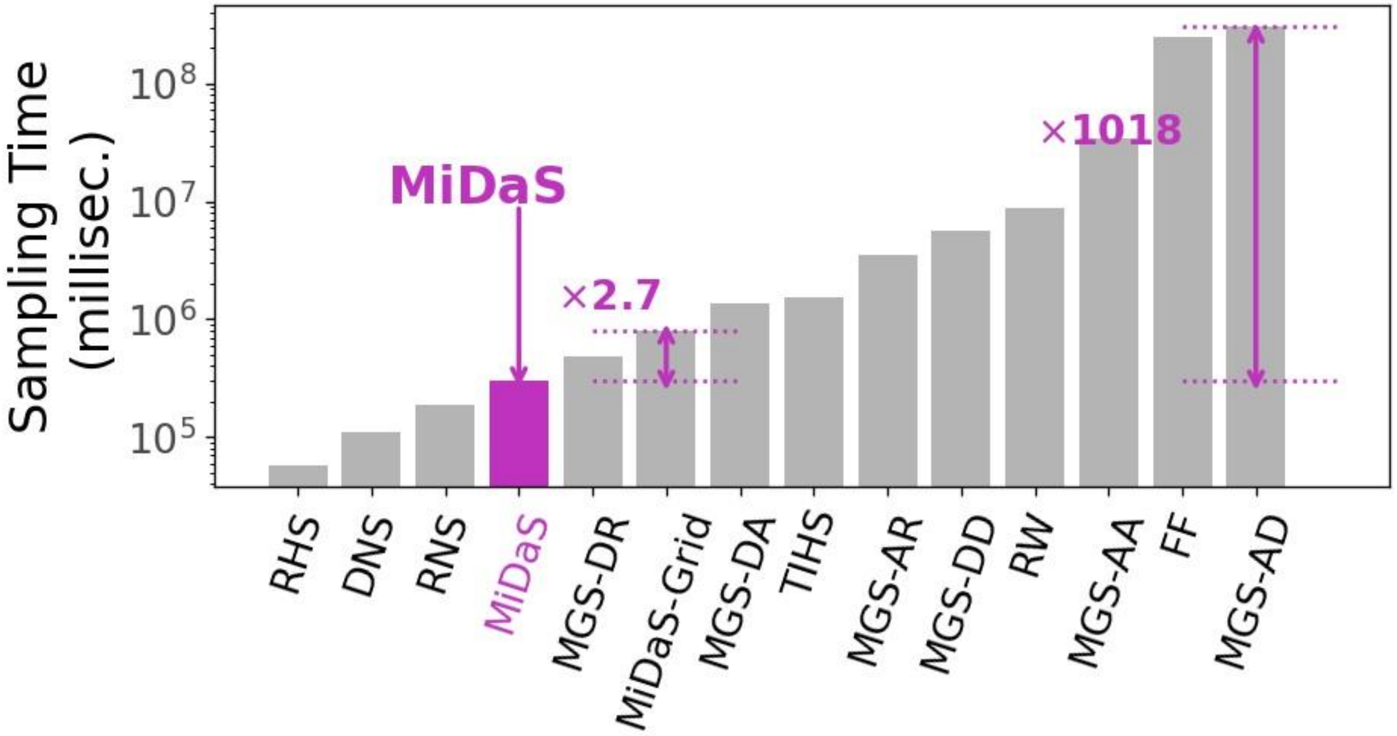
MGS: Metropolis Graph Sampling



M. Choe, J. Yoo, G. Lee, W. Baek, U Kang, K. Shin. "MiDaS: Representative Sampling from Real-world Hypergraphs", **WWW 2022**

MiDaS: Evaluation (cont.)

- **MiDaS** is the fastest except for the simplest methods.



Roadmap

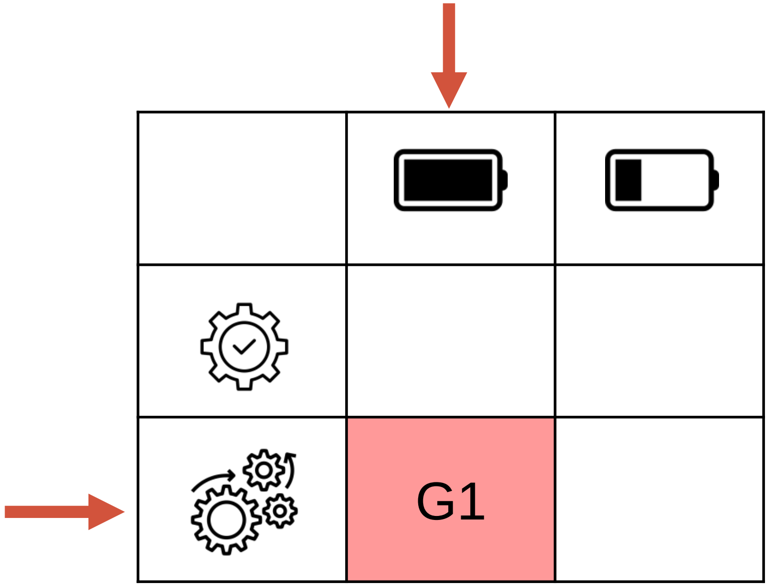
- **Part 1. Static Structural Patterns**
  - Basic Patterns
  - Advanced Patterns
- **Part 2. Dynamic Structural Patterns**
  - Basic Patterns
  - Advanced Patterns
- **Part 3. Generative Models**
  - Static Hypergraph Generator
  - Dynamic Hypergraph Generator **<<**

Part 3-2. Dynamic Hypergraph Generative Models

		Part 3. Generative Models
Static Models	Full-Hypergraphs	C20, LCS21
	Sub-Hypergraphs	CYLBKS22
Dynamic Models	Full-Hypergraphs	DYHS20, KKS20, KBCYS23, GLLB23
	Sub-Hypergraphs	BKT18, CK21

DYHS20: Dynamic Full-Hypergraph Generator

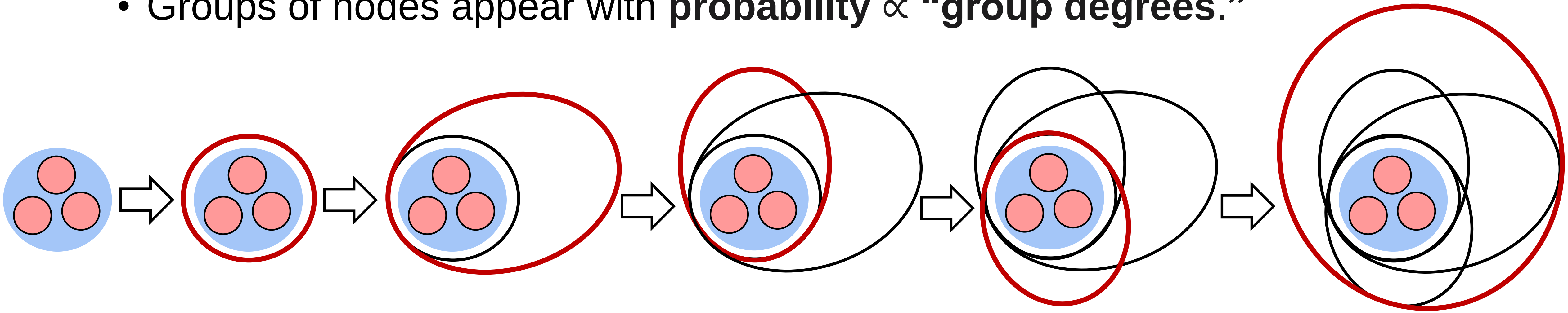
- **G1.** HyperPA: **Hyper**graph **P**referential **A**ttachment



M. T. Do, S. Yoon, B. Hooi, K. Shin, "Structural Patterns and Generative Models of Real-world Hypergraphs", **KDD 2020**

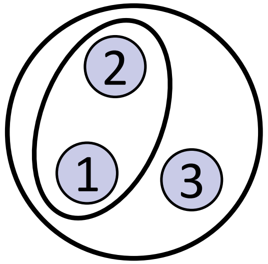
HyperPA: Preferential Attachment

- Inspired by the PA model for pairwise graphs [AA02]
  - Preferential attachment or "the rich get richer"
- Main idea: **"Subsets get rich together"**
  - Groups of nodes appear with **probability** ∝ **"group degrees**."



HyperPA: Preferential Attachment (cont.)

Hyperedge Generation



{1}: 2 {2}: 2 {3}: 1

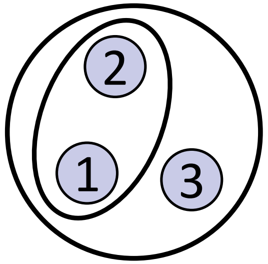
{1,2}: 2 {2,3}: 1 {3,1}: 1

{1,2,3}: 1

Group Degrees

HyperPA: Preferential Attachment (cont.)

Hyperedge Generation



{1}: 2 {2}: 2 {3}: 1

{1,2}: 2 {2,3}: 1 {3,1}: 1

{1,2,3}: 1

Group Degrees

A new **node 4** is added.

⇩ *Number of hyperedges*

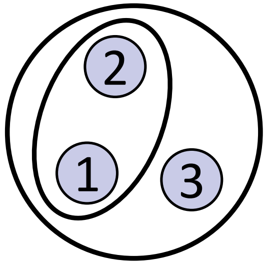
Add **2 hyperedges**.

⬇ *Size of a hyperedge*

A hyperedge size: **3**

HyperPA: Preferential Attachment (cont.)

Hyperedge Generation



{1}: 2 {2}: 2 {3}: 1

{1,2}: 2 {2,3}: 1 {3,1}: 1

{1,2,3}: 1

Group Degrees

Sample group prob. ∝ degree

A new **node 4** is added.

Number of hyperedges

Add **2 hyperedges**.

Size of a hyperedge

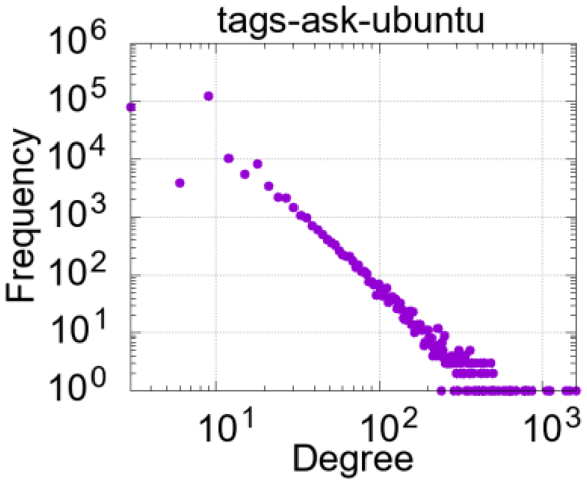
A hyperedge size: **3**

Sample groups

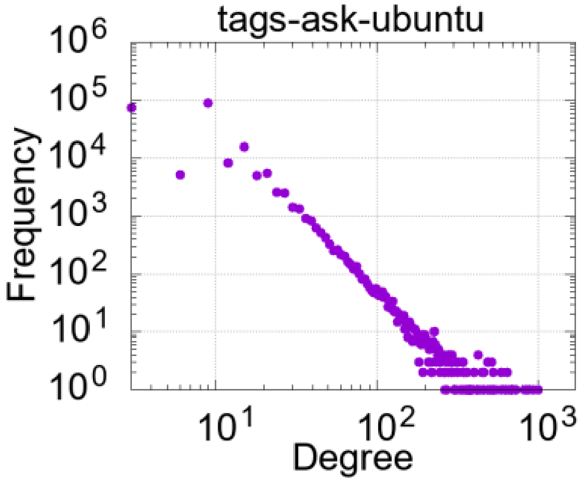
Generate hyperedge: **{1,2,4}**

HyperPA: Evaluation (cont.)

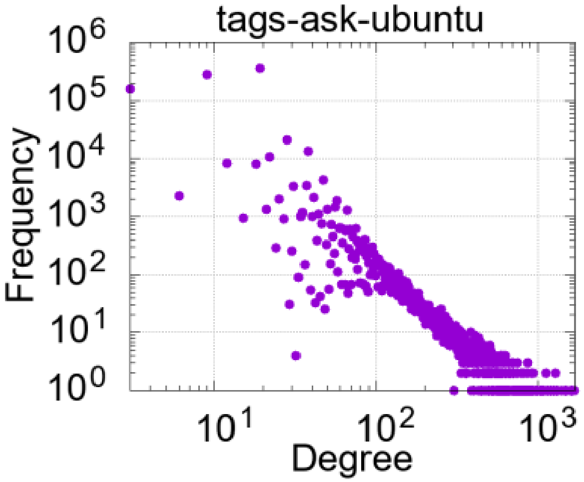
- **HyperPA** generates realistic higher-order structures.
  - HyperPA considers “group degrees.”
  - **NaivePA** (baseline) considers node degrees individually.



