ICDM 2025
November 12–15, 2025
Washington DC, USA
IEEE International Conference on Data Mining

KAIST

Attributed Hypergraph Generation with Realistic Interplay Between Structure and Attributes

Best Paper Award


Jaewan Chun*


Seokbum Yoon*


Minyoung Choe


Geon Lee


Kijung Shin

Group Interactions are Everywhere

- Various types of group interactions are observed in the real world.

  - Example 1. Co-authorship of researchers

Random walk with restart on hypergraphs: fast computation and an application to anomaly detection

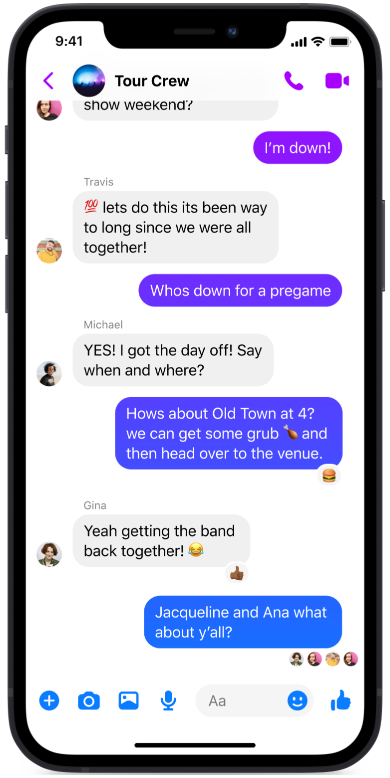
Jaewan Chun[1] · Geon Lee[1] · Kijung Shin[1] · Jinhong Jung[2]

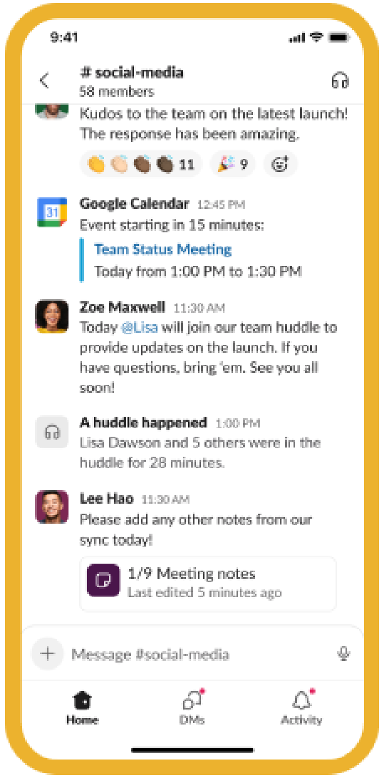
How Do Hyperedges Overlap in Real-World Hypergraphs? - Patterns, Measures, and Generators

Geon Lee*	Minyoung Choe*	Kijung Shin
KAIST AI	KAIST AI	KAIST AI & EE
Daejeon, South Korea	Daejeon, South Korea	Daejeon, South Korea
geonlee0325@kaist.ac.kr	minyoung.choe@kaist.ac.kr	kijungs@kaist.ac.kr

Group Interactions are Everywhere (cont.)

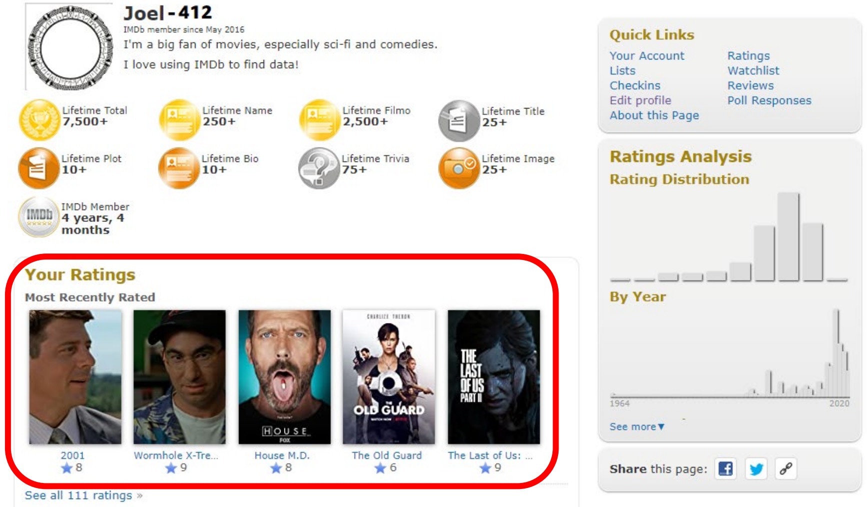
- Various types of group interactions are observed in the real world.
  - Example 2. Online group chats

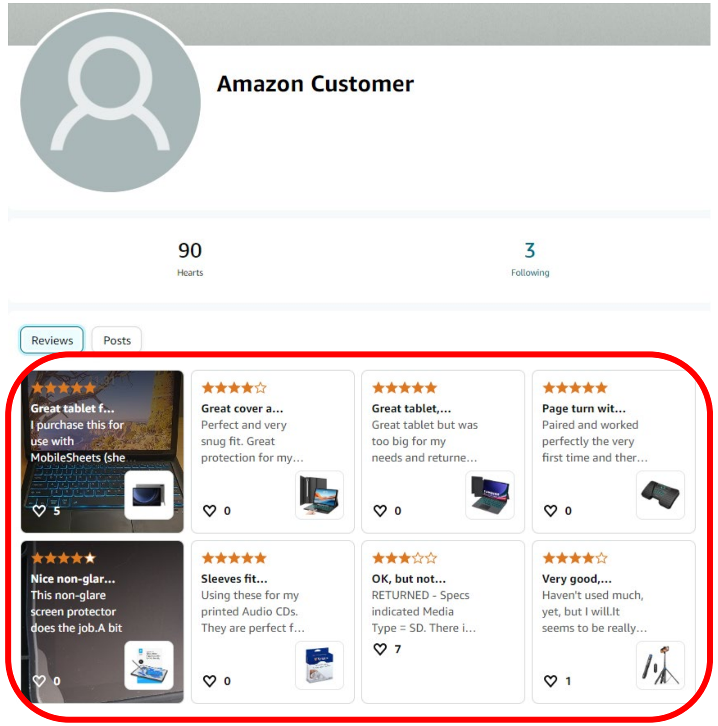




Group Interactions are Everywhere (cont.)

- Various types of group interactions are observed in the real world.
  - Example 3. Reviewed item set





Hypergraphs: Model for Group Interactions

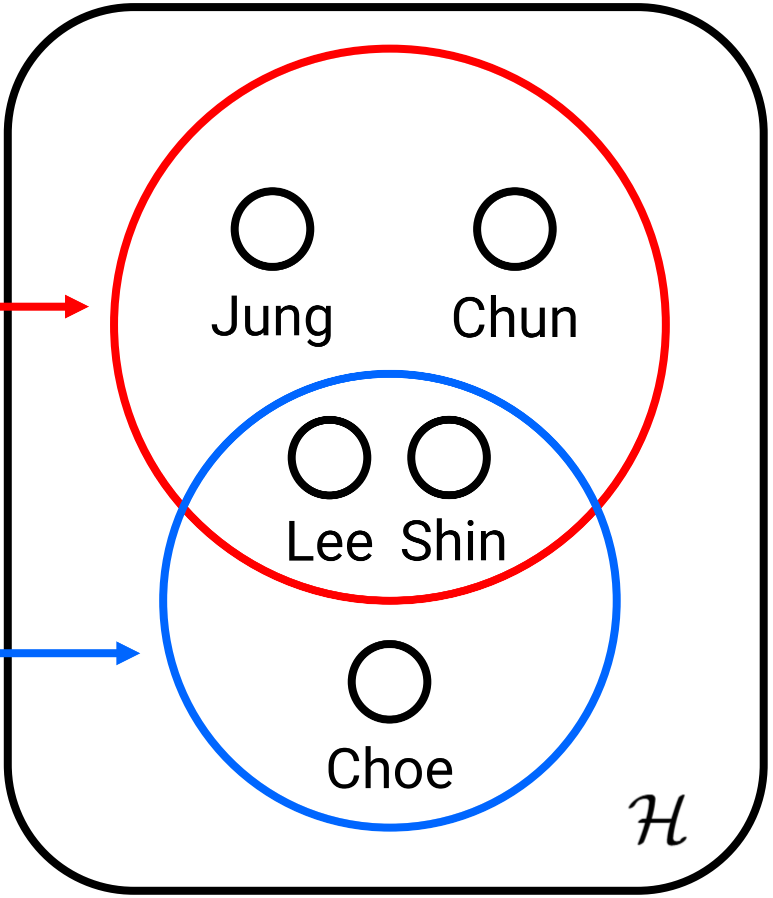
- **Hypergraphs** model group interactions among individuals or objects using hyperedges.

Random walk with restart on hypergraphs: fast computation and an application to anomaly detection

Jaewan Chun[1] · Geon Lee[1] · Kijung Shin[1] · Jinhong Jung[2]

How Do Hyperedges Overlap in Real-World Hypergraphs? - Patterns, Measures, and Generators

Geon Lee*	Minyoung Choe*	Kijung Shin
KAIST AI	KAIST AI	KAIST AI & EE
Daejeon, South Korea	Daejeon, South Korea	Daejeon, South Korea
geonlee0325@kaist.ac.kr	minyoung.choe@kaist.ac.kr	kijungs@kaist.ac.kr



Hypergraph Generative Models

- In real-world hypergraphs, hyperedges are formed in a systematic and realistic manner.

- Hypergraph generative models are tools to generate realistic synthetic hypergraphs.

- They explicitly incorporate mechanisms underlying hyperedge formation:
  - HyperPA [3] generalizes the preferential attachment model from graphs to hypergraphs.
  - HyperSBM [11] generalizes the stochastic block model (SBM) from graphs to hypergraphs.
  - HyperLAP [1] and Thera [2] model unique characteristic of real-world hypergraphs.

Hypergraph Generative Models

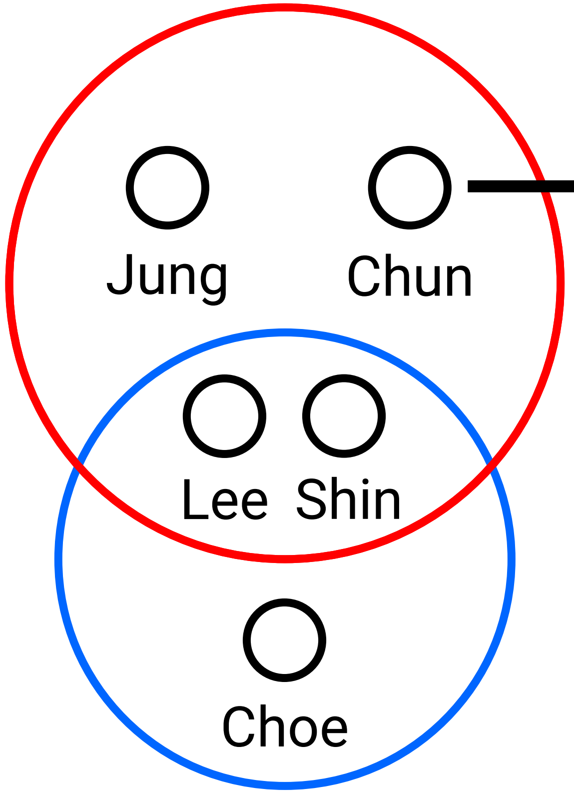
- In real-world hypergraphs, hyperedges are formed in a systematic and realistic manner.

- Hypergraph generative models are tools to generate realistic synthetic hypergraphs.

- Applications of hypergraph generative models:
    - Community detection [11]
    - Hyperedge prediction [19]
    - Pattern discovery [20]

Node Attributes: Information Associated with Each Node

- **Node attributes** are the properties or characteristics of each node.