Real data



HyperPA

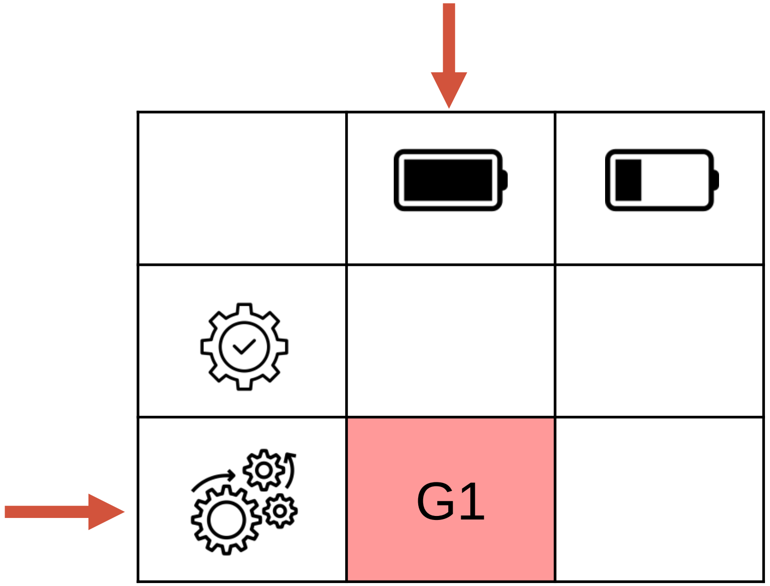


NaivePA

Degree distributions of the triangle-level decomposed graphs

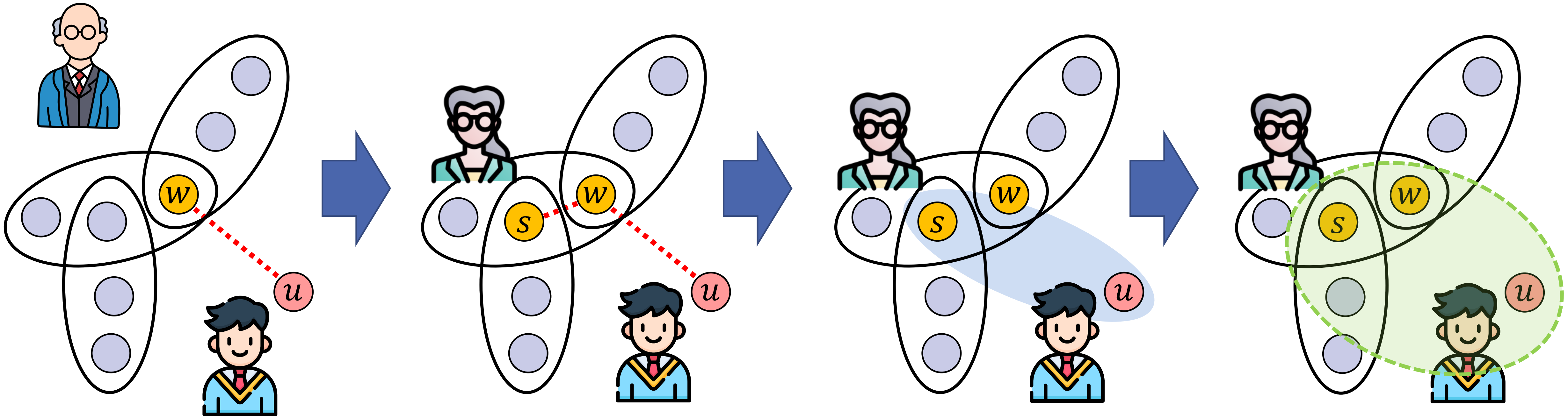
KKS20: Dynamic Full-Hypergraph Generator

- **G1.** HyperFF: **Hyper**graph **F**orest **F**ire



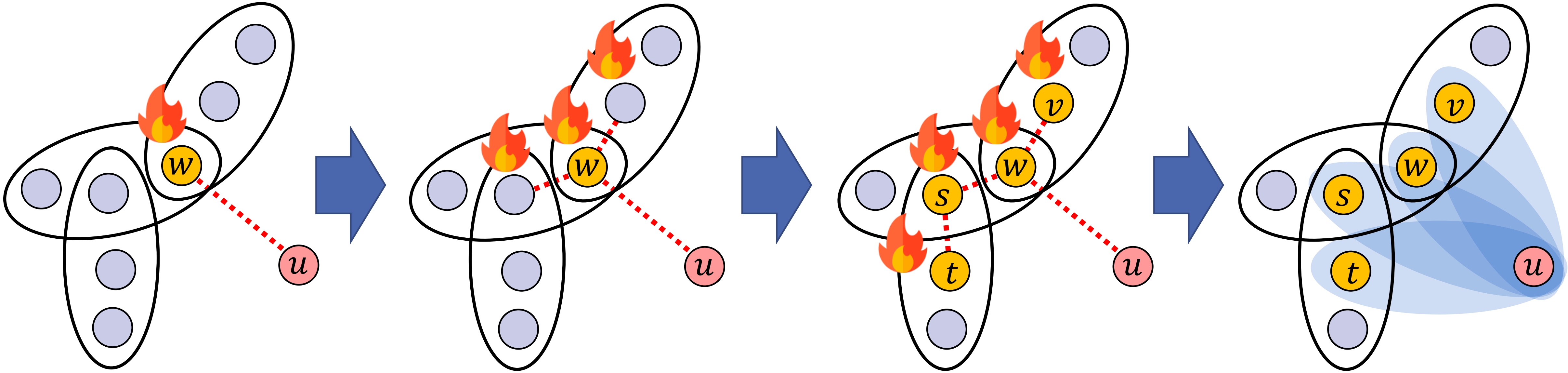
HyperFF: Motivation

- Inspired by the **forest fire model** for pairwise graphs [LKF05]



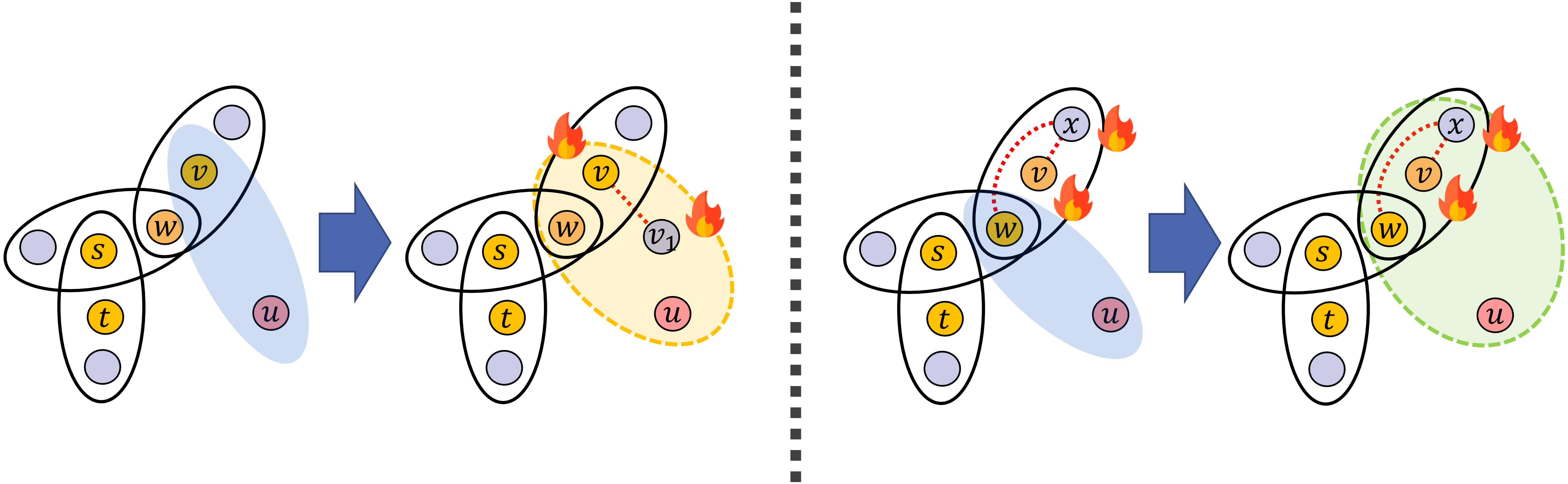
HyperFF: (Step 1) Hyperedge Formation

- For a new node, **HyperFF** simulates a **forest fire** from an ambassador.
- The forest fire is spread through hyperedges.
- The new nodes forms a **size-2 hyperedge** with each burned node.



HyperFF: (Step 2) Expansion

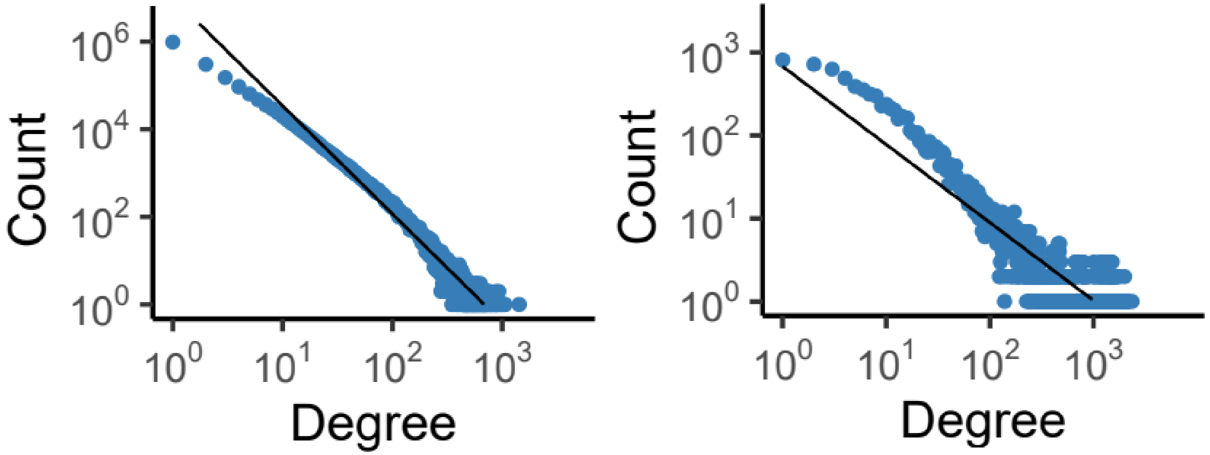
- Each size-2 hyperedge is **expanded again through a forest fire**.



HyperFF: Evaluation

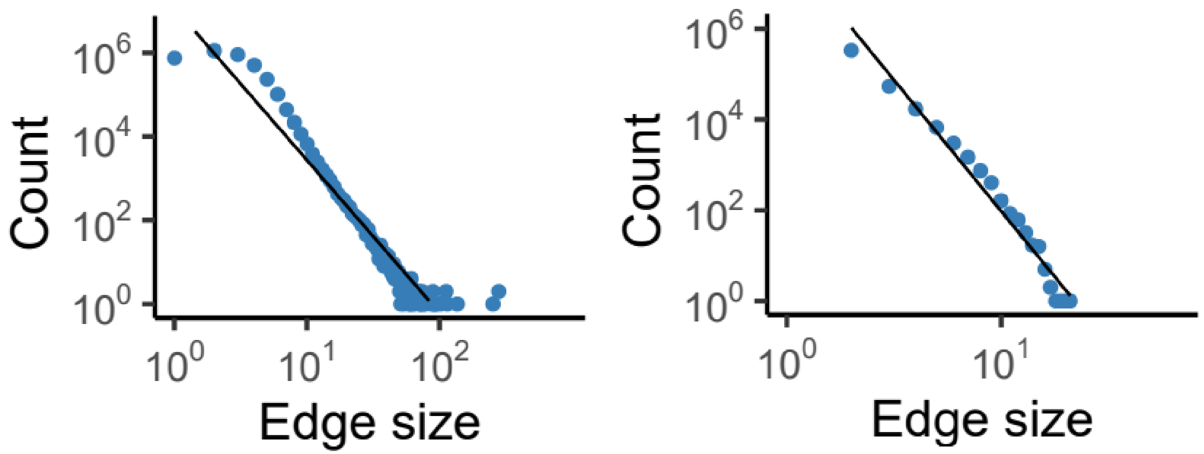
- **HyperFF** reproduces static structural patterns in real hypergraphs.

Real data | HyperFF



Degree distribution

Real data | HyperFF

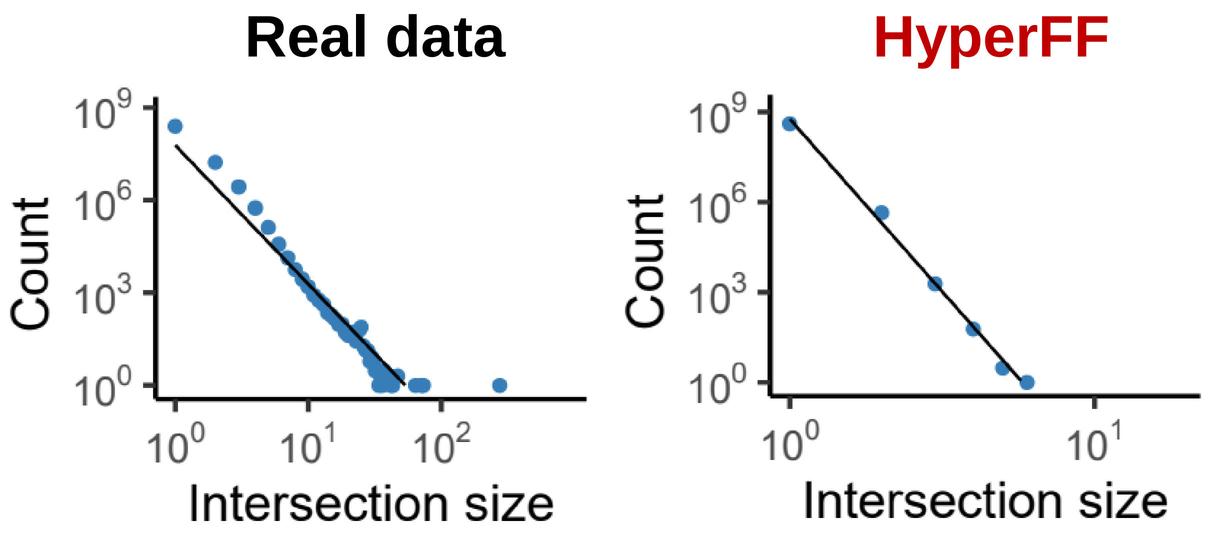


Hyperedge size distribution

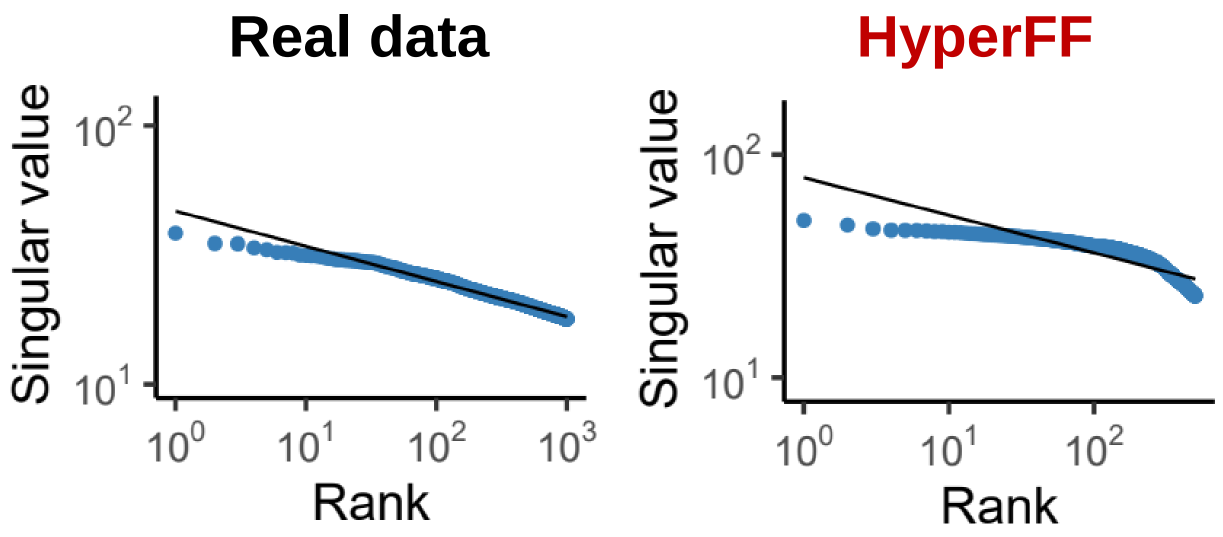
Y. Kook, J. Ko, K. Shin, "Evolution of Real-world Hypergraphs: Patterns and Models without Oracles", **ICDM 2020**

HyperFF: Evaluation (cont.)

- **HyperFF** reproduces static structural patterns in real hypergraphs.



Intersection size distribution

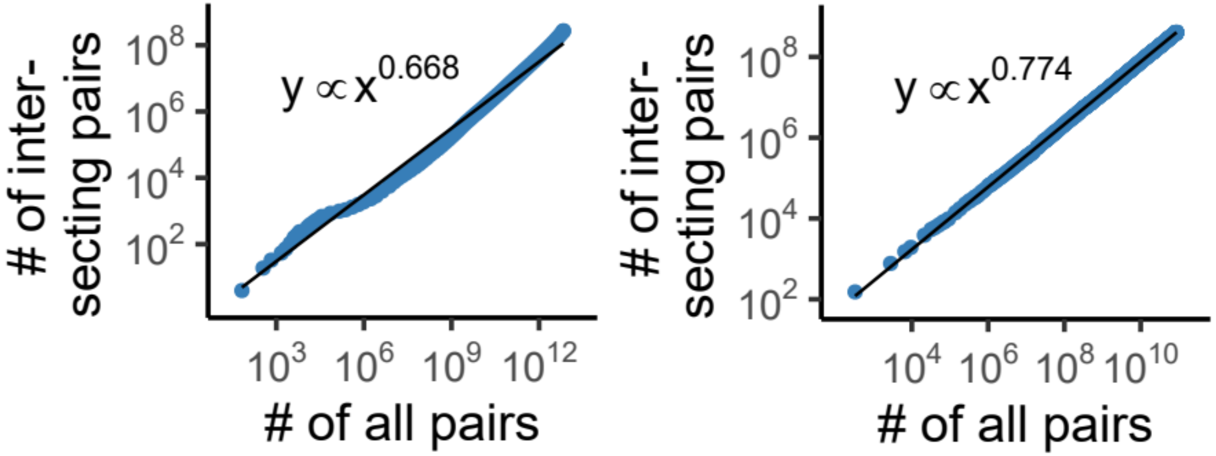


Singular value distribution

HyperFF: Evaluation (cont.)

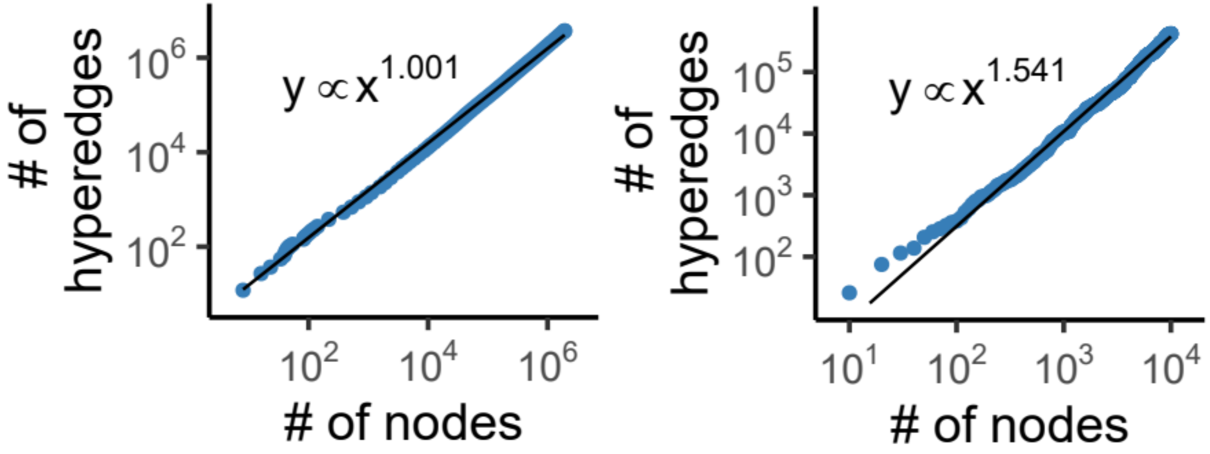
- **HyperFF** reproduces dynamic structural patterns in real hypergraphs.

Real data **HyperFF**



Diminishing overlaps

Real data **HyperFF**



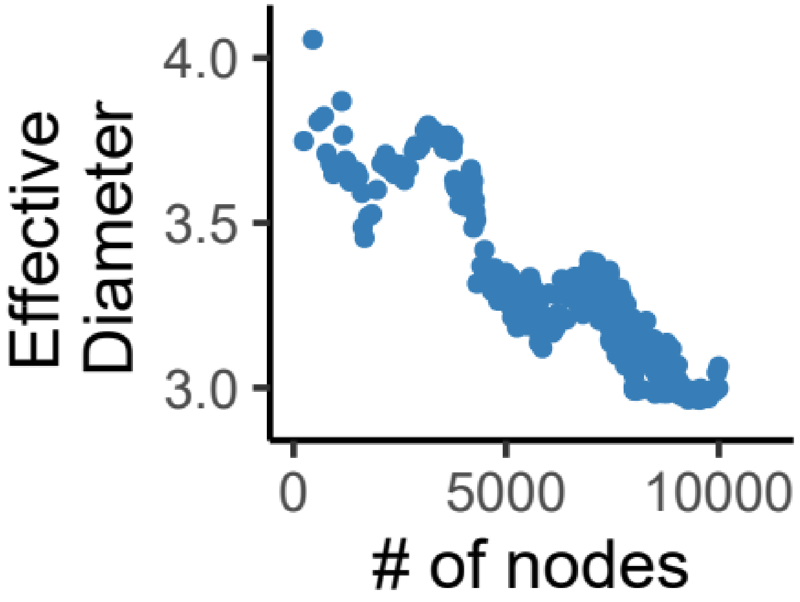
Increasing edge density

Y. Kook, J. Ko, K. Shin, "Evolution of Real-world Hypergraphs: Patterns and Models without Oracles", **ICDM 2020**

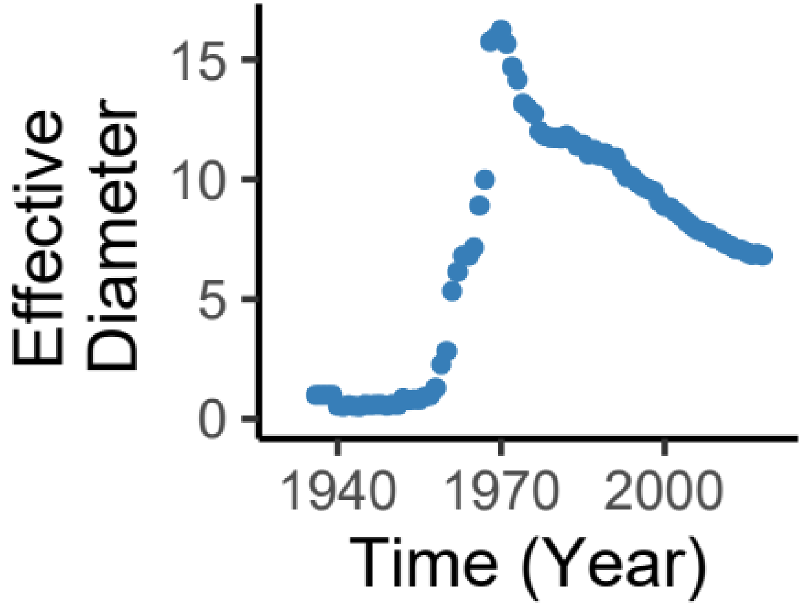
HyperFF: Evaluation (cont.)

- **HyperFF** reproduces dynamic structural patterns in real hypergraphs.

Real data



HyperFF

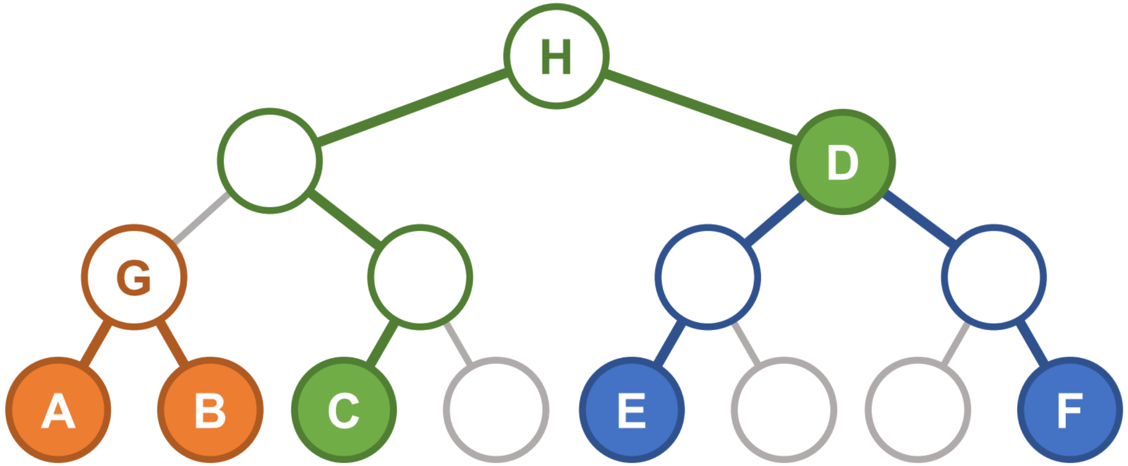


Shrinking diameter

Details

How Can HyperFF be Realistic?

- By simplifying **HyperFF**, several properties can be “proven.”
  - Heavy-tailed degree distribution
  - Densification

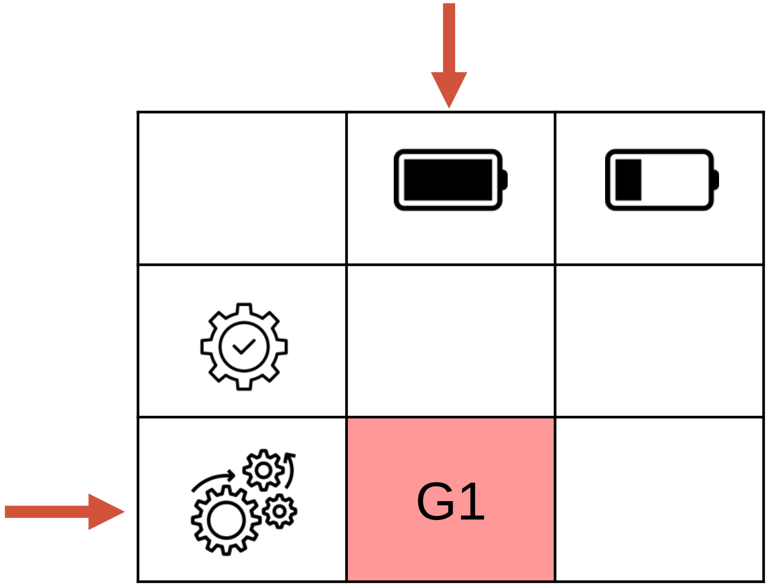


Simplified HyperFF: nodes are burned with prob. depending on the distance in a hierarchy tree

J. Ko, Y. Kook, K. Shin, “Growth patterns and models of real-world hypergraphs”, **Knowledge and Information Systems (2022)**

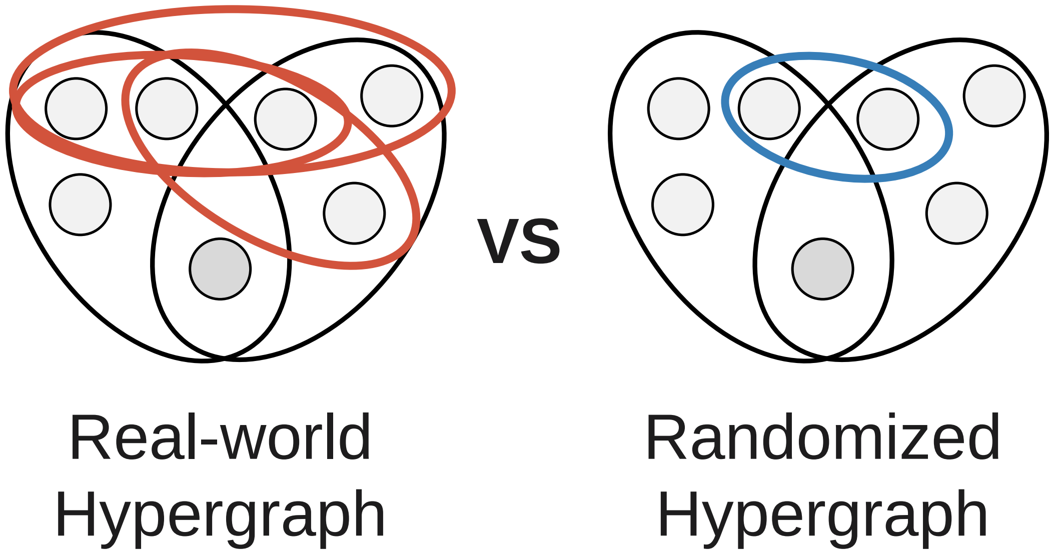
KBCYS23: Dynamic Full-Hypergraph Generator