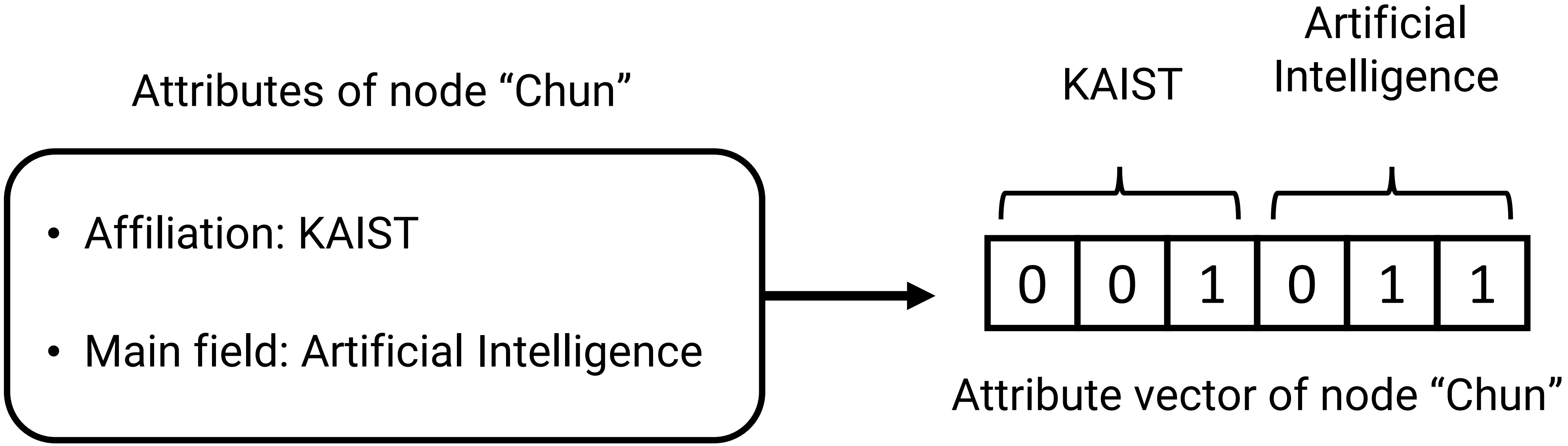


Attributes of node "Chun"

- Affiliation: KAIST
- Main field: Artificial Intelligence

Node Attributes: Information Associated with Each Node (cont.)

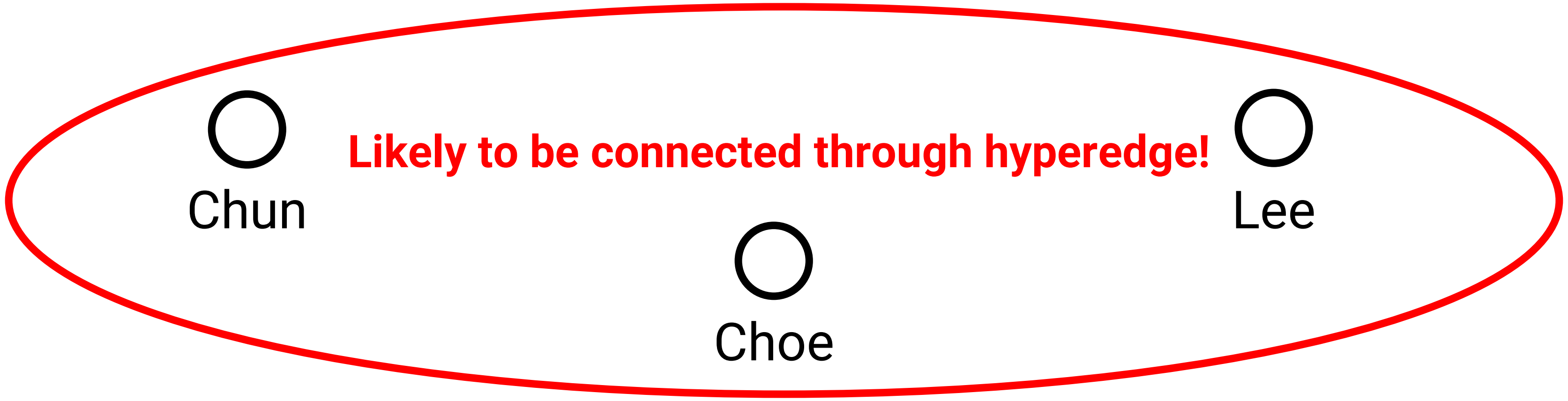
- **Node attributes** are the properties or characteristics of each node.



- In this work, we use binary attribute vector, since we can convert other types to binary:
  - Continuous → thresholding
  - Categorical → one-hot encoding

Node Attributes: Information Associated with Each Node (cont.)

- **Node attributes** may influence hyperedge formation (e.g. Homophily).



Shared attribute of Chun, Choe, and Lee

- Affiliation: KAIST
- Main field: Artificial Intelligence

Hypergraph Generative Model Based on Node Attributes

- However, most of hypergraph generative models overlook node attributes.
  - They are structure-based, with attribute-independent generation mechanisms.

- In this work, we propose **NoAH**, **No**de **A**ttribute based **H**ypergraph generator.
  - **NoAH** produces realistic hypergraph using **node attributes**.
  - We propose **NoAHFit**, a **parameter fitting algorithm** for NoAH.

Hypergraph Generative Model Based on Node Attributes

- However, most of hypergraph generative models overlook node attributes.
    - They are structure-based, with attribute-independent generation mechanisms.

- In this work, we propose **NoAH**, **No**de **A**ttribute based **H**ypergraph generator.
    - **NoAH** produces realistic hypergraph using **node attributes**.
    - We propose **NoAHFit**, a **parameter fitting algorithm** for NoAH.
    - NoAH (fitted by NoAHFit) experimentally show:
        - Superior modeling of **structure-attribute interplay** compared to existing generators.
        - **Scalability** with the number of hyperedges and attributes.

Roadmap

- Overview
- Preliminaries <<
- Proposed Method: NoAH, NoAHFit
- Experiments
- Conclusion

Preliminary Model: Multiplicative Attributed Graph (MAG) Model

- There exist attributed-based models for pairwise graphs.

- Particularly, the **Multiplicative Attributed Graph (MAG) model** [6] aims to capture how node attributes affect edge formation in pairwise graphs.

Multiplicative Attributed Graph (MAG) Model

- For each attribute dimension, MAG defines an **affinity matrix**.
  - Each entry quantifies **edge formation probability** based on **attribute value pair**.

$$
\left\{
\begin{array}{c}
\theta_1 \\
\begin{array}{cc}
 & \begin{array}{cc} 0 & 1 \end{array} \\
\begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.9 & 0.5 \\ 0.5 & 0.9 \end{bmatrix}
\end{array}
\end{array}
\quad
\begin{array}{c}
\theta_2 \\
\begin{array}{cc}
 & \begin{array}{cc} 0 & 1 \end{array} \\
\begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.8 & 0.3 \\ 0.3 & 0.2 \end{bmatrix}
\end{array}
\end{array}
\quad
\begin{array}{c}
\theta_3 \\
\begin{array}{cc}
 & \begin{array}{cc} 0 & 1 \end{array} \\
\begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.7 & 0.6 \\ 0.6 & 0.9 \end{bmatrix}
\end{array}
\end{array}
\quad
\begin{array}{c}
\theta_4 \\
\begin{array}{cc}
 & \begin{array}{cc} 0 & 1 \end{array} \\
\begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.8 & 0.9 \\ 0.9 & 0.4 \end{bmatrix}
\end{array}
\end{array}
\right\}
$$

Affinity Matrices

Multiplicative Attributed Graph (MAG) Model

- For each attribute dimension, MAG defines an **affinity matrix**.
  - Each entry quantifies **edge formation probability** based on **attribute value pair**.

Affinity matrix for dimension 1

$$\left\{ \boxed{\theta_1} \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.9 & 0.5 \\ 1 & 0.5 & 0.9 \end{array} \quad \theta_2 \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.3 \\ 1 & 0.3 & 0.2 \end{array} \quad \theta_3 \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.7 & 0.6 \\ 1 & 0.6 & 0.9 \end{array} \quad \theta_4 \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.9 \\ 1 & 0.9 & 0.4 \end{array} \right\}$$

Affinity Matrices

Multiplicative Attributed Graph (MAG) Model (cont.)

- **Extract entries** from the affinity matrices based on **attribute value pair**.

Node u — Attribute vector of u: $\begin{bmatrix} \color{red}{1} & 0 & 1 & 0 \end{bmatrix}$

Node v — Attribute vector of v: $\begin{bmatrix} \color{red}{0} & 1 & 1 & 0 \end{bmatrix}$

$$\left\{ \theta_1 = \begin{array}{c|cc} & \color{red}{0} & 1 \\ \hline 0 & 0.9 & 0.5 \\ \color{red}{1} & \color{red}{0.5} & 0.9 \end{array}, \quad \theta_2 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.3 \\ 1 & 0.3 & 0.2 \end{array}, \quad \theta_3 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.7 & 0.6 \\ 1 & 0.6 & 0.9 \end{array}, \quad \theta_4 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.9 \\ 1 & 0.9 & 0.4 \end{array} \right\}$$

Affinity Matrices

$\color{red}{0.5}$

Multiplicative Attributed Graph (MAG) Model (cont.)

- **Extract entries** from the affinity matrices based on **attribute value pair**.

Node u

1	0	1	0

Attribute vector of u

Node v

0	1	1	0

Attribute vector of v

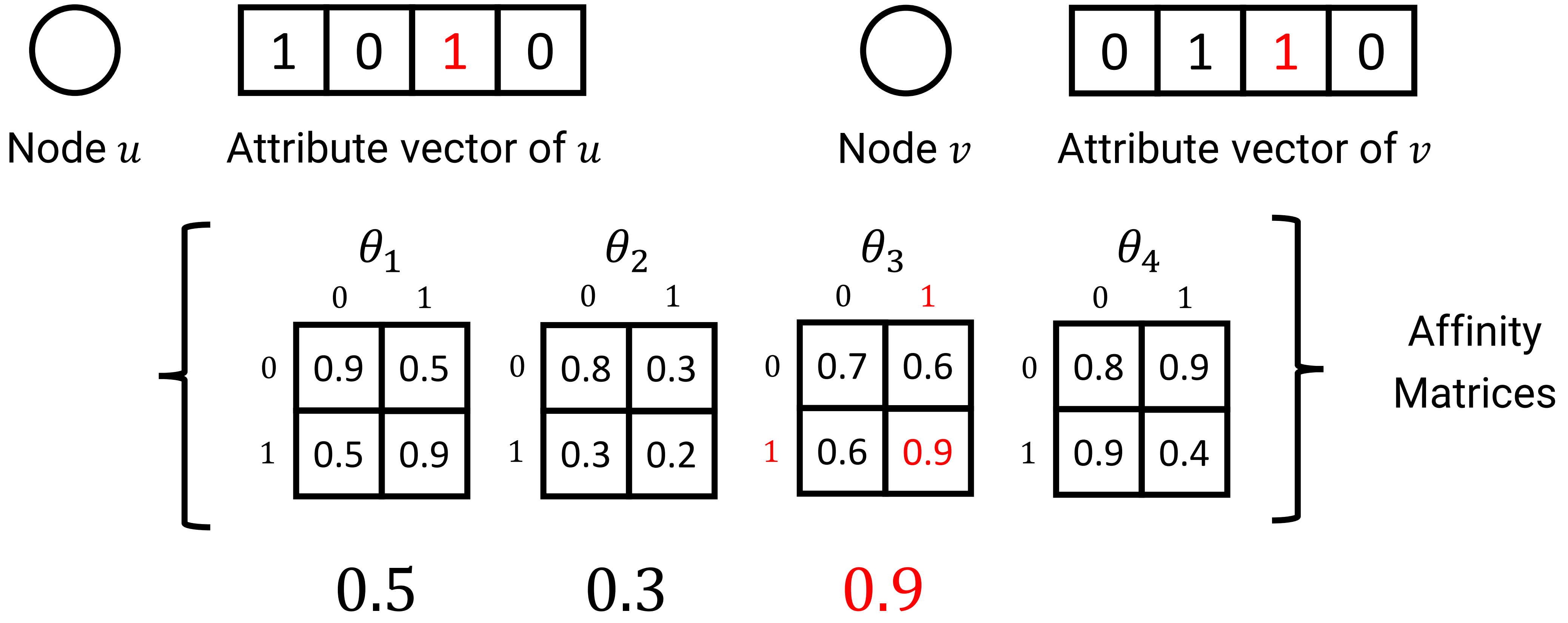
$$\left\{ \theta_1 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.9 & 0.5 \\ 1 & 0.5 & 0.9 \end{array} \quad \theta_2 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.3 \\ 1 & 0.3 & 0.2 \end{array} \quad \theta_3 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.7 & 0.6 \\ 1 & 0.6 & 0.9 \end{array} \quad \theta_4 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.9 \\ 1 & 0.9 & 0.4 \end{array} \right\}$$

Affinity Matrices

$0.5 \quad 0.3$

Multiplicative Attributed Graph (MAG) Model (cont.)

- **Extract entries** from the affinity matrices based on **attribute value pair**.



Multiplicative Attributed Graph (MAG) Model (cont.)

- **Extract entries** from the affinity matrices based on **attribute value pair**.

1	0	1	0

Node u Attribute vector of u

0	1	1	0

Node v Attribute vector of v

$$\left\{ \begin{array}{cccc} \theta_1 & \theta_2 & \theta_3 & \theta_4 \\ \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.9 & 0.5 \\ 1 & 0.5 & 0.9 \end{array} & \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.3 \\ 1 & 0.3 & 0.2 \end{array} & \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.7 & 0.6 \\ 1 & 0.6 & 0.9 \end{array} & \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.9 \\ 1 & 0.9 & 0.4 \end{array} \end{array} \right\}$$

Affinity Matrices

0.5 0.3 0.9 0.8

Multiplicative Attributed Graph (MAG) Model (cont.)