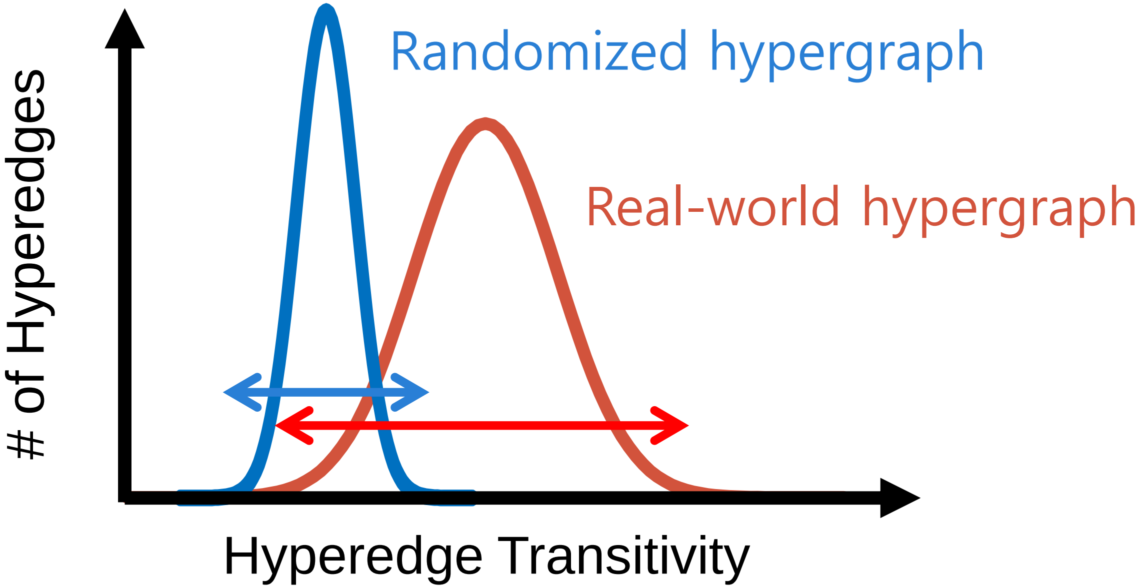
- **G1.** Naïve THera: Preliminary Version
- **G2.** THera: <u>**T**</u>ransitive <u>**H**</u>ypergraph Gen<u>**era**</u>tor



Recap: Transitivity of Hypergraphs

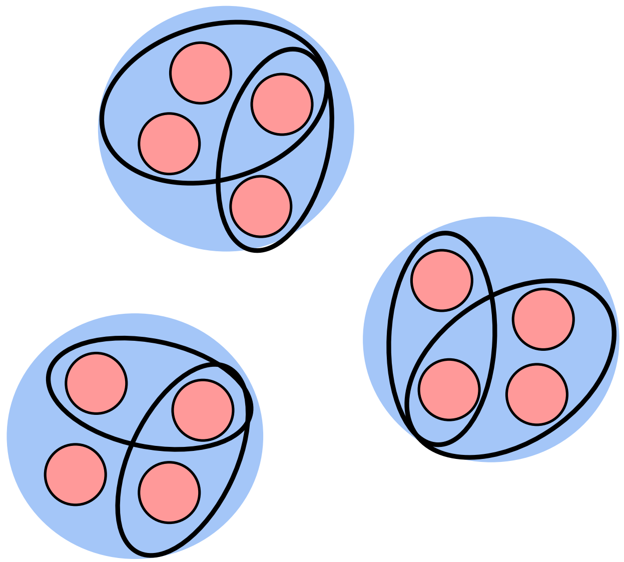
- Real-world hypergraphs exhibit **transitivity**.
- The level of transitivity introduced by HyperPA and HyperPA often deviates much from that in real-world hypergraphs.

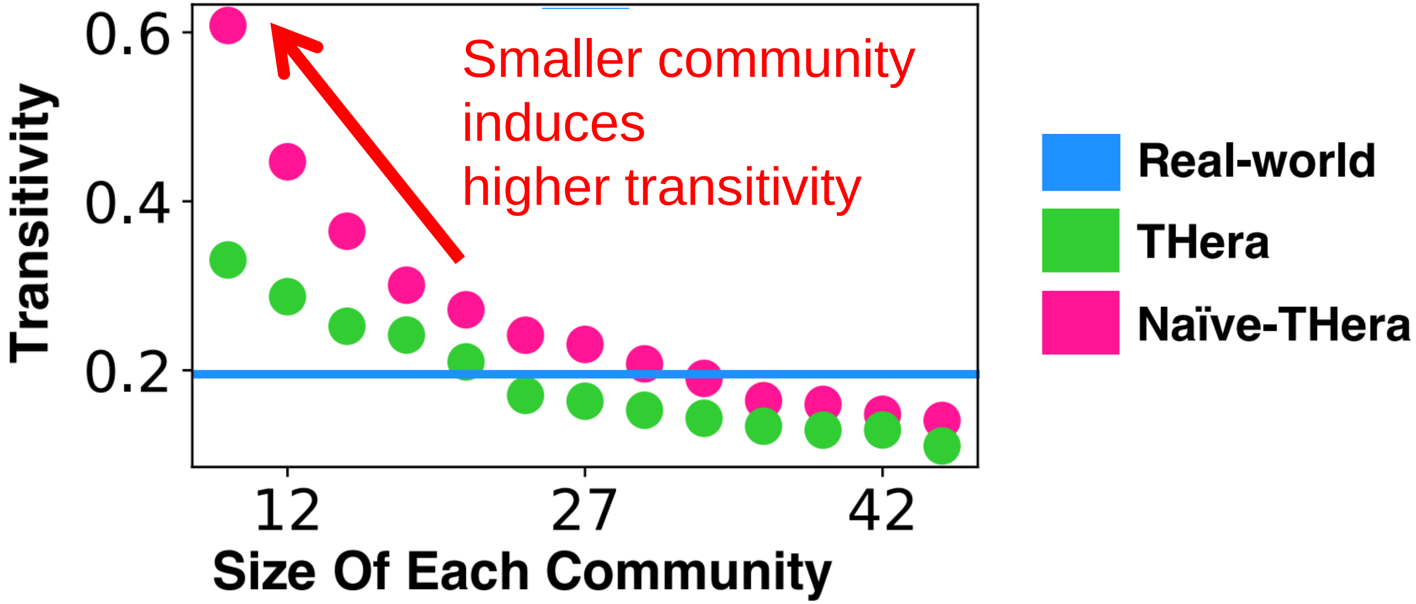




Naïve-THera: Preliminary Version

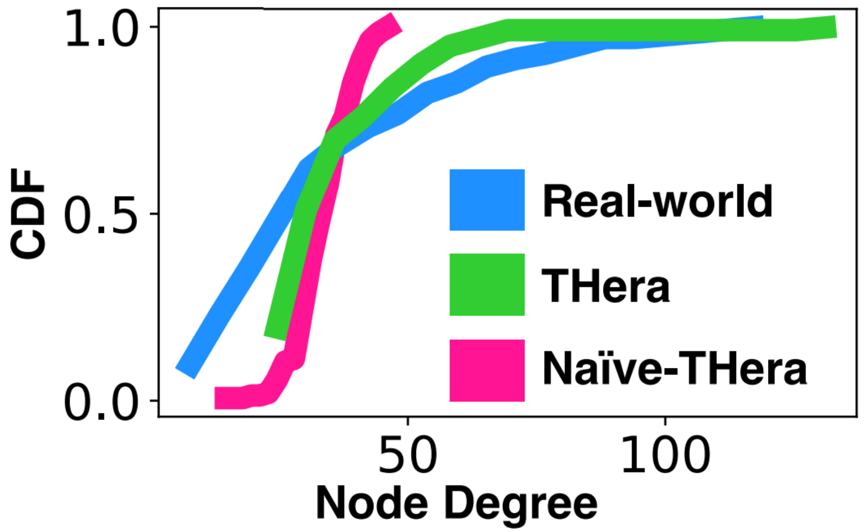
- **Naïve-THera** assigns each node to a community and generates intra-community hyperedges.
- The level of transitivity can be controlled by the size of communities.

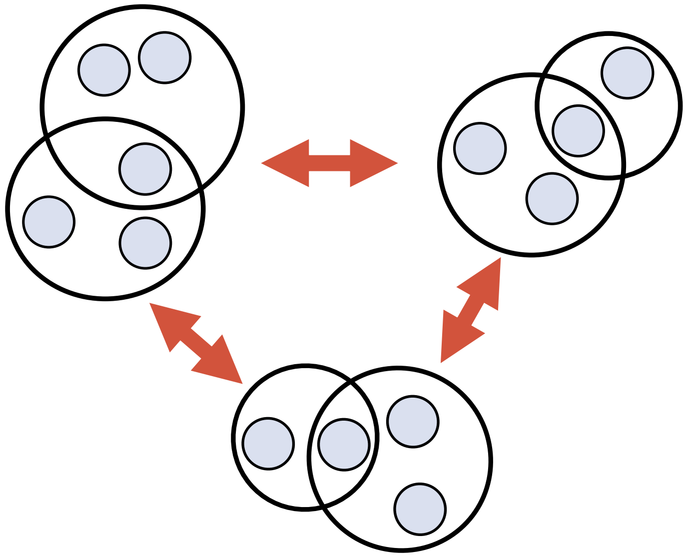




Naïve-THera: Preliminary Version (cont.)

- However, **Naïve-THera** generates unrealistic hypergraphs:
  - Near-uniform node degree distribution
  - Divided and disconnected hypergraphs

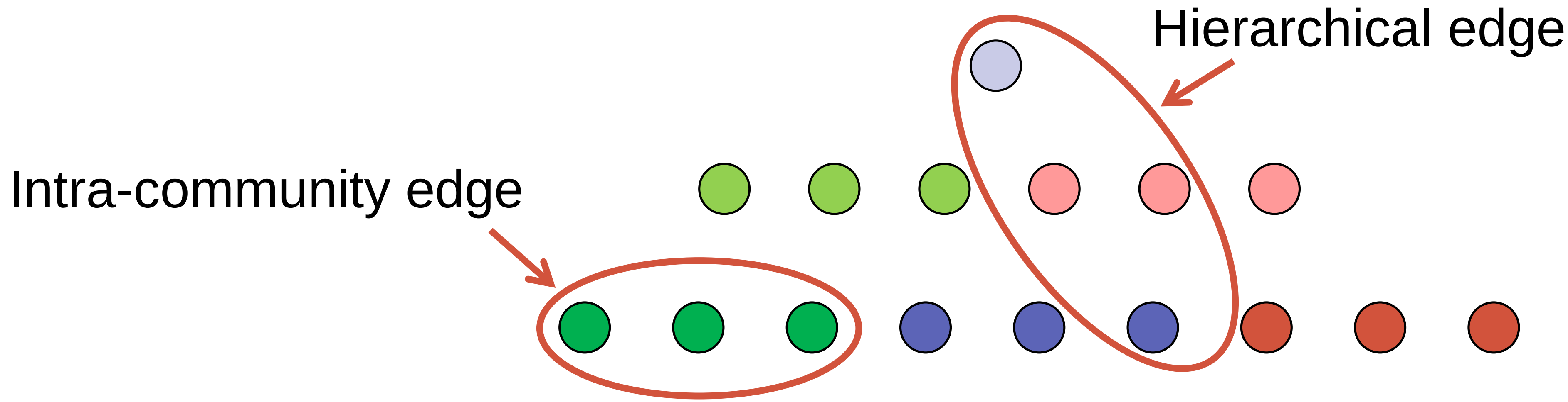




S. Kim, F. Bu, M. Choe, J. Yoo, and K. Shin, “How Transitive Are Real-World Group Interactions? Measurement and Reproduction”, **KDD 2023**

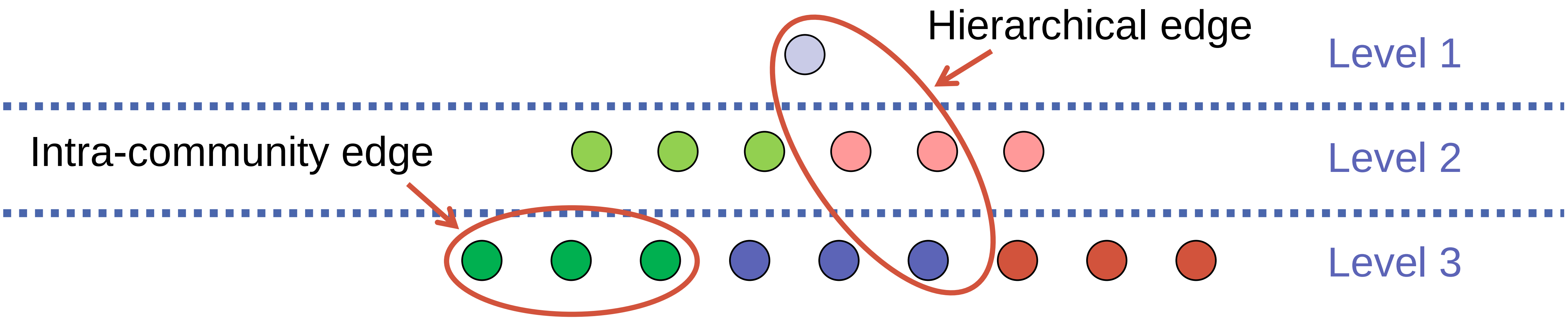
THera: Transitivity-Preserving Generator

- **THera** generates two types of hyperedges:
  - **Intra-community edges** consist of nodes within the same community.
  - **Hierarchical edges** consist of nodes from different communities.
- Hierarchical edges **address disconnectivity** by bridging communities.