- **Edge probability** between two nodes: **multiplication of entries** from the affinity matrix.

1	0	1	0

Node u — Attribute vector of u

0	1	1	0

Node v — Attribute vector of v

$$\underbrace{\left\{ \theta_1 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.9 & 0.5 \\ 1 & 0.5 & 0.9 \end{array} \quad \theta_2 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.3 \\ 1 & 0.3 & 0.2 \end{array} \quad \theta_3 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.7 & 0.6 \\ 1 & 0.6 & 0.9 \end{array} \quad \theta_4 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.8 & 0.9 \\ 1 & 0.9 & 0.4 \end{array} \right\}}_{\text{Affinity Matrices}}$$

$$P(u,v) = 0.5 \times 0.3 \times 0.9 \times 0.8 = 0.108$$

Multiplicative Attributed Graph (MAG) Model (cont.)

- **Edge probability** between two nodes: **multiplication of entries** from the affinity matrix.

$$P(u, v) = \prod_{l=1}^{k} \boldsymbol{\theta}_l[\mathbf{x}_u^{(l)}, \mathbf{x}_v^{(l)}]$$

- k: the dimension of node attributes
- $\mathbf{x}_u$: attribute of node u

Measures for Structure-Attribute Interplay

- Recall) Node attributes may influence hyperedge formation (e.g. Homophily).

Measures for Structure-Attribute Interplay (cont.)

- Recall) Node attributes may influence hyperedge formation (e.g. Homophily).

- **Q.** How can we **quantify the interplay** between **structure and attributes**?

Measures for Structure-Attribute Interplay (cont.)

- Recall) Node attributes may influence hyperedge formation (e.g. Homophily).

- **Q.** How can we **quantify the interplay** between **structure and attributes**?

- **A.** We utilize three measures, which can be categorized into:
  1. **Hyperedge-level** measures (affinity ratio scores and hyperedge entropy)
     - Measure how node attributes are distributed within hyperedge.
  2. **Node-level** measure (node homophily score)
     - Measure the tendency of nodes to form hyperedges with others of similar attributes.

Measure 1. Type-s Affinity Ratio Scores

- *Type-*(s, t) *affinity ratio score* [7] quantifies the significance of a particular attribute value appearing exactly $\boldsymbol{t}$ **times** in **size** $\boldsymbol{s}$ hyperedges.

- For attribute value v in dimension d, we compute type-(s, t) affinity ratio score as:

***Normalized* frequency** of (d, v) nodes in (s, t) hyperedges

$$= \frac{\textbf{\textit{Observed} frequency of } (d, v) \text{ nodes in } (s, t) \text{ hyperedges}}{\textbf{\textit{Expected} frequency of } (d, v) \text{ nodes in } (s, t) \text{ hyperedges}}$$

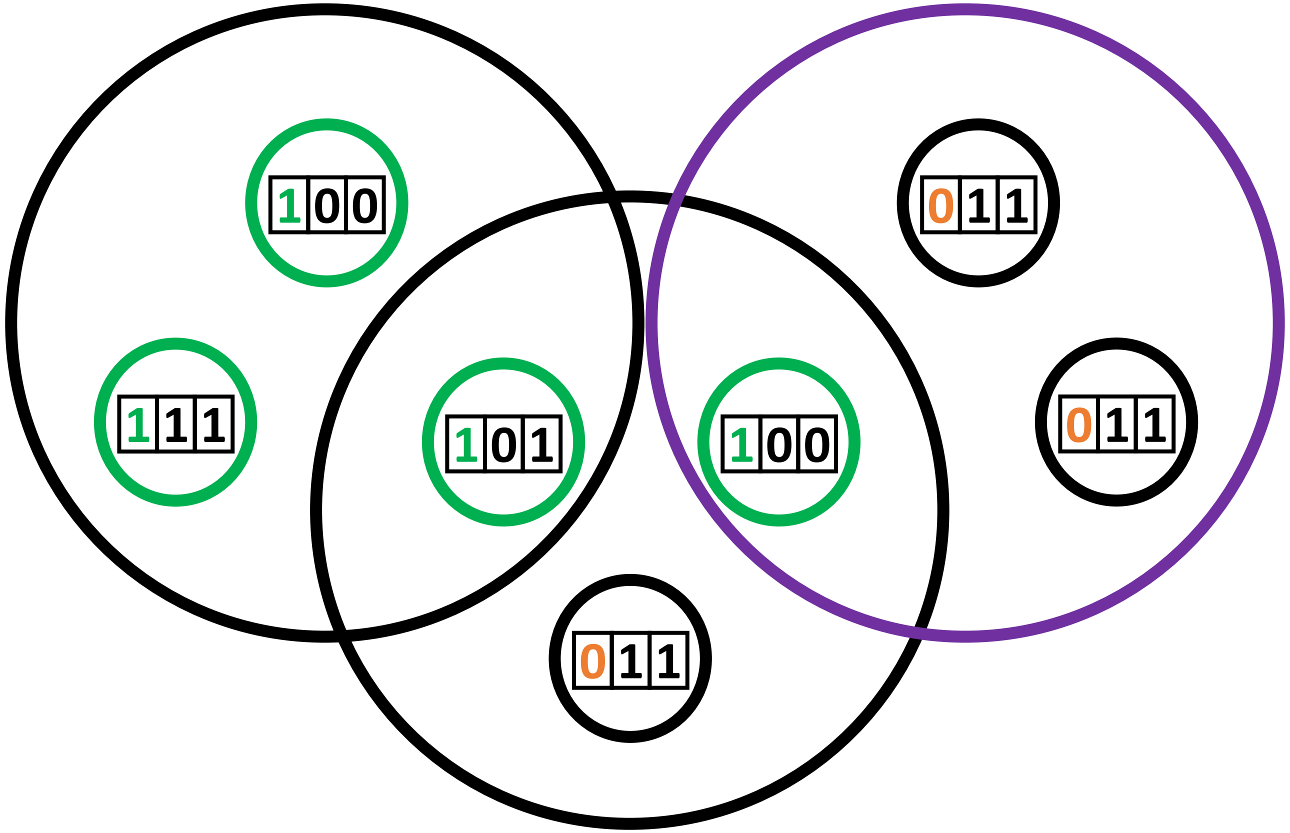
where

- (d, v) *nodes* are nodes with attribute value v in dimension d.
- (s, t) *hyperedges* are hyperedges of size s with t nodes of value v.

Measure 1. Type-s Affinity Ratio Scores (cont.)

- E.g. type-$(3, 1)$ affinity ratio score for value 1 in dimension 1

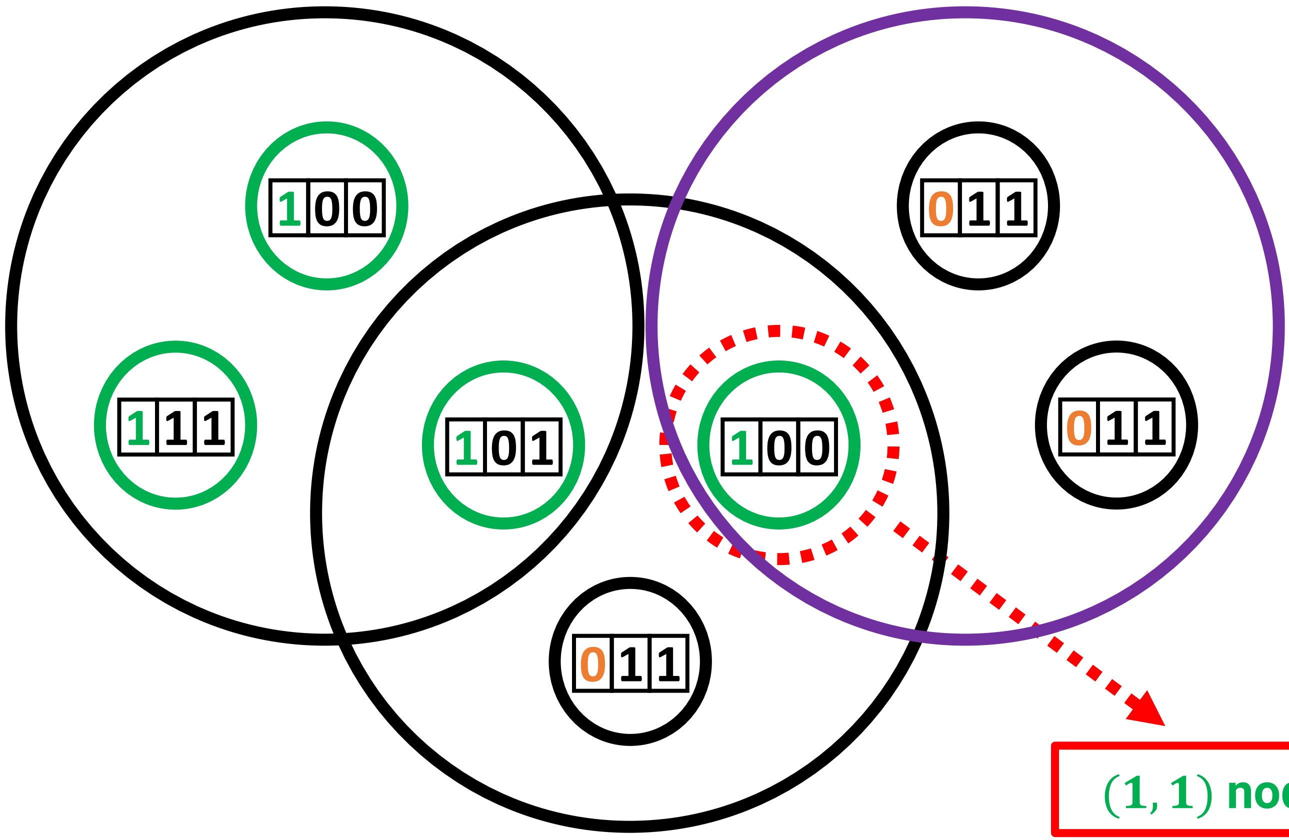
  $=$ normalized frequency of $(1, 1)$ nodes in $(3, 1)$ hyperedges



Measure 1. Type-s Affinity Ratio Scores (cont.)

- E.g. type-$(3, 1)$ affinity ratio score for value 1 in dimension 1

  $=$ normalized frequency of $(1, 1)$ nodes in $(3, 1)$ hyperedges



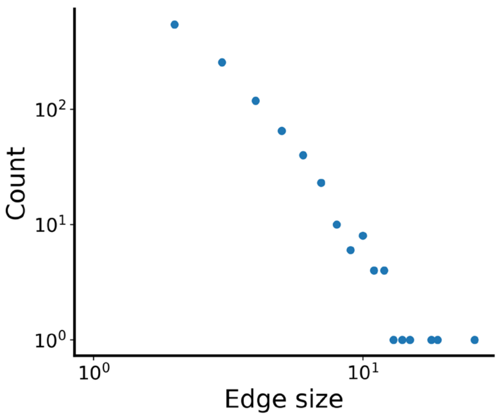
$(1, 1)$ **node in a** $(3, 1)$ **hyperedge!**

Measure 1. Type-s Affinity Ratio Scores (cont.)

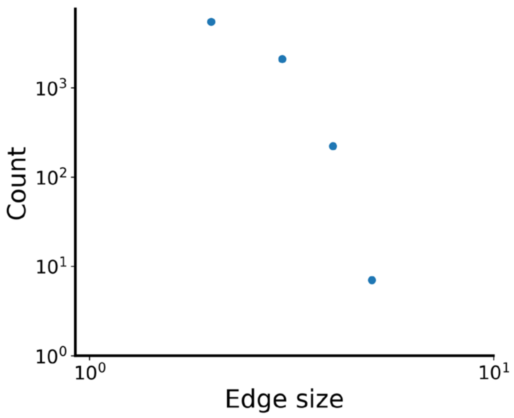
- E.g. type-(s, t) affinity ratio score for value 1 in any attribute dimension
  - **High** ratio score
    - If $t \approx s$ → Homophily w.r.t. value 1
    - If $t \ll s$ → Homophily w.r.t. value 0
  - **Low** ratio score
    - If $t \approx s$ → Anti-homophily w.r.t. value 1
    - If $t \ll s$ → Anti-homophily w.r.t. value 0

Measure 1. Type-s Affinity Ratio Scores (cont.)

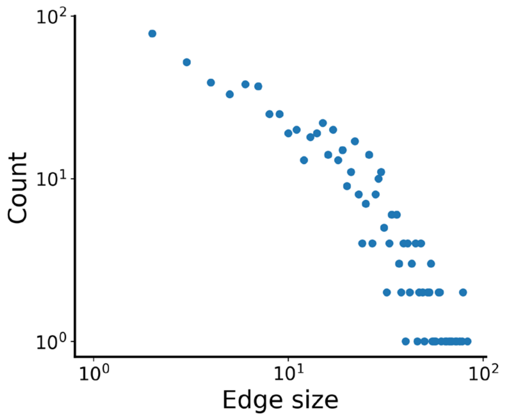
- We denote type-(s, t) affinity ratio scores from $t = 1$ to $t = s$ as *type-s affinity ratio scores*.
- We consider hyperedge size $s \in \{2,3,4\}$ (each denoted as **T2**, **T3**, and **T4**).
    - Most hyperedges in real world are small sized.



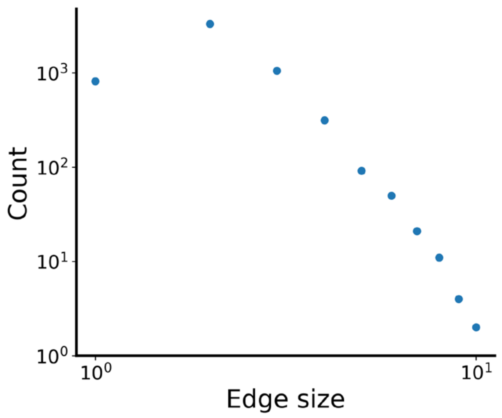
Citeseer
(academic paper)



High School
(contact)



Amazon Music
(review)

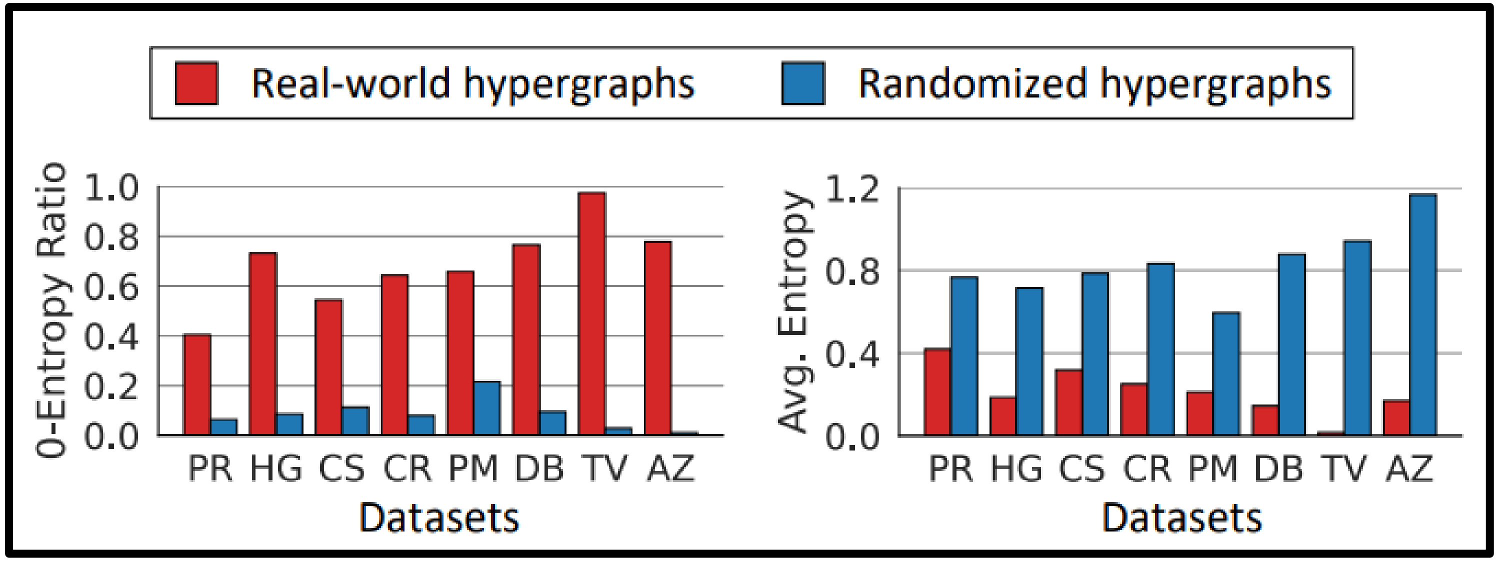


Devops
(online Q&A)

Measure 2. Hyperedge Entropy

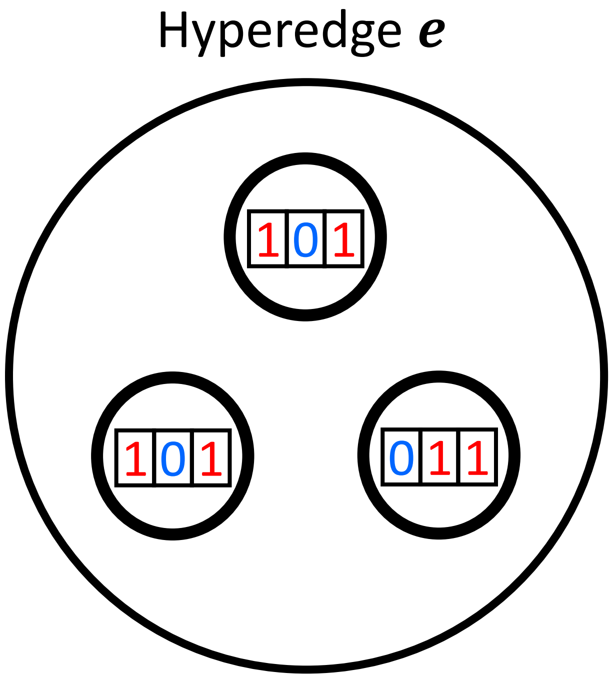
- *Hyperedge entropy* quantifies the hyperedge-level attribute homogeneity.

- Lee et al. [8] observed real-world hyperedges exhibit homogeneity for node labels.
  - Note) labels are also node attributes.



Measure 2. Hyperedge Entropy (cont.)

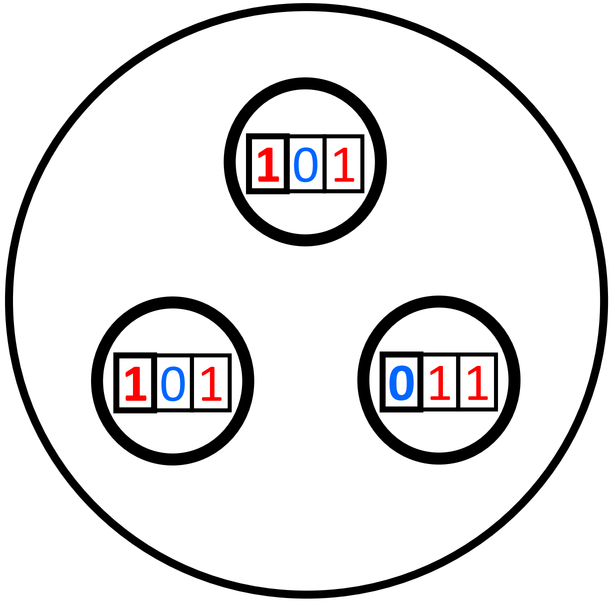
- Hyperedge entropy (**HE**) is defined for each attribute dimension.



Measure 2. Hyperedge Entropy (cont.)

- Hyperedge entropy (**HE**) is defined for each attribute dimension.

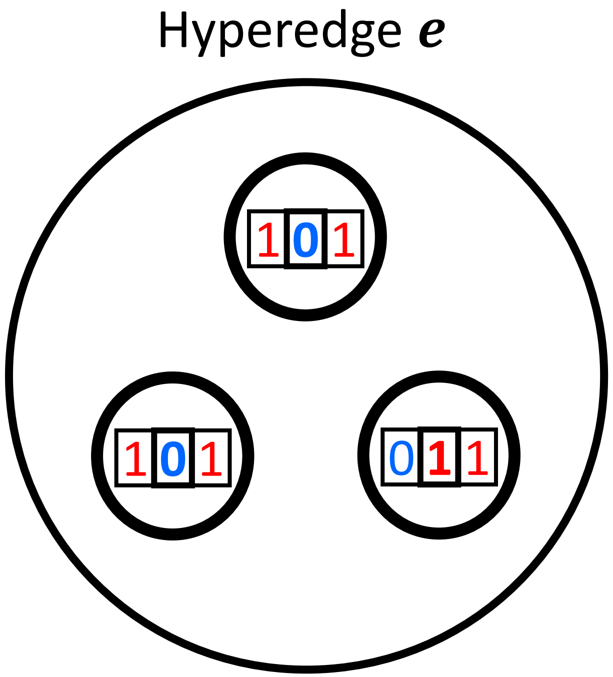
Hyperedge e



$$\mathbf{HE_e[1]} = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.6365$$

Measure 2. Hyperedge Entropy (cont.)

- Hyperedge entropy (**HE**) is defined for each attribute dimension.



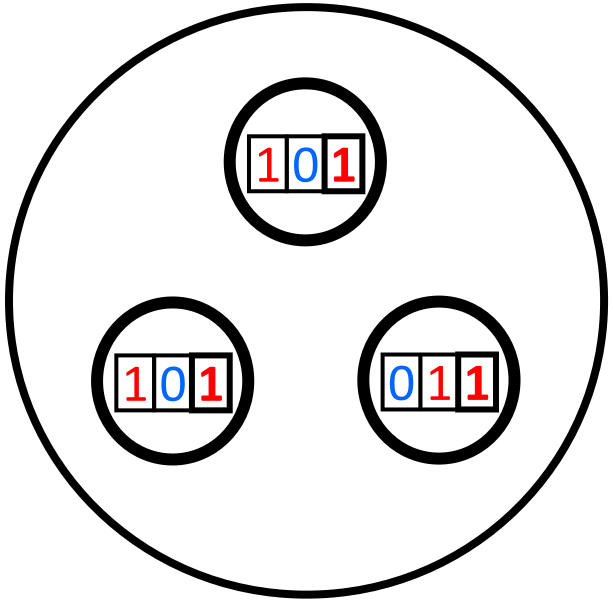
$$\mathrm{HE_e}[1] = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.6365$$

$$\mathrm{HE_e}[2] = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3} = 0.6365$$

Measure 2. Hyperedge Entropy (cont.)

- Hyperedge entropy (**HE**) is defined for each attribute dimension.

Hyperedge $\boldsymbol{e}$



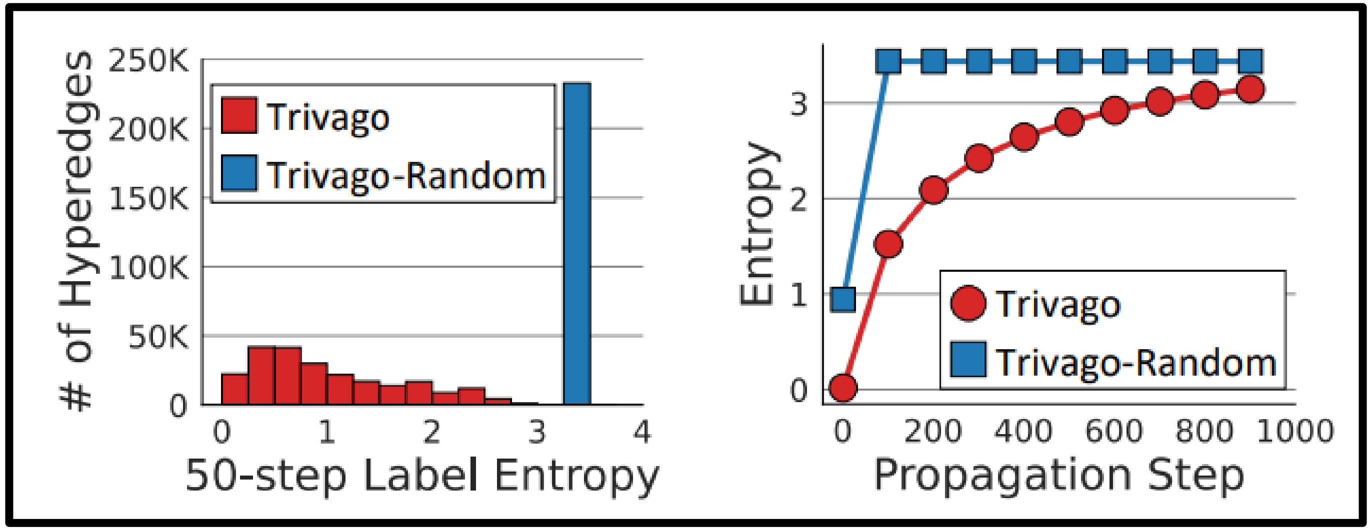
$$\mathbf{HE_e[1]} = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = \mathbf{0.6365}$$

$$\mathbf{HE_e[2]} = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3} = \mathbf{0.6365}$$

$$\mathbf{HE_e[3]} = -\frac{3}{3}\log\frac{3}{3} - \frac{0}{3}\log\frac{0}{3} = \mathbf{0.0000}$$