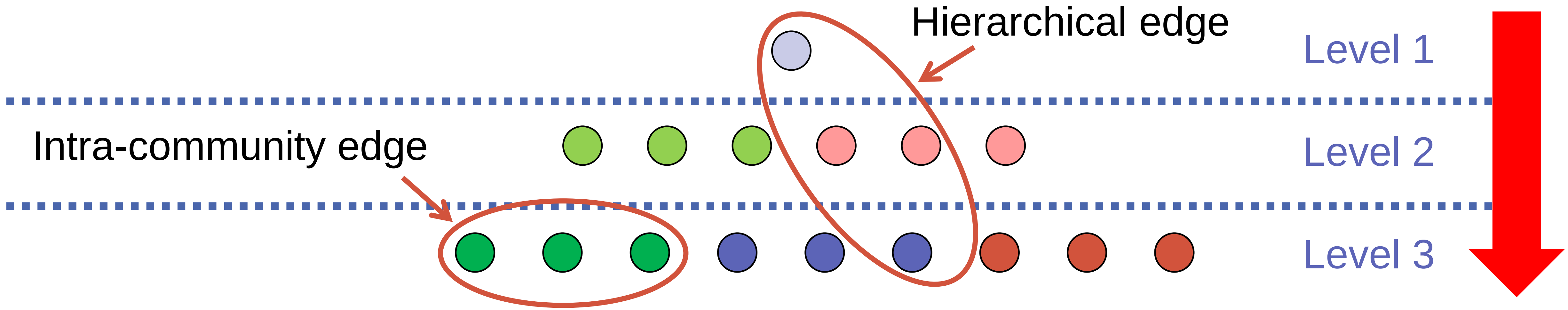
THera: Transitivity-Preserving Generator (cont.)

- Hierarchical edges lead to a **skewed node degree distribution**.
  - **Tree-like hierarchy** of nodes is assumed.
  - Each node appears in edges with priority based on the level it belongs to.



THera: Transitivity-Preserving Generator (cont.)

- The number of levels in hierarchy grows over time.
- This growth models the growth of real-world hypergraphs.



S. Kim, F. Bu, M. Choe, J. Yoo, and K. Shin, “How Transitive Are Real-World Group Interactions? Measurement and Reproduction”, **KDD 2023**

THera: Evaluation

- **THera** reproduces **a realistic level of transitivity** while addressing the obvious limitations of Naïve-THera.

Hypergraph Transitivity

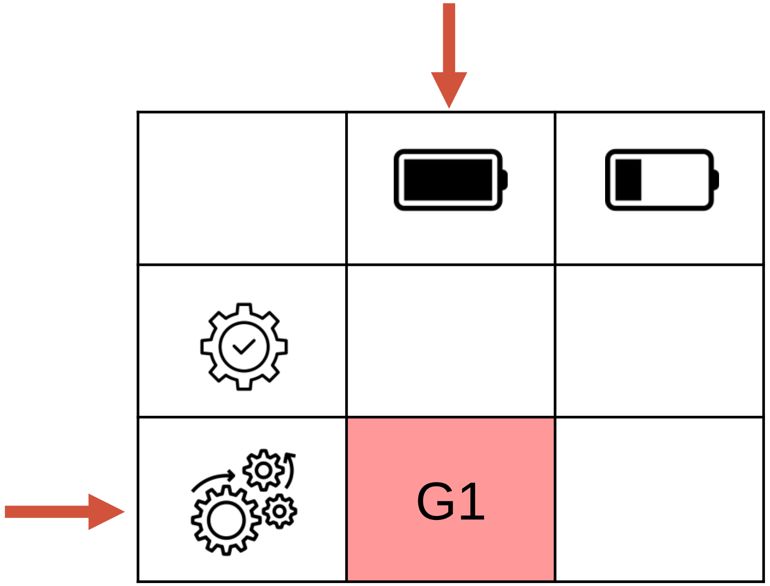
Generator	email		NDC		contact		coauthorship			q&a		
	enron	eu	classes	substances	high	primary	dblp	geology	history	ubuntu	server	math
Real World	0.195	0.125	0.052	0.019	0.345	0.336	0.007	0.005	0.002	0.005	0.005	0.025
THera	0.192	0.124	0.052	0.019	0.344	0.334	0.007	0.005	0.002	0.004	0.004	0.025
HyperCL [33]	0.078	0.053	0.008	0.005	0.119	0.223	0.000*	0.000*	0.000*	0.014	0.017	0.040
HyperPA [16]	0.090	0.110	0.070	-	0.121	0.153	-	-	-	0.003	-	-
HyperFF [30]	0.176	0.125	0.006	0.003	0.006	0.007	0.047	0.048	0.048	0.051	0.050	0.054
HyperLap [33]	0.123	0.085	0.008	0.008	0.220	0.301	0.001	0.000*	0.000*	0.016	0.015	0.004
HyperLap+ [33]	0.231	0.144	0.026	0.016	0.322	0.338	0.042	0.019	0.005	0.029	0.023	0.007

competitors

S. Kim, F. Bu, M. Choe, J. Yoo, and K. Shin, “How Transitive Are Real-World Group Interactions? Measurement and Reproduction”, **KDD 2023**

GLLB23: Dynamic Full Hypergraph Generator

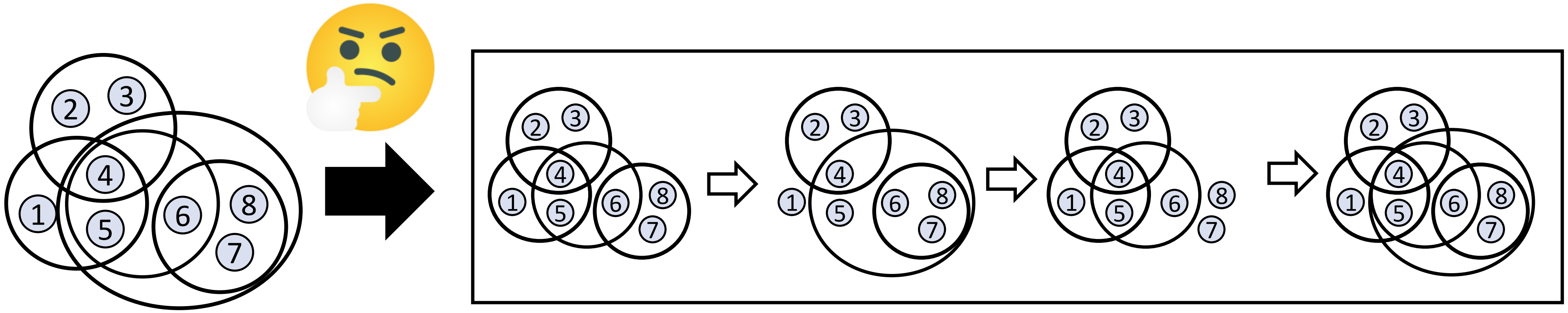
- **G1.** Discrete Auto Regressive Hypergraph (DARH) model
- **G2.** Cross-memory DARH (cDARH) model



Reproduction of Temporal Dynamics

?

Question: Given a set of hyperedges, how can we produce a realistic **sequence of hypergraph snapshots** over time?



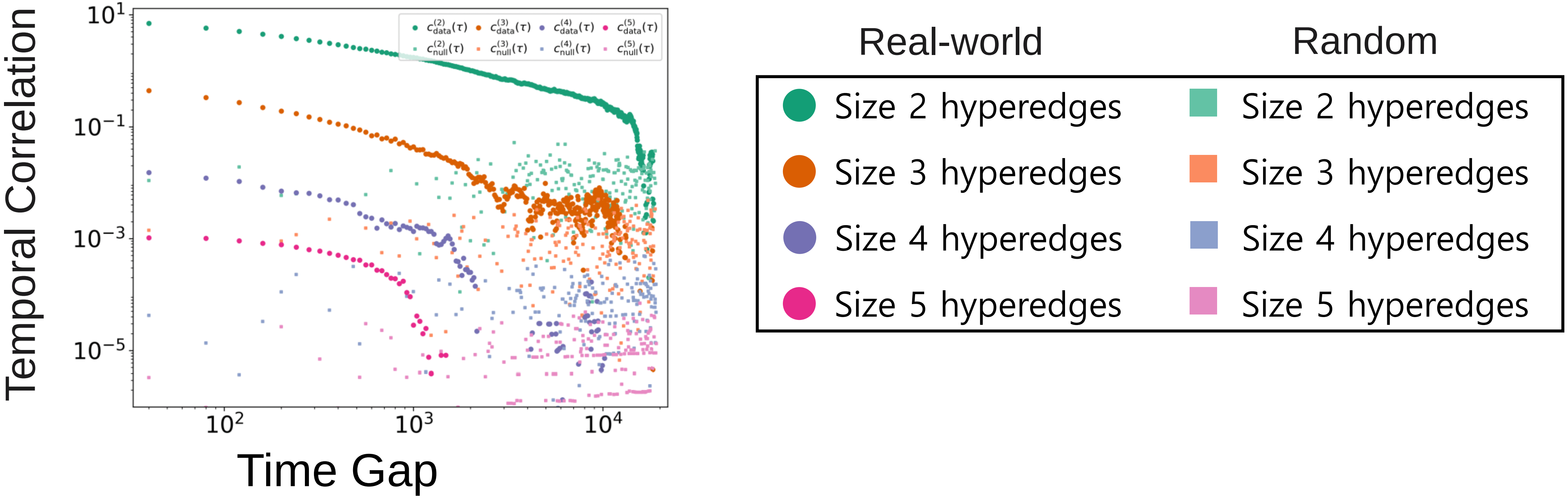
Answer: Exploit intra-order and cross-order correlations in real-world hypergraphs!

!

L. Gallo, L. Lacasa, V. Latora, and F. Battiston, "Higher-order Correlations Reveal Complex Memory in Temporal Hypergraphs", **arXiv (2023)**

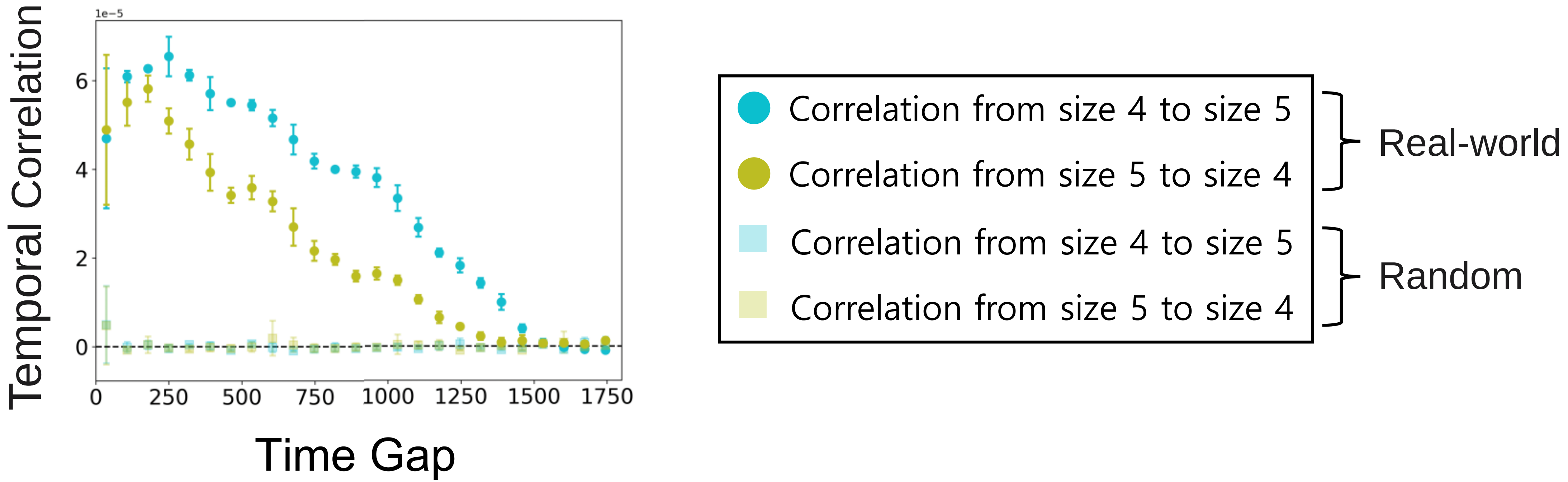
Recap: Empirical Observations

- **Intra-order correlations:** Temporal correlations emerge between hyperedges of the **same sizes**.



Recap: Empirical Observations (cont.)

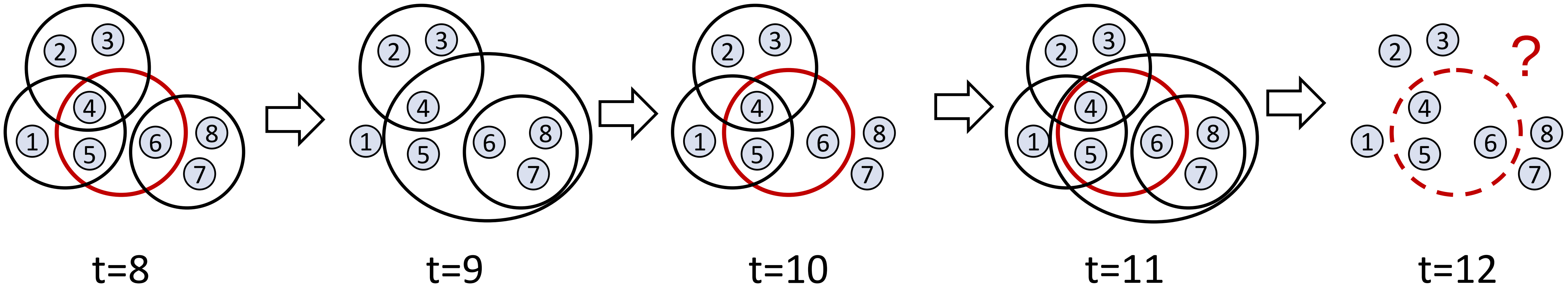
- **Cross-order correlations:** Temporal correlations emerge between hyperedges of **different sizes**.