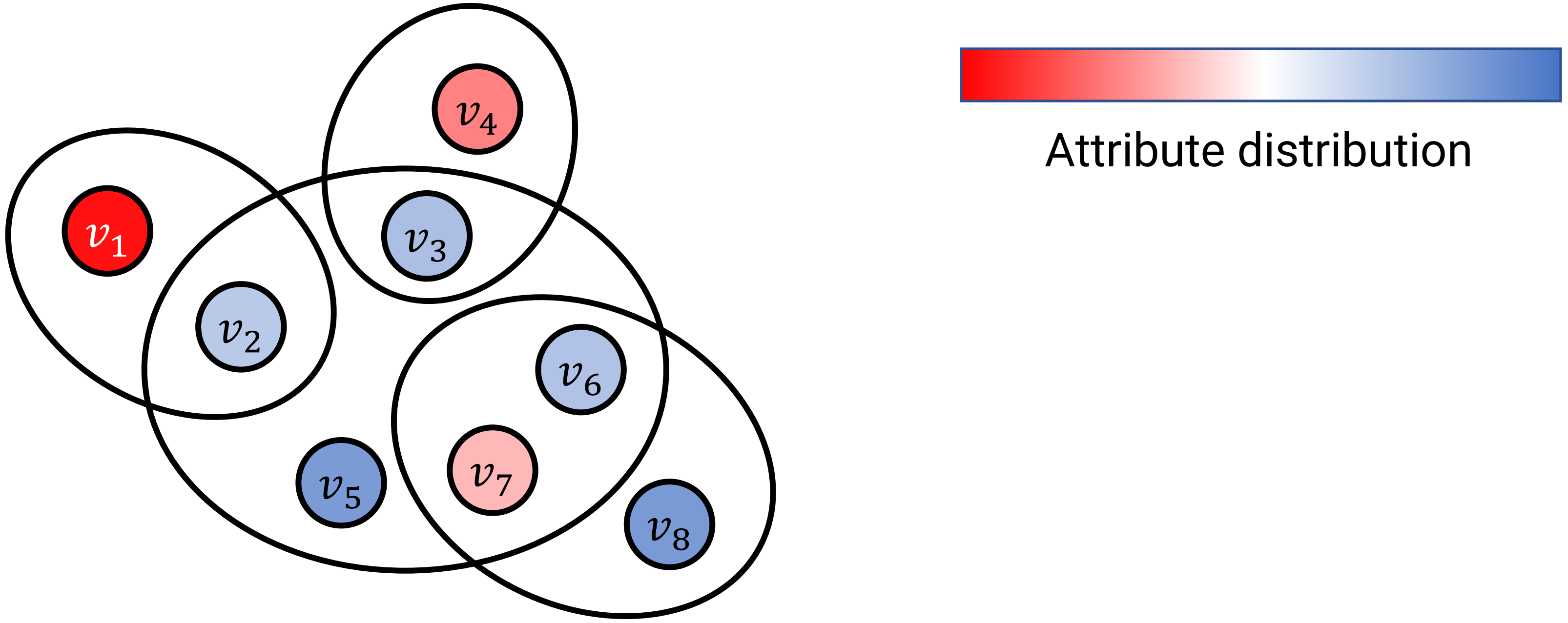
Measure 2. Hyperedge Entropy (cont.)

- Lee et al. [8] observed real-world hyperedges exhibit homogeneity for node labels, even after propagation step between incident nodes and hyperedges.



Measure 2. Hyperedge Entropy (cont.)

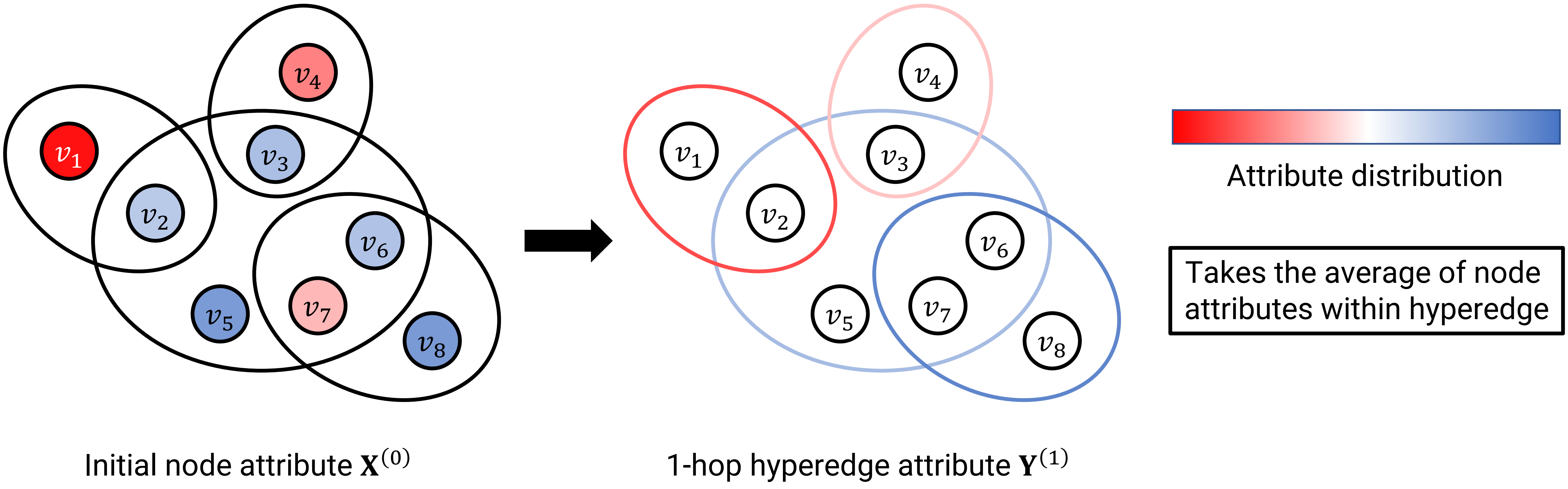
- We measure *higher-order hyperedge entropy* (**HOHE**), which is hyperedge entropy after multiple rounds of attribute propagation:



Initial node attribute $\mathbf{X}^{(0)}$

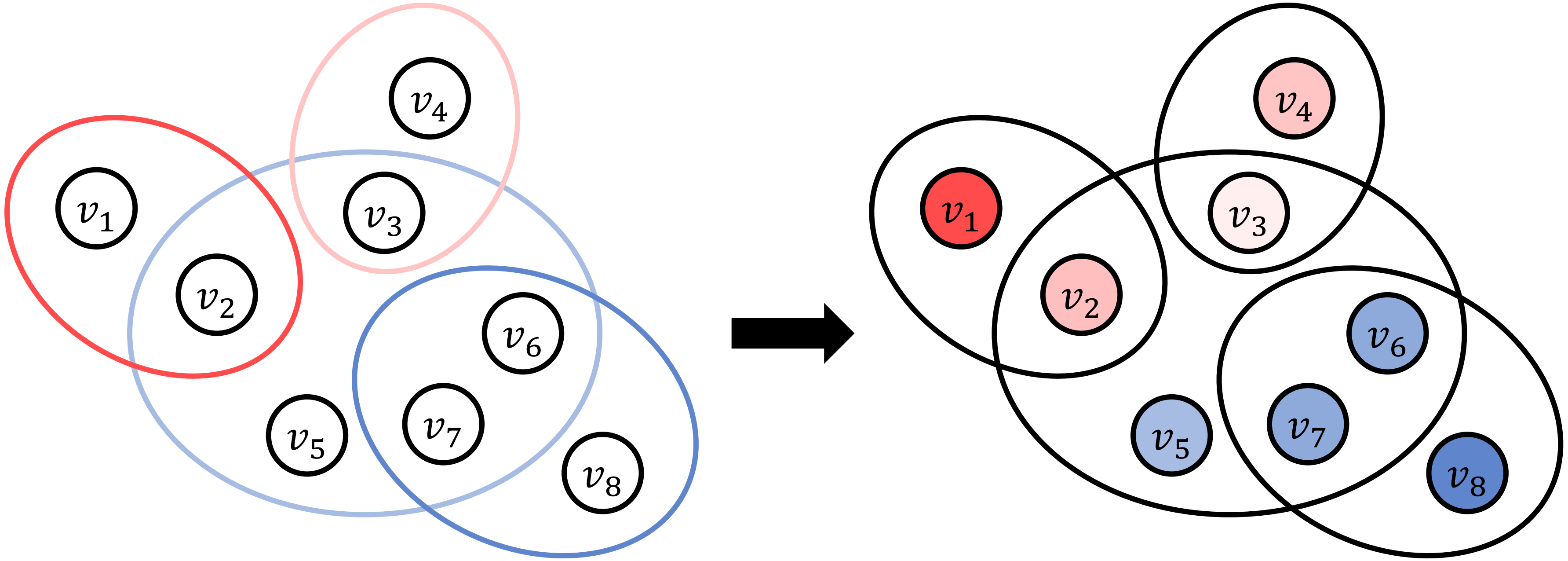
Measure 2. Hyperedge Entropy (cont.)

- We measure higher-order hyperedge entropy (**HOHE**), which is hyperedge entropy after multiple rounds of attribute propagation: **1) node → hyperedge**



Measure 2. Hyperedge Entropy (cont.)

- We measure higher-order hyperedge entropy (**HOHE**), which is hyperedge entropy after multiple rounds of attribute propagation: 1) node → hyperedge, **2) hyperedge → node**.



1-hop hyperedge attribute $\mathbf{Y}^{(1)}$

1-hop node attribute $\mathbf{X}^{(1)}$

Attribute distribution

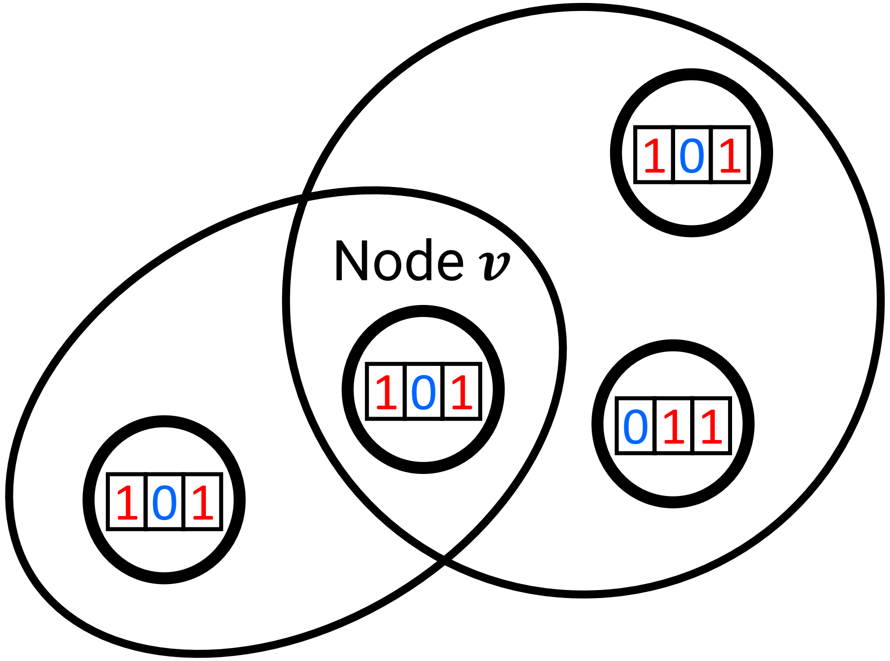
Takes the average of hyperedge attributes node is contained in

Measure 3. Node Homophily Score

- *Node homophily score* (**NHS**) quantifies the node-level attribute homogeneity.

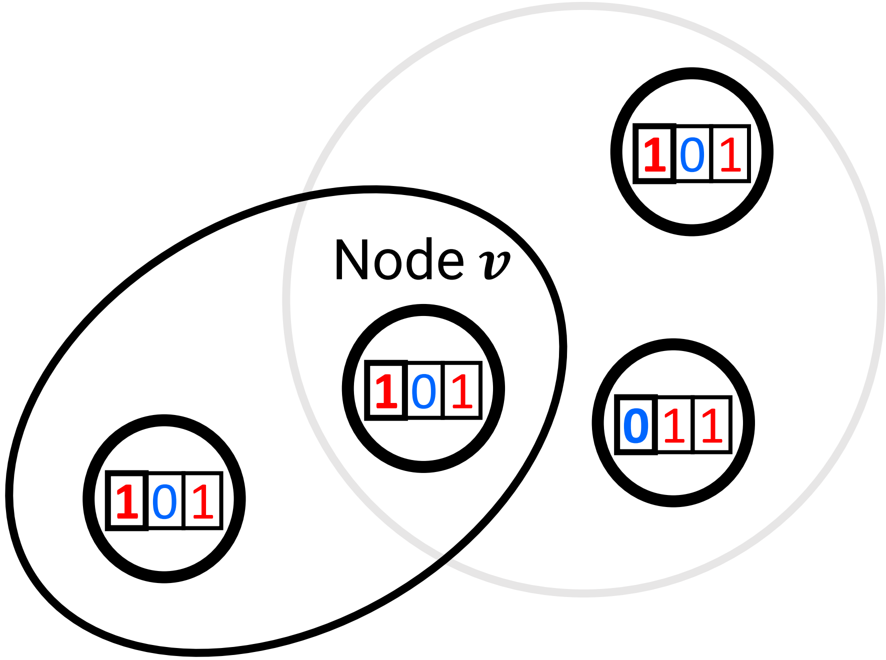
Measure 3. Node Homophily Score (cont.)

- *Node homophily score* (**NHS**) quantifies the node-level attribute homogeneity.
  - NHS is measured for each node (v in this example) and each attribute dimension.



Measure 3. Node Homophily Score (cont.)

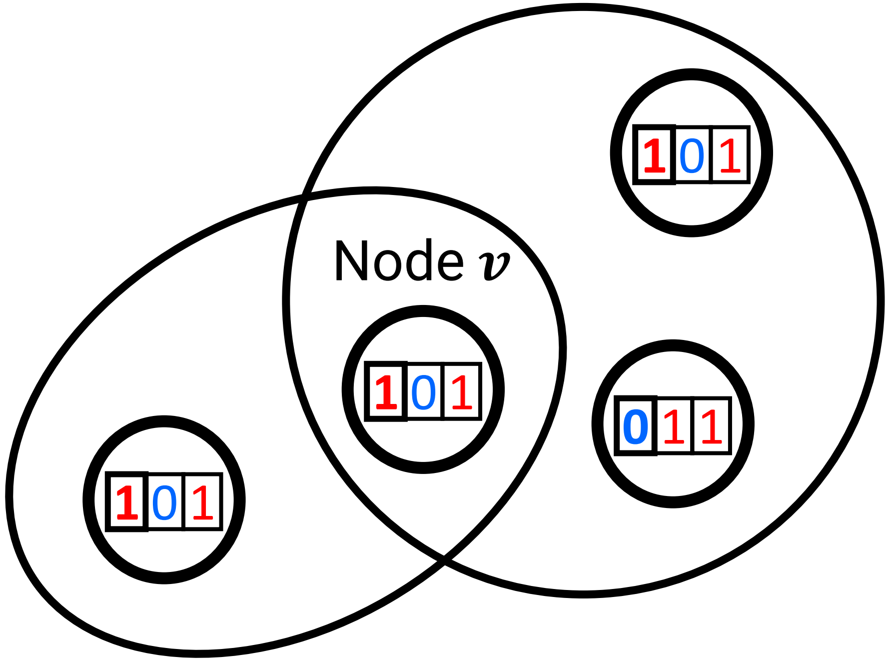
- *Node homophily score* (**NHS**) quantifies the node-level attribute homogeneity.
  - NHS is measured for each node (v in this example) and each attribute dimension.



$$\mathbf{NHS}_v[\mathbf{1}] = \frac{\mathbf{1}}{(\mathbf{2}-\mathbf{1})}$$

Measure 3. Node Homophily Score (cont.)

- *Node homophily score* (**NHS**) quantifies the node-level attribute homogeneity.
  - NHS is measured for each node (v in this example) and each attribute dimension.

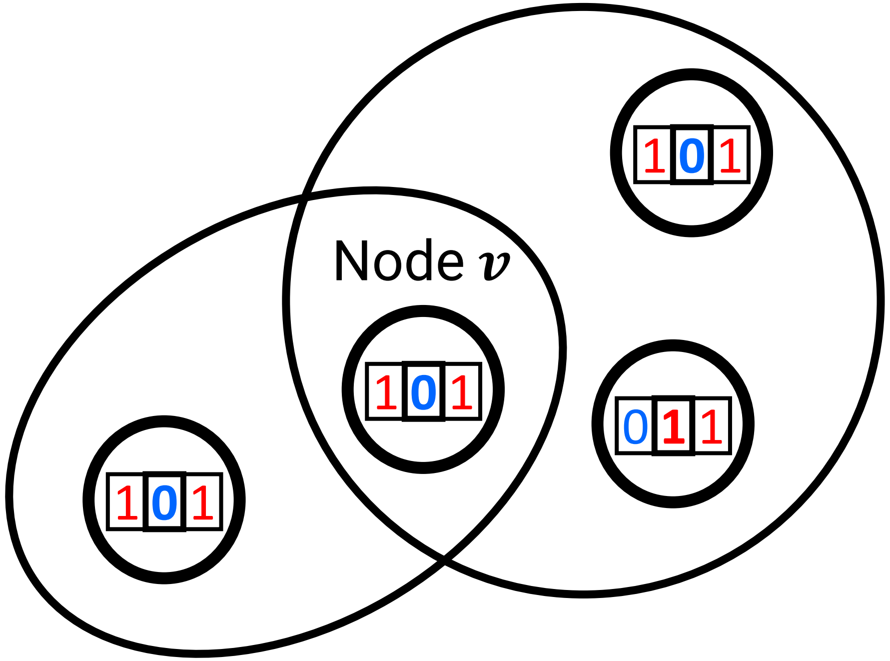


$$\mathbf{NHS}_{v}[\mathbf{1}] = \frac{\mathbf{1}+\mathbf{1}}{(\mathbf{2}-\mathbf{1})+(\mathbf{3}-\mathbf{1})} = \frac{\mathbf{2}}{\mathbf{3}}$$

Ratio of incident nodes with same attribute value

Measure 3. Node Homophily Score (cont.)

- *Node homophily score* (**NHS**) quantifies the node-level attribute homogeneity.
  - NHS is measured for each node ($\boldsymbol{v}$ in this example) and each attribute dimension.

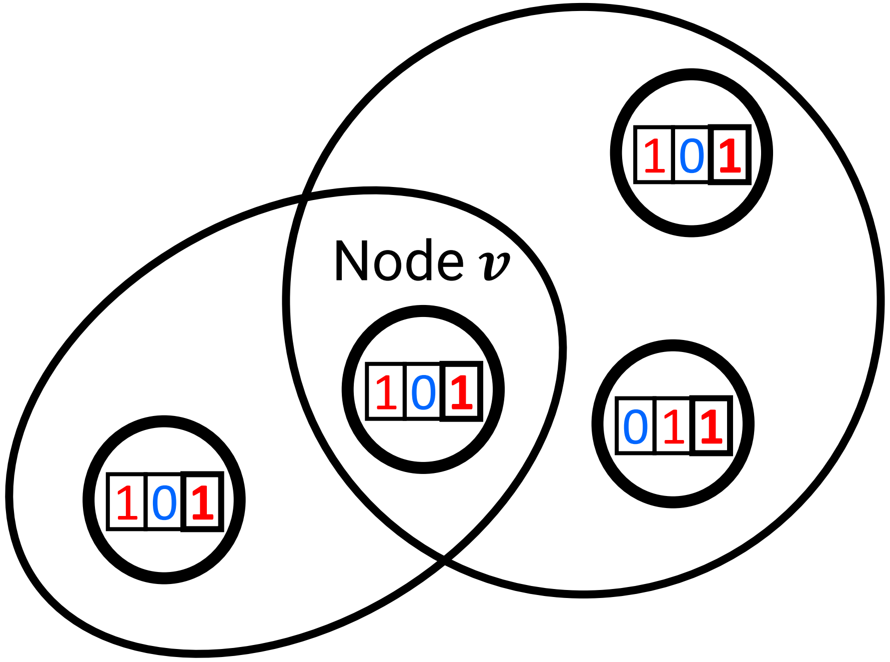


$$\mathbf{NHS}_{\boldsymbol{v}}[\mathbf{1}] = \frac{\mathbf{1}+\mathbf{1}}{(\mathbf{2}-\mathbf{1})+(\mathbf{3}-\mathbf{1})} = \frac{\mathbf{2}}{\mathbf{3}}$$

$$\mathbf{NHS}_{\boldsymbol{v}}[\mathbf{2}] = \frac{\mathbf{1}+\mathbf{2}}{(\mathbf{2}-\mathbf{1})+(\mathbf{3}-\mathbf{1})} = \frac{\mathbf{3}}{\mathbf{3}}$$

Measure 3. Node Homophily Score (cont.)

- *Node homophily score* (**NHS**) quantifies the node-level attribute homogeneity.
  - NHS is measured for each node ($\boldsymbol{v}$ in this example) and each attribute dimension.



$$\mathbf{NHS}_v[\mathbf{1}] = \frac{\mathbf{1}+\mathbf{1}}{(\mathbf{2}-\mathbf{1})+(\mathbf{3}-\mathbf{1})} = \frac{\mathbf{2}}{\mathbf{3}}$$

$$\mathbf{NHS}_v[\mathbf{2}] = \frac{\mathbf{1}+\mathbf{2}}{(\mathbf{2}-\mathbf{1})+(\mathbf{3}-\mathbf{1})} = \frac{\mathbf{3}}{\mathbf{3}}$$

$$\mathbf{NHS}_v[\mathbf{3}] = \frac{\mathbf{1}+\mathbf{2}}{(\mathbf{2}-\mathbf{1})+(\mathbf{3}-\mathbf{1})} = \frac{\mathbf{3}}{\mathbf{3}}$$

Roadmap

- Overview
- Preliminaries
- Proposed Method: NoAH, NoAHFit <<
- Experiments
- Conclusion

NoAH: Node Attribute based Hypergraph Generator

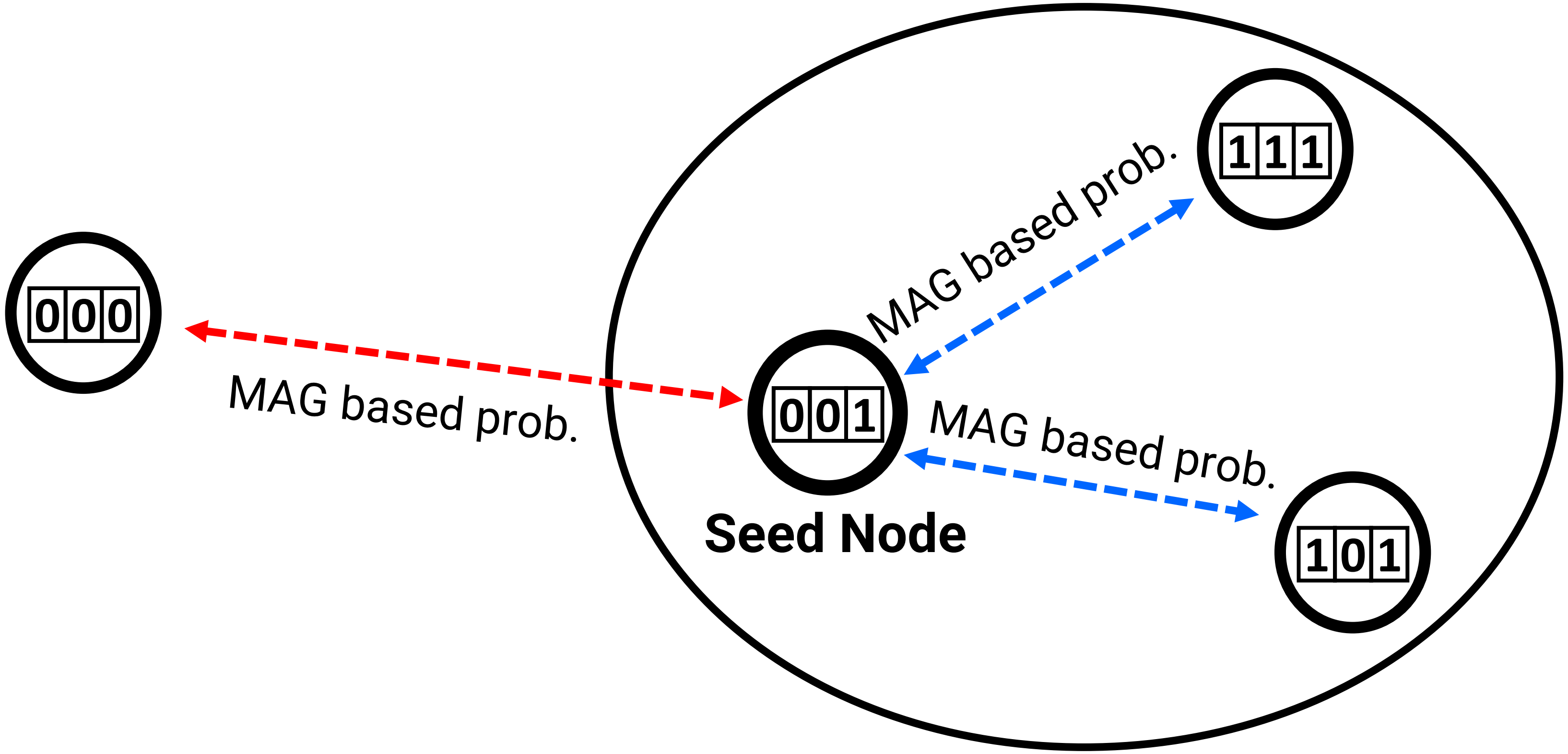
- In this work, we propose **NoAH**: **No**de **A**ttribute based **H**ypergraph Generator.