L. Gallo, L. Lacasa, V. Latora, and F. Battiston, “Higher-order Correlations Reveal Complex Memory in Temporal Hypergraphs”, **arXiv (2023)**

Discrete Auto Regressive Hypergraph

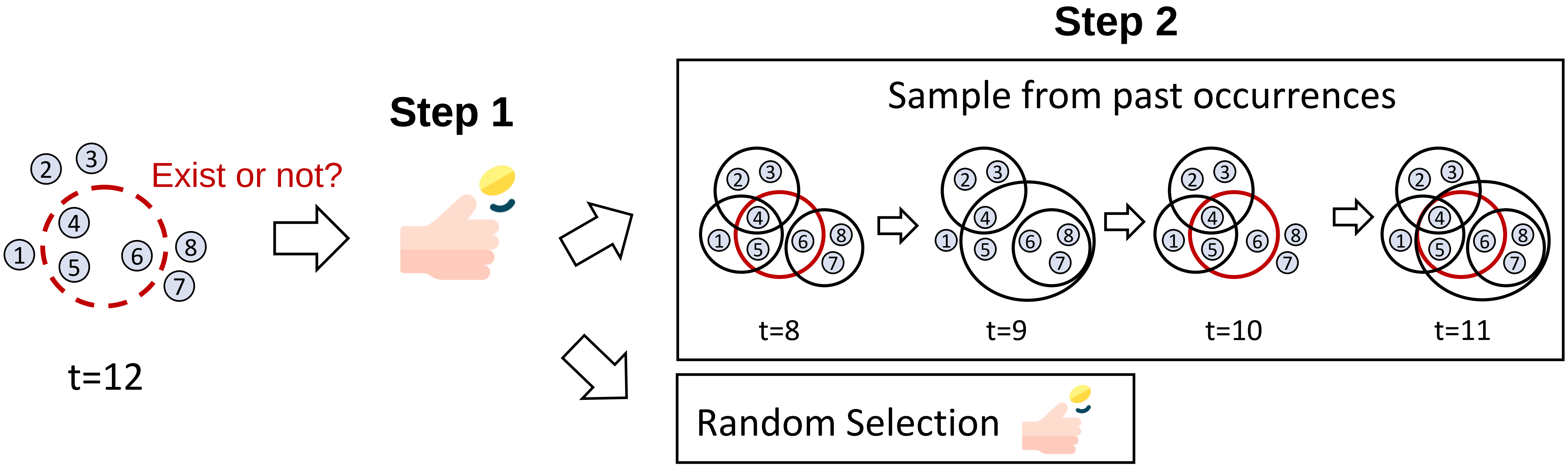
- In **Discrete Auto Regressive Hypergraph (DARH)**, each hyperedge evolves based on an independent stochastic process.
- At each time t, it determines whether each hyperedge occurs or not.



L. Gallo, L. Lacasa, V. Latora, and F. Battiston, "Higher-order Correlations Reveal Complex Memory in Temporal Hypergraphs", **arXiv (2023)**

Discrete Auto Regressive Hypergraph (cont.)

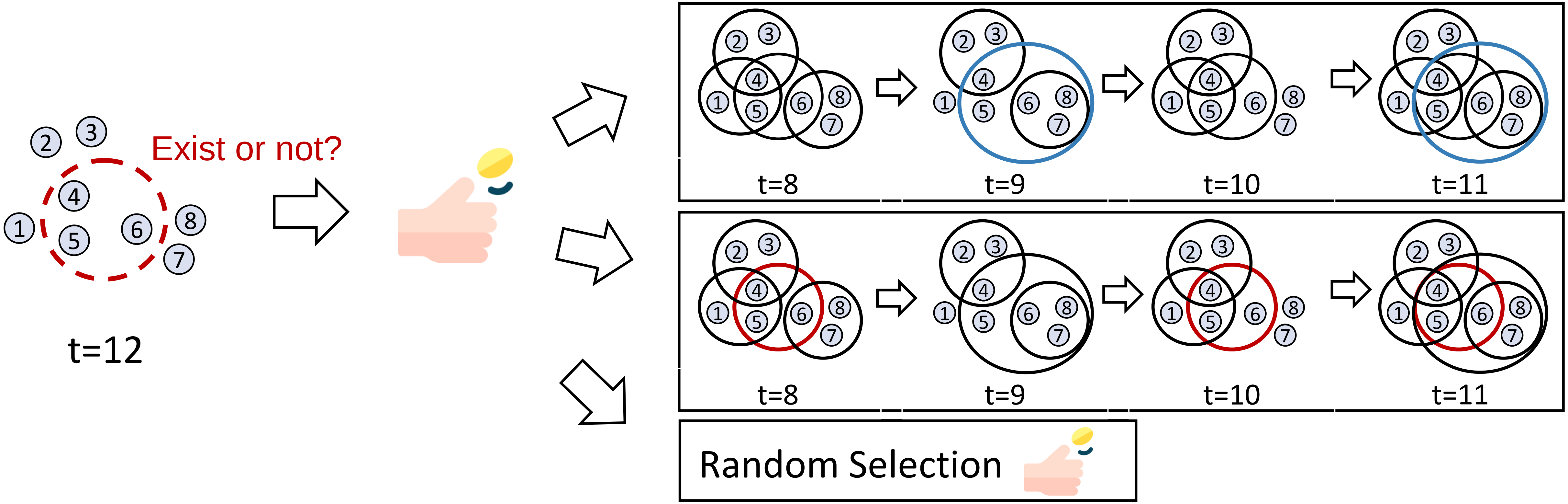
- Step 1. Determine the criterion of the hyperedge's occurrence
- Step 2. Sample from the past occurrences of the hyperedge



L. Gallo, L. Lacasa, V. Latora, and F. Battiston, "Higher-order Correlations Reveal Complex Memory in Temporal Hypergraphs", **arXiv (2023)**

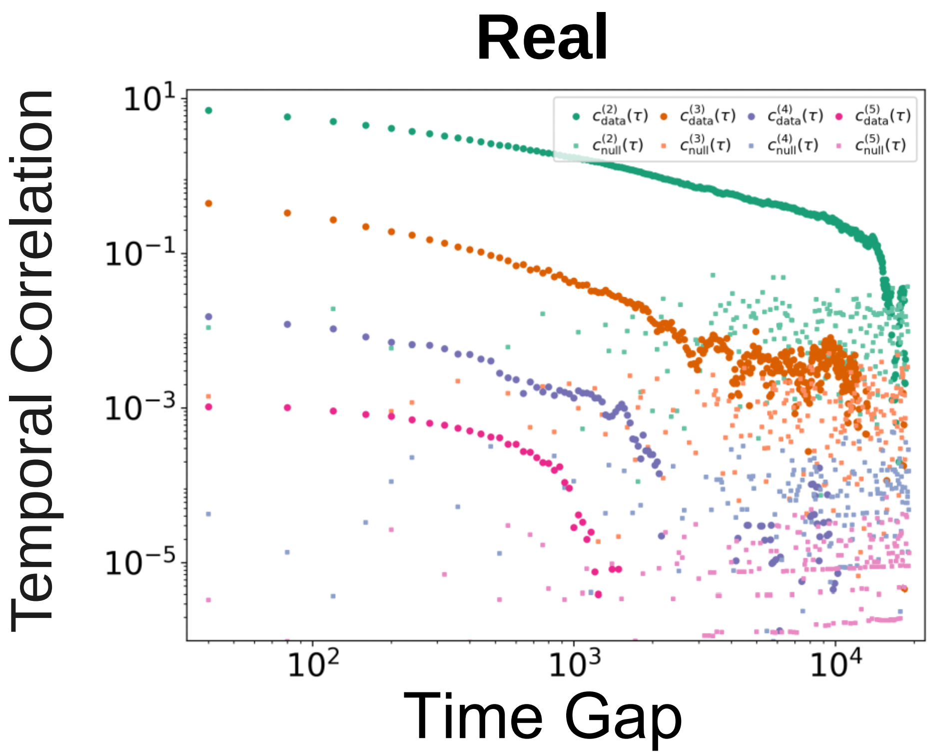
Cross-memory DARH (cDARH) model

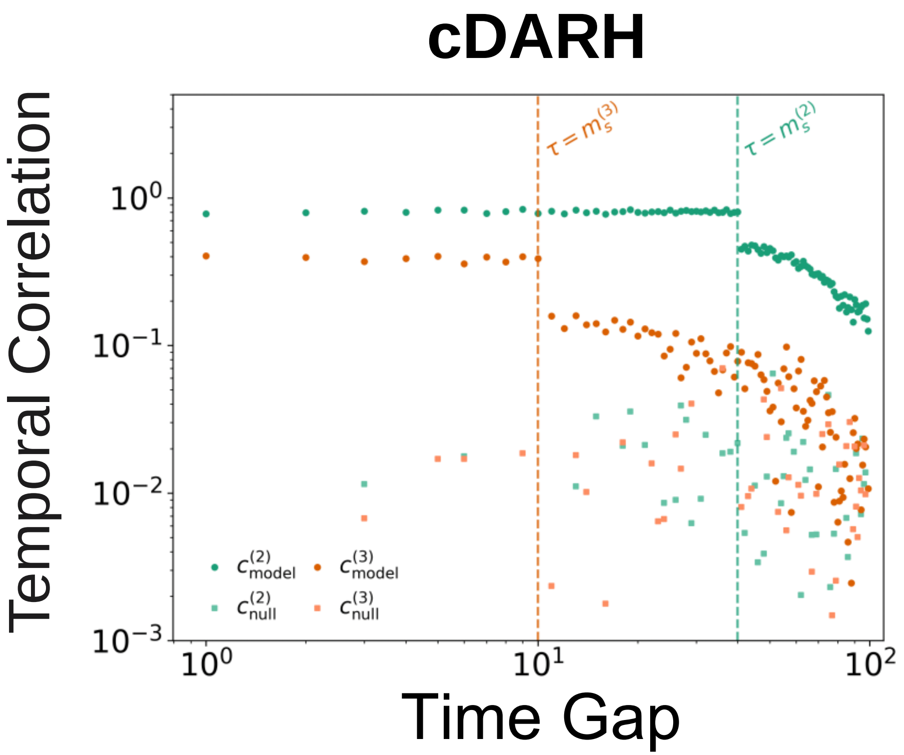
- **cDARH** samples an occurrence also from the occurrences of **supersets & subsets** to model cross-order correlations.



Cross-memory DARH (cDARH) model (cont.)

- **cDARH** reproduces **intra-order** temporal correlations of hyperedges.



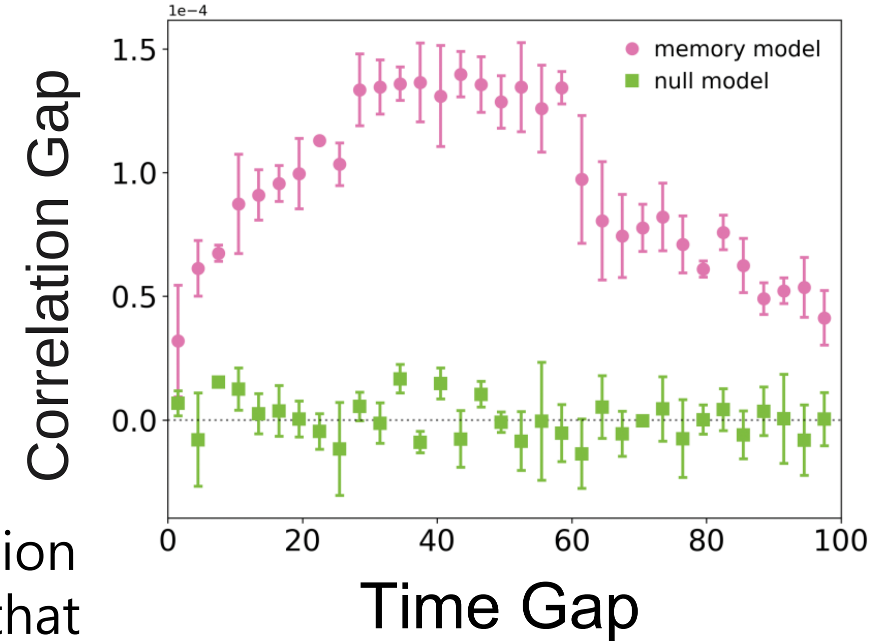


L. Gallo, L. Lacasa, V. Latora, and F. Battiston, "Higher-order Correlations Reveal Complex Memory in Temporal Hypergraphs", **arXiv (2023)**

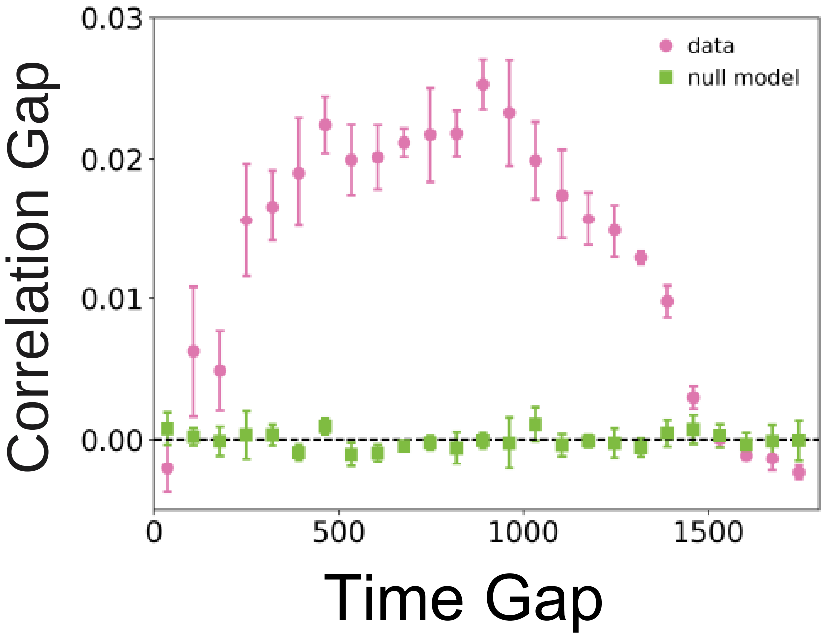
Cross-memory DARH (cDARH) model (cont.)

- **cDARH** reproduces **cross-order** temporal correlations of hyperedges.