- NoAH aims to generate realistic hypergraphs that capture the **interplay between structure and attribute** observed in real-world hypergraphs.

- NoAH is built on three key ideas.

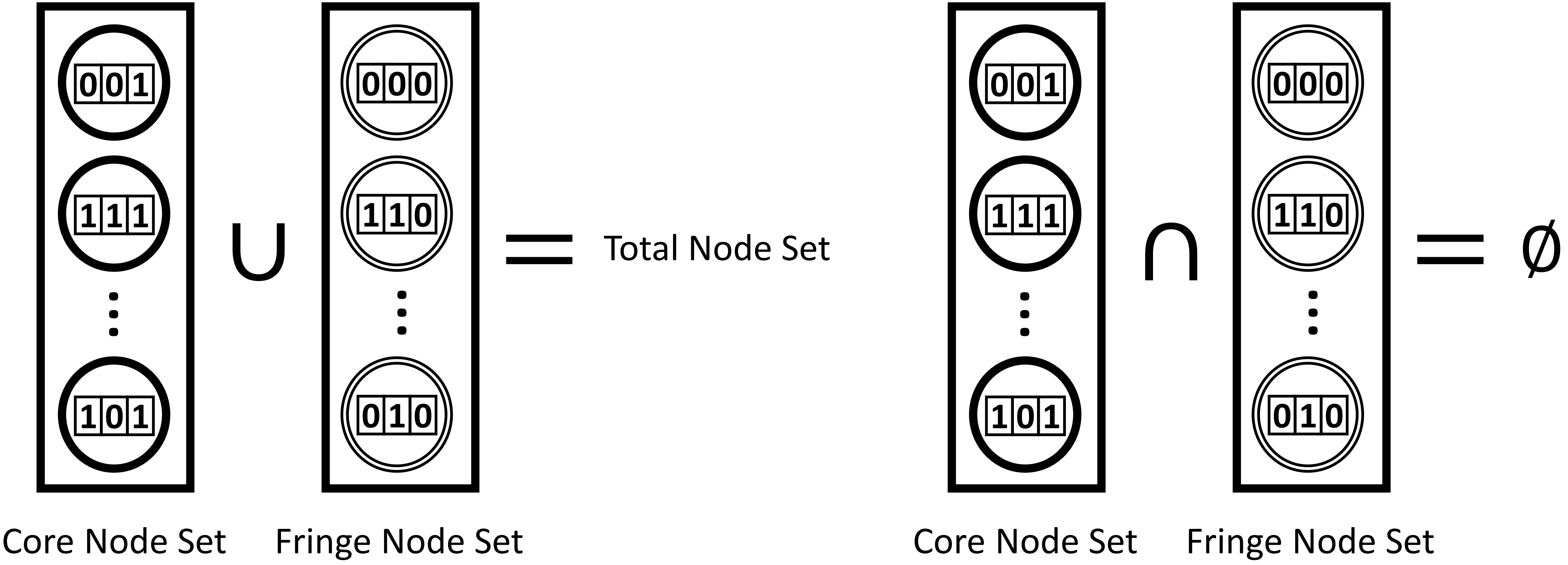
Ideas behind NoAH

- **Idea 1.** We model hyperedge formation based on node attributes.
  - NoAH takes into account the interplay between structure and attributes.



Ideas behind NoAH (cont.)

- **Idea 2.** We divide nodes into core and fringe nodes.
  - NoAH captures hierarchical structure prevalent in real-world hypergraphs [4, 5].



Ideas behind NoAH (cont.)

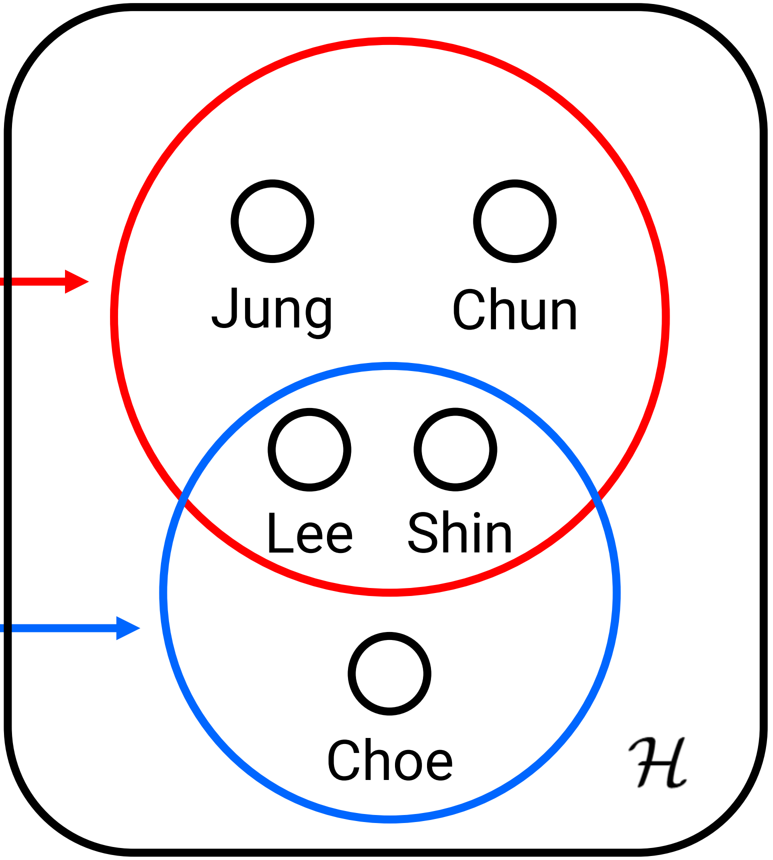
- **Idea 2.** We divide nodes into core and fringe nodes.
  - NoAH captures hierarchical structure prevalent in real-world hypergraphs [4, 5].

Random walk with restart on hypergraphs: fast computation and an application to anomaly detection

Jaewan Chun[1] · Geon Lee[1] · Kijung Shin[1] · Jinhong Jung[2]

How Do Hyperedges Overlap in Real-World Hypergraphs? - Patterns, Measures, and Generators

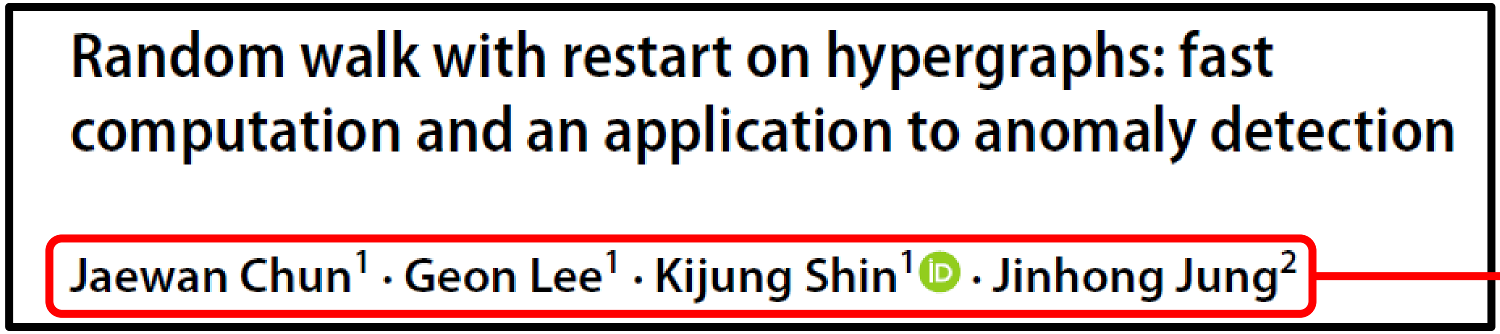
Geon Lee*	Minyoung Choe*	Kijung Shin
KAIST AI	KAIST AI	KAIST AI & EE
Daejeon, South Korea	Daejeon, South Korea	Daejeon, South Korea
geonlee0325@kaist.ac.kr	minyoung.choe@kaist.ac.kr	kijungs@kaist.ac.kr



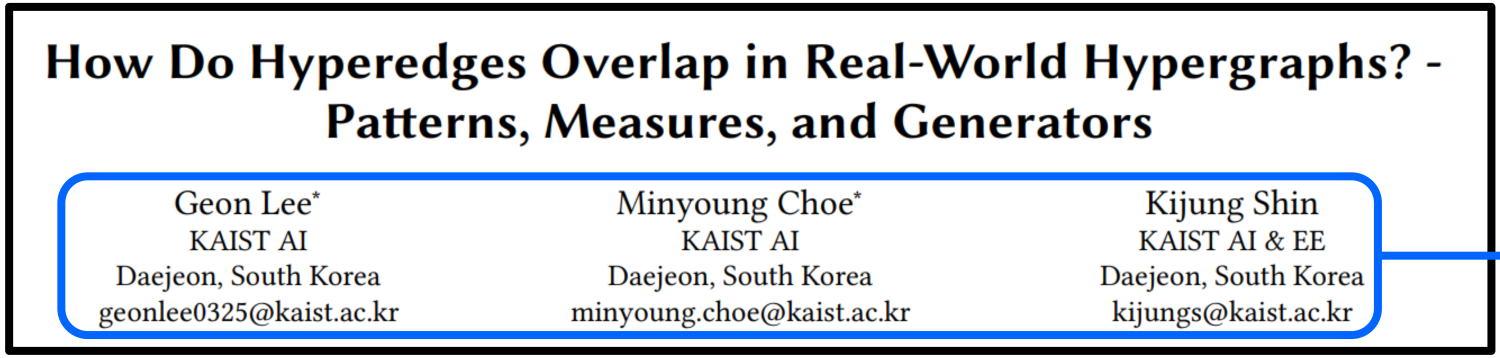
Ideas behind NoAH (cont.)

Corresponding author **(core node)** playing a central role in establishing collaborations **(hyperedge formation)**.

- **Idea 2.** We divide nodes into core and fringe nodes.
  - NoAH captures hierarchical structure prevalent in real-world hypergraphs [4, 5].


Random walk with restart on hypergraphs: fast computation and an application to anomaly detection

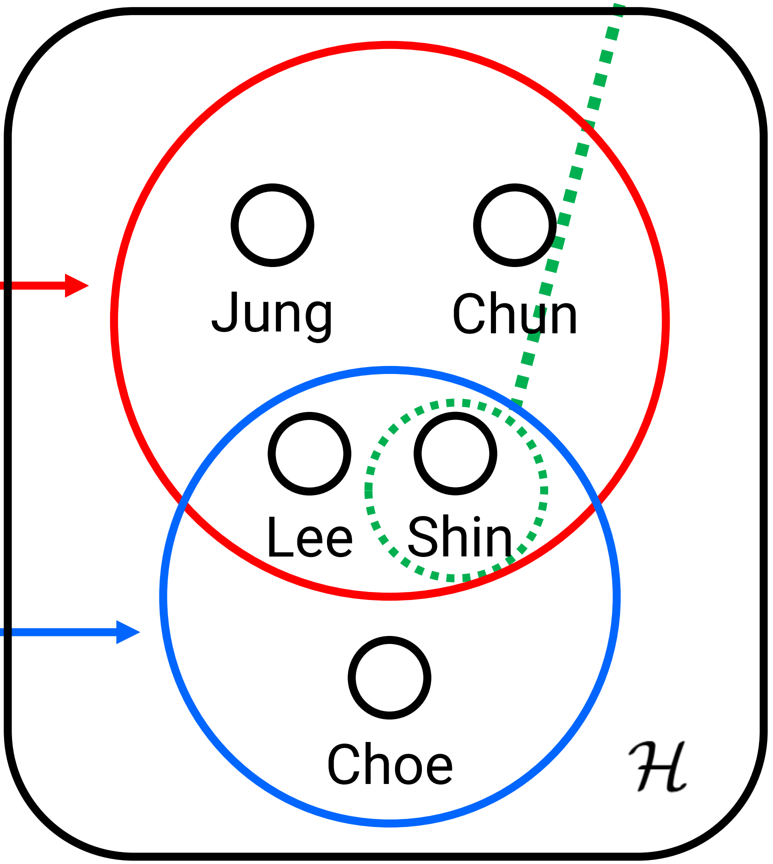
Jaewan Chun[1] · Geon Lee[1] · Kijung Shin[1] · Jinhong Jung[2]