Real



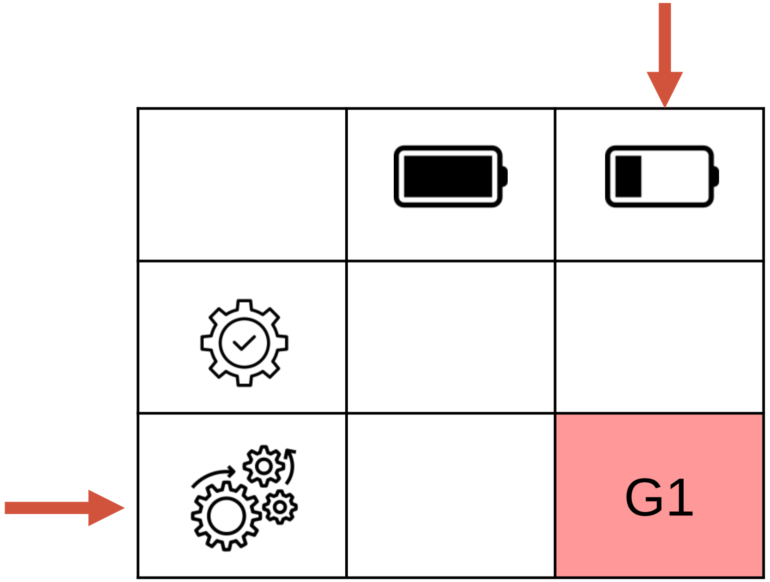
cDARH



Gap between correlation from size 2 to 3 and that from size 3 to 2

BKT18: Dynamic Sub-Hypergraph Generator

- **G1.** Correlated Repeated Unions (CRU) model



Next Hyperedge Prediction

?

Question:

Given a (sub-)sequence of temporal hyperedges, how can we predict the **next hyperedge**?

Answer:

The CRU model predicts the next hyperedge based on three empirical observations: **(1) repeat behavior**, **(2) subset correlation**, and **(3) recency bias**.

!

Recap: Three Empirical Observations

Repeat Behavior

Temporal hyperedges tend to **repeat** previous ones.

Subset Correlation

Subsets of nodes tend to be **correlated**.

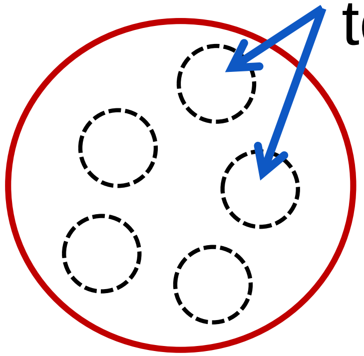
Recency Bias

Temporal hyperedges tend to be similar to **recent** ones.

CRU: Correlated Repeated Unions (cont.)

Step 0. Initialization

Number of nodes to be included: $\boldsymbol{r}$



The next hyperedge

$\boldsymbol{e_{n+1}} = \emptyset$

Recency weight vector $\boldsymbol{w}$

$t-4$	$t-3$	$t-2$	$t-1$
0.08	0.15	0.29	0.48

Correlation probability

$\boldsymbol{p} = 0.80$

Two parameters of CRU

CRU: Correlated Repeated Unions (cont.)

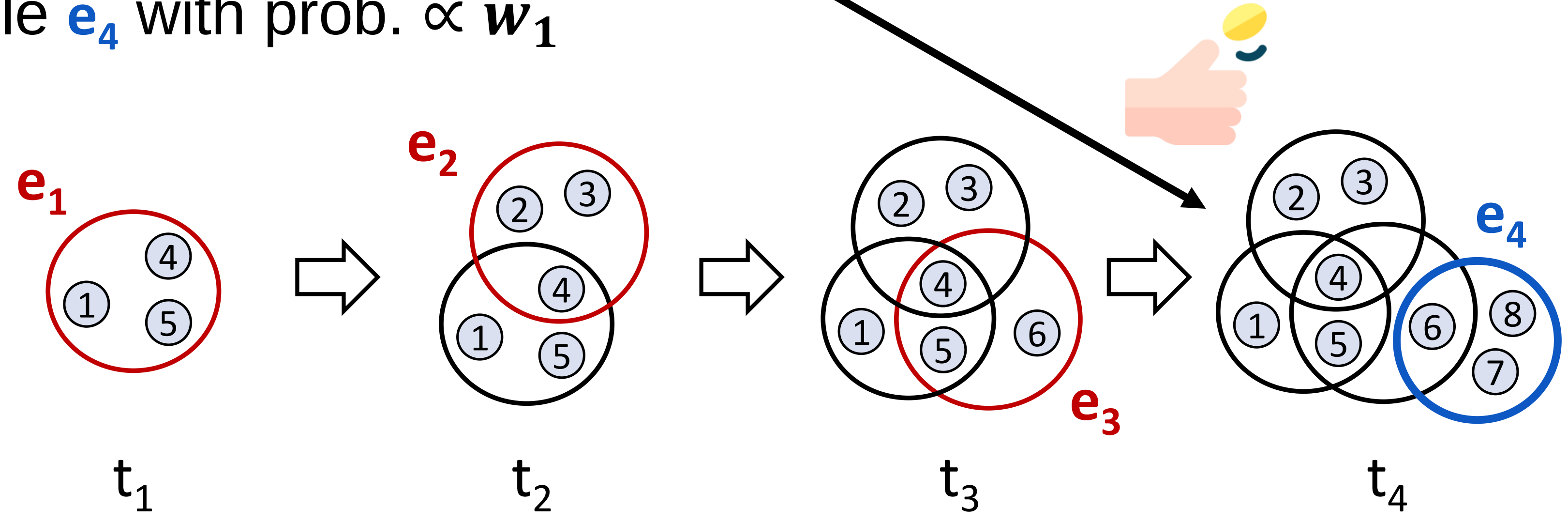
Step 1. Sample an existing hyperedge

$\boldsymbol{w} =$

0.08	0.15	0.29	0.48
$t-1$	$t-2$	$t-3$	$t-4$

Intuition: $\boldsymbol{w}$ controls the **recency bias**.
More skewed toward recent hyperedges
→ *More likely to sample recent hyperedges*

Sample $\mathbf{e_4}$ with prob. $\propto \boldsymbol{w_1}$

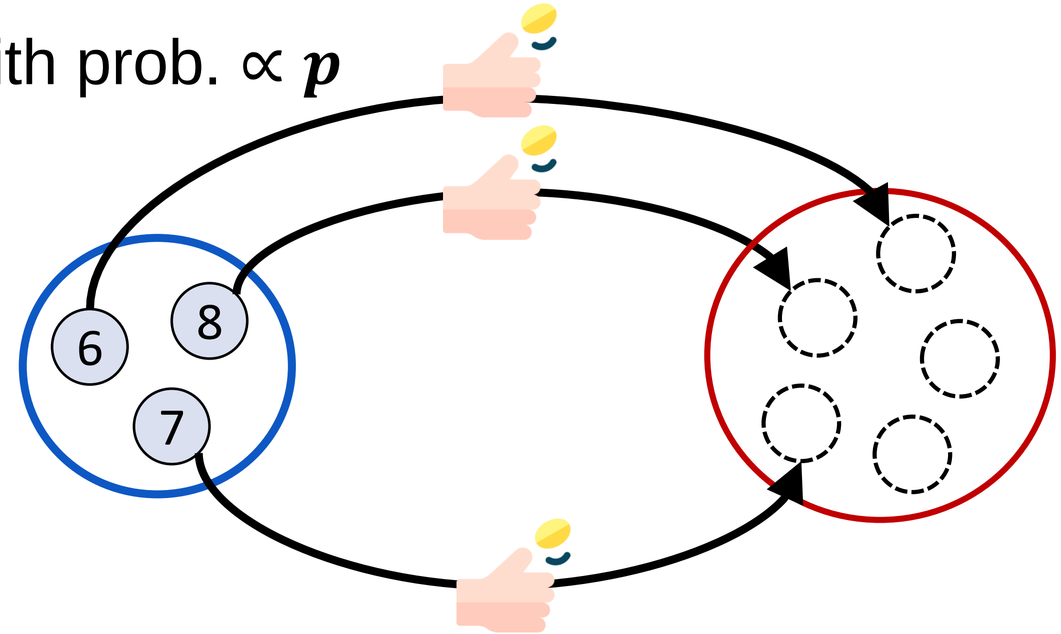


CRU: Correlated Repeated Unions (cont.)

Step 2. Sample nodes

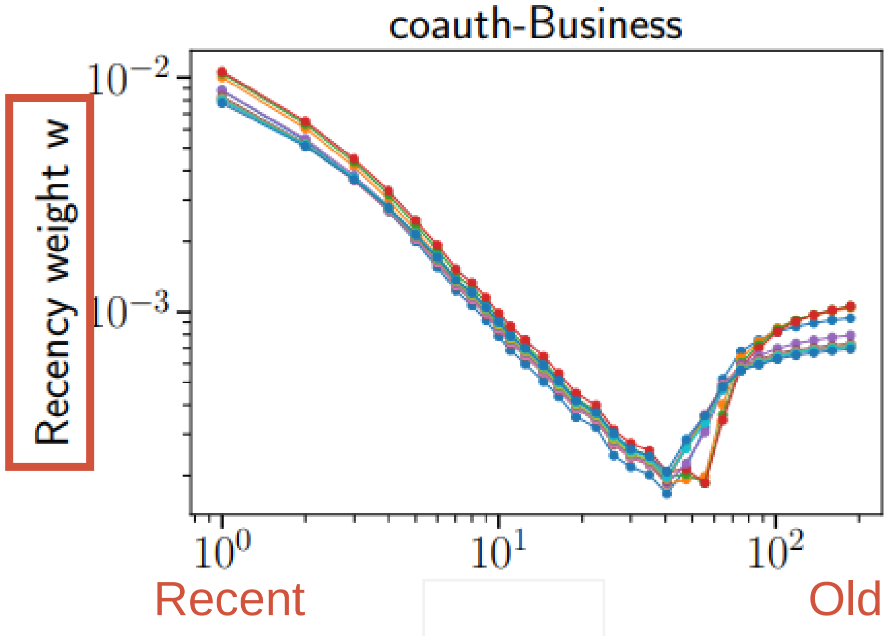
<u>Intuition:</u> $\boldsymbol{p}$ controls the **subset correlation**.
A larger $\boldsymbol{p}$ → *More correlation in selecting items from the same hyperedge*

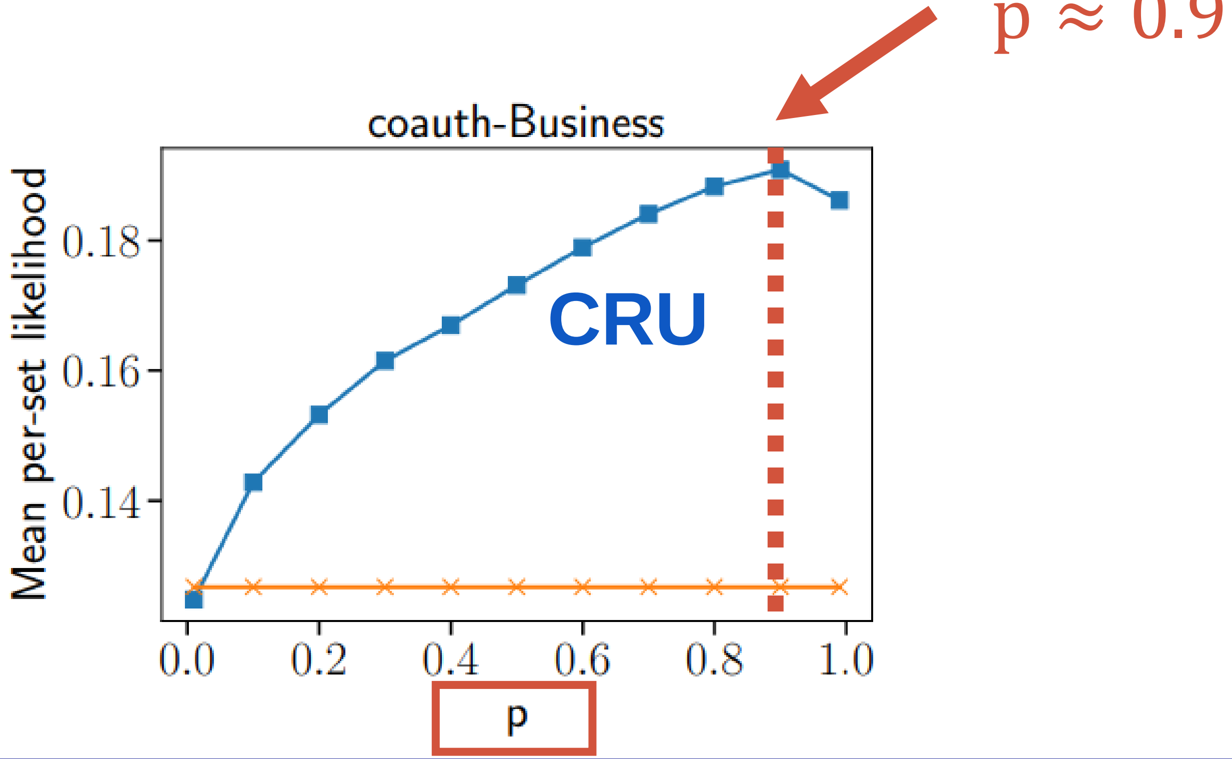
Sample node with prob. $\propto \boldsymbol{p}$



CRU: Trained Parameters

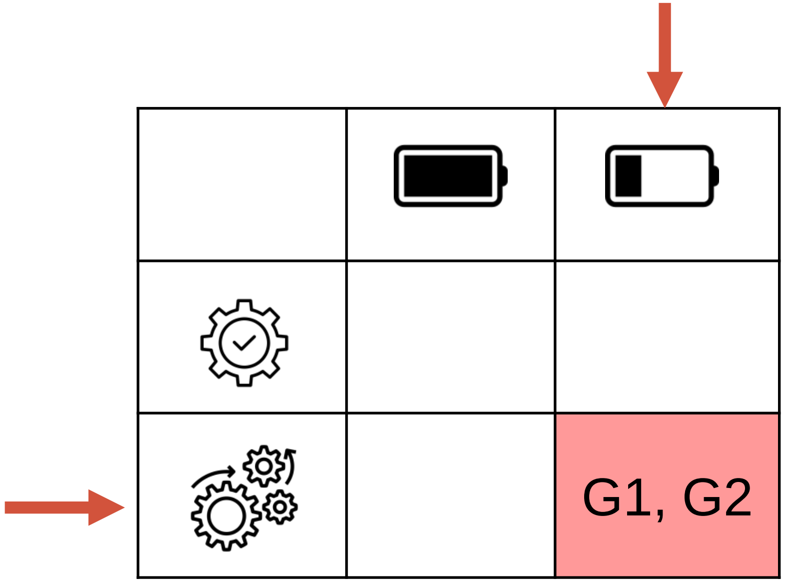
- Correlation probability p and recency weight vector w are trained
- **Recency bias** and **subset correlation** are observed





CK21: Dynamic Sub-Hypergraph Generator

- **G1.** Temporal order prediction model
- **G2.** Temporal reconstruction model



Task 1: Temporal Order Prediction

?

Question:

Has the given dynamic ego-network **evolved** reasonably?

Answer:

- A supervised **binary classification** task is defined.
- Are the hyperedges in a given dynamic ego-network **correctly** or **randomly** ordered?

!

Corrected ordered ego-network **Randomly ordered ego-network**

Recap: Four Empirical Observations

Intersection Size of Ego-networks

Temporally adjacent hyperedges in ego-networks are similar.

Spread of Alter-networks

Spread of alter-networks are temporally local.

Anthropic Principle of Ego-networks

The arrival of ego-nodes occurs after pre-dated hyperedges.

Novelty of Ego-networks

Novelty decreases in ego-networks.

C. Comrie, J. Kleinberg, "Hypergraph Ego-networks and Their Temporal Evolution", **ICDM 2021**

Temporal Order Prediction (cont.)

- We extract six features based on the patterns to train **a classifier.**
- Compared to **random guessing (baseline)**, the **trained classifier** significantly outperforms on all datasets and ego-network types.

	Star ego-network		Radial ego-network		Contracted ego-network	
	Random	**Proposed**	Random	**Proposed**	Random	**Proposed**
coauth-DBLP	0.50	**0.93 $\pm$ 0.01**	0.50	**0.91 $\pm$ 0.01**	0.50	**0.85 $\pm$ 0.01**
email-Avocado	0.50	**0.84 $\pm$ 0.09**	-	-	-	-
threads-ask-ubuntu	0.50	**0.72 $\pm$ 0.05**	-	-	-	-

Omitted due to their sizes

Classification accuracy

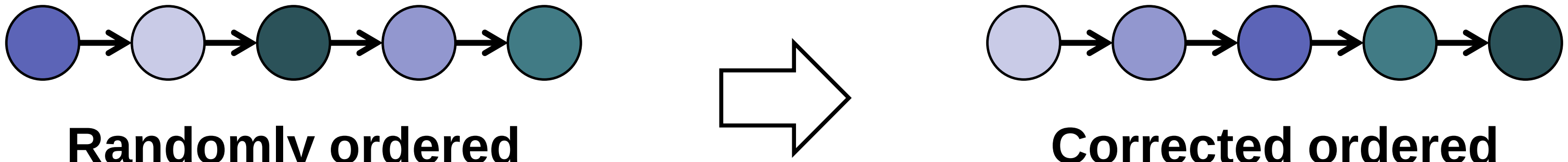
Task 2: Temporal Reconstruction

Question:

How can we properly **reconstruct** the temporal order of hyperedges in the given ego-network?

Answer:

A local search algorithm is used to iteratively update the temporal order

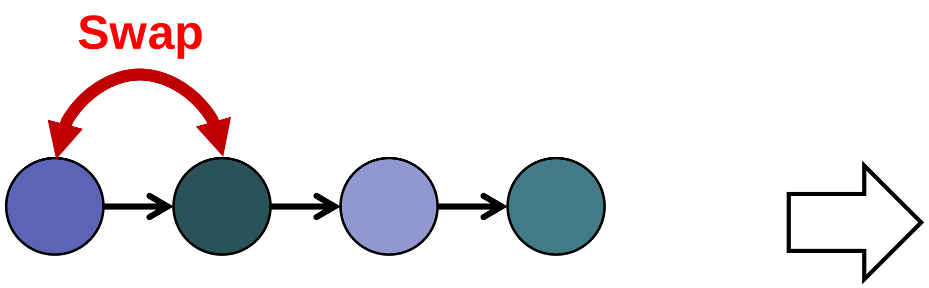


Randomly ordered ego-network

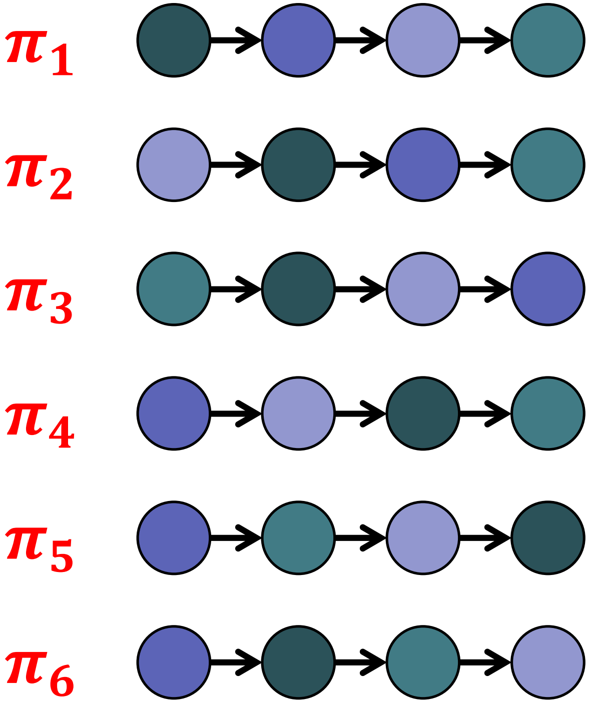
Corrected ordered ego-network

Temporal Reconstruction (cont.)

Step 1. Swap pairs



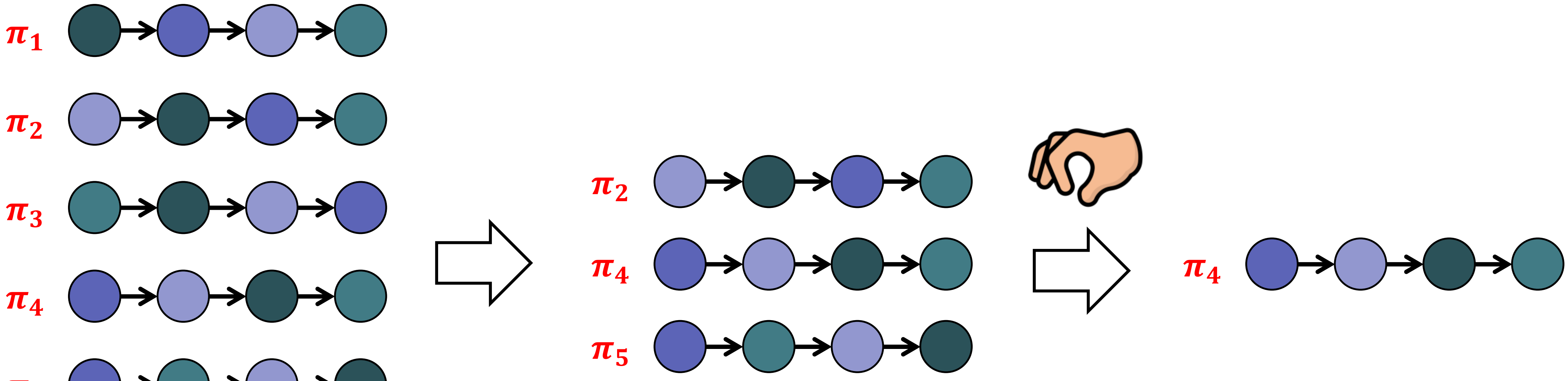
Randomly ordered ego-network π



All possible swaps

Temporal Reconstruction (cont.)

Step 2. Predict the order based on Fitness ($\mathcal{M}$: Model for Task 1)



All possible swaps

Filtered orders π_i such that $\mathcal{M}(\pi_i) > \mathcal{M}(\pi)$

Random sampling

Repeat steps (1) – (2) until convergence

Temporal Reconstruction (cont.)

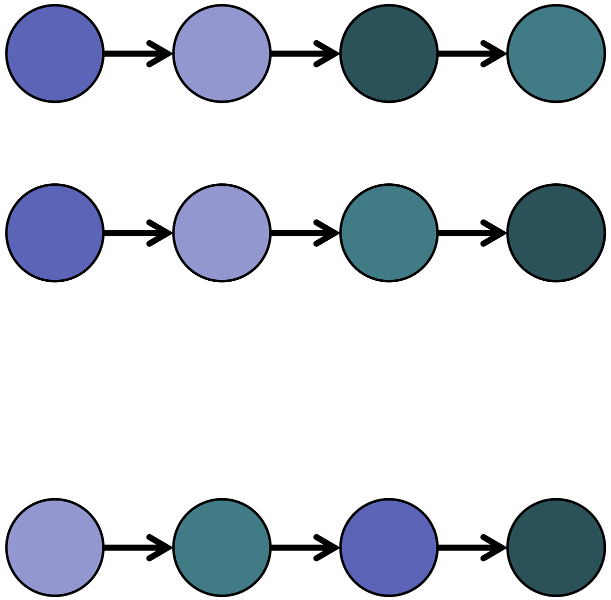
Step 3. Multiple trials

Trial 1: $\pi^{(1)}$

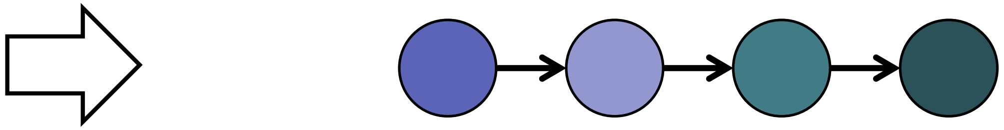
Trial 2: $\pi^{(2)}$

⋮

Trial T : $\pi^{(T)}$



Best orders from each trial



Select the best one:
$\max_{1\le t\le T}\mathcal{M}\left(\pi^{(t)}\right)$

Final output

Temporal Reconstruction (cont.)

- The proposed algorithm shows a **non-trivial improvement** over random guessing (baseline).

	Star ego-network		Radial ego-network		Contracted ego-network	
	Random	**Proposed**	Random	**Proposed**	Random	**Proposed**
coauth-DBLP	0.50	**0.65 $\pm$ 0.08**	0.50	**0.56 $\pm$ 0.05**	0.50	**0.65 $\pm$ 0.08**
email-Avocado	0.50	**0.63 $\pm$ 0.11**	-	-	-	-
threads-ask-ubuntu	0.50	**0.70 $\pm$ 0.07**	-	-	-	-

Omitted due to their sizes

Reconstruction accuracy, ***i.e., the ratio of corrected predicted pairs of hyperedges***

Part 3. Hypergraph Generative Models

		Part 3. Generative Models
Static Models	Full-Hypergraphs	C20, LCS21
	Sub-Hypergraphs	CYLBKS22
Dynamic Models	Full-Hypergraphs	DYHS20, KKS20, KBCYS23, GLLB23
	Sub-Hypergraphs	BKT18, CK21

Conclusions

- Hypergraph modeling of group interactions
  - Provides a new perspective ("set of sets")
  - Reveals new structural patterns that are previously overlooked
- Hypergraph mining tools and measures
  - Many tools (e.g., hypergraph motifs) and measures are available
- Hypergraph generative models
  - Shed a light on why the patterns occur
  - Applications: statistical testing, anonymization, benchmark data generation

Future Research Directions

- Mining of directed hypergraphs (e.g., chemical reactions)
  - How do real-world directed hypergraphs look and evolve time?
- Hypergraph representation learning
  - How can we embed hyperedges while preserving their structural properties?
- Anomaly detection
  - How can we identify abnormal group interactions?

Tutorial Materials

- https://sites.google.com/view/hypergraph-tutorial
  - Slides
  - Videos
  - Code and Datasets

References

1. [AA02] Albert, Réka, and Albert-László Barabási. "Statistical mechanics of complex networks." Reviews of modern physics 74.1 (2002): 47.
2. [BKT18] Benson, Austin R., Ravi Kumar, and Andrew Tomkins, “Sequences of Sets.” KDD 2018.
3. [CK21] Comrie, Cazamere, and Jon Kleinberg. “Hypergraph Ego-networks and Their Temporal Evolution.” ICDM 2021.
4. [CYLBKS22] Choe, Minyoung, et al. “MiDaS: Representative Sampling from Real-world Hypergraphs.” WWW 2022.
5. [DYHS20] Do, Manh Tuan, et al. “Structural Patterns and Generative Models of Real-world Hypergraphs.” KDD 2020.
6. [GLLB23] Gallo, Luca et al. “Higher-order Correlations Reveal Complex Memory in Temporal Hypergraphs”, arXiv 2023.

References (cont.)

7. [KBCYS23] Kim, Sunwoo et al. “How Transitive Are Real-World Group Interactions? - Measurement and Reproduction.” KDD 2023.
8. [KKS20] Kook, Yunbum, Jihoon Ko, and Kijung Shin. “Evolution of Real-world Hypergraphs: Patterns and Models without Oracles.” ICDM 2020.
9. [KKS22] Ko, Jihoon, Yunbum Kook, and Kijung Shin. "Growth patterns and models of real-world hypergraphs." Knowledge and Information Systems 64.11 (2022): 2883-2920.
10. [LCS21] Lee, Geon, Minyoung Choe, and Kijung Shin. “How Do Hyperedges Overlap in Real-world Hypergraphs? – Patterns, Measures, and Generators." WWW 2021.
11. [LKF05] Leskovec, Jure, Jon Kleinberg, and Christos Faloutsos. "Graph evolution: Densification and shrinking diameters." KDD 2005