How Do Hyperedges Overlap in Real-World Hypergraphs? - Patterns, Measures, and Generators

Geon Lee*
KAIST AI
Daejeon, South Korea
geonlee0325@kaist.ac.kr

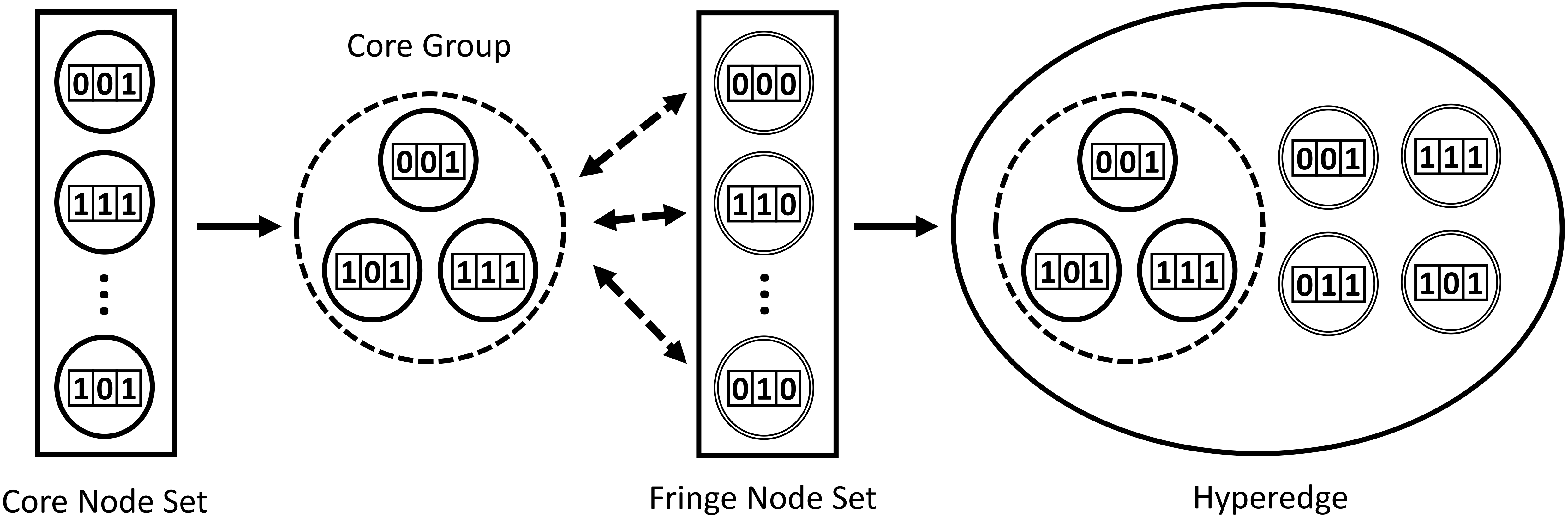
Minyoung Choe*
KAIST AI
Daejeon, South Korea
minyoung.choe@kaist.ac.kr

Kijung Shin
KAIST AI & EE
Daejeon, South Korea
kijungs@kaist.ac.kr



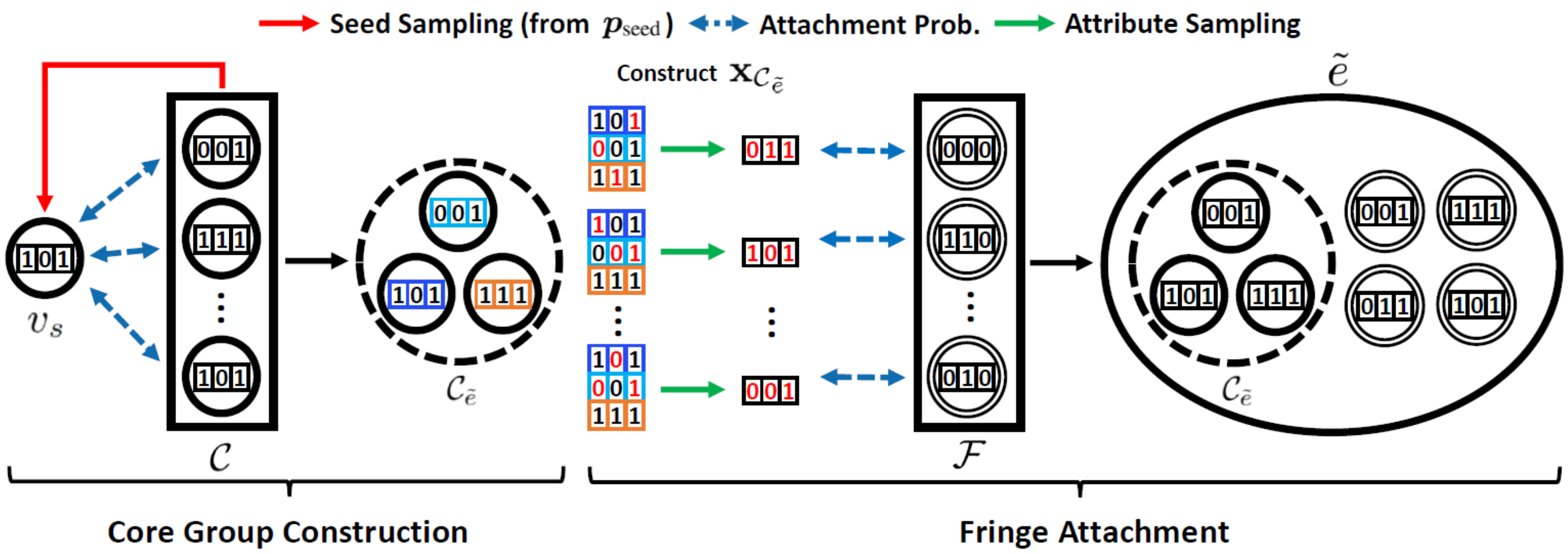
Ideas behind NoAH (cont.)

- **Idea 3.** We model the formation of each hyperedge as a series of attachments.
  - NoAH reduces computational complexity via incremental construction.



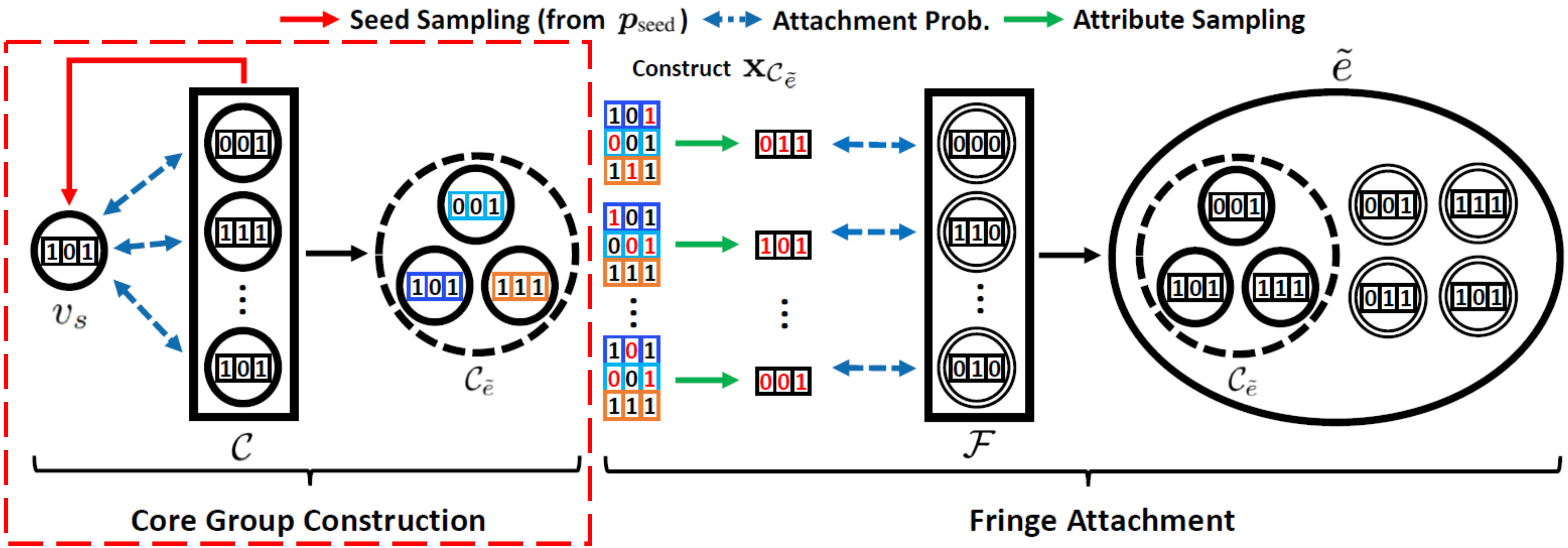
Details of NoAH

- NoAH generates each hyperedge through a two-step process:
    1. Core group construction
    2. Fringe attachment



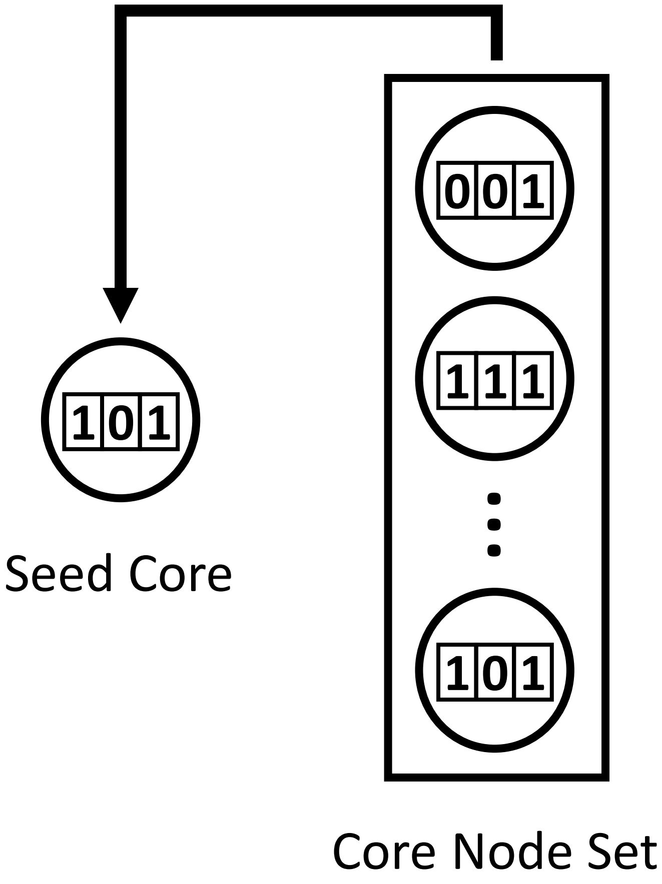
Details of NoAH: 1. Core Group Construction

- NoAH generates each hyperedge through a two-step process:
  1. **Core group construction**
  2. Fringe attachment



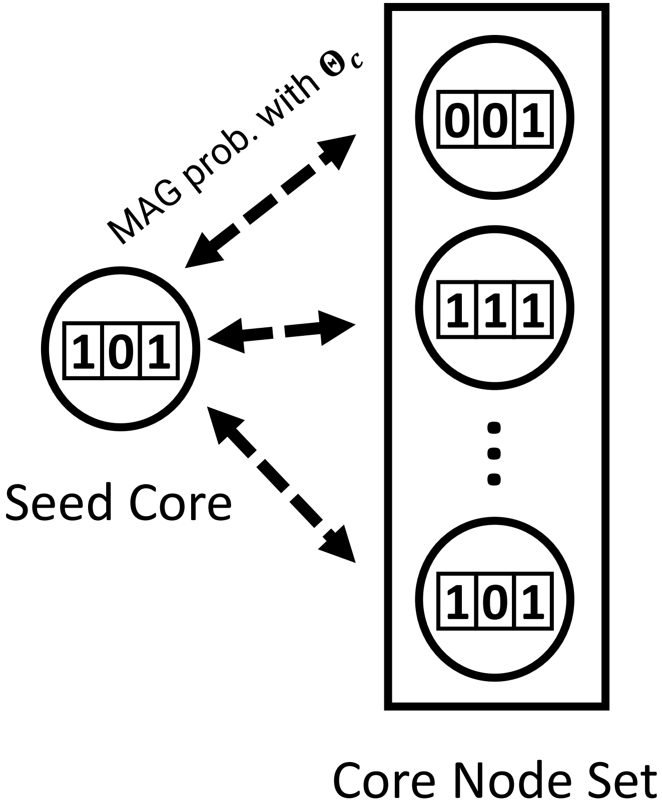
Details of NoAH: 1. Core Group Construction (cont.)

- 1) Select a seed core from $\boldsymbol{p}_{\text{seed}}$.



Details of NoAH: 1. Core Group Construction (cont.)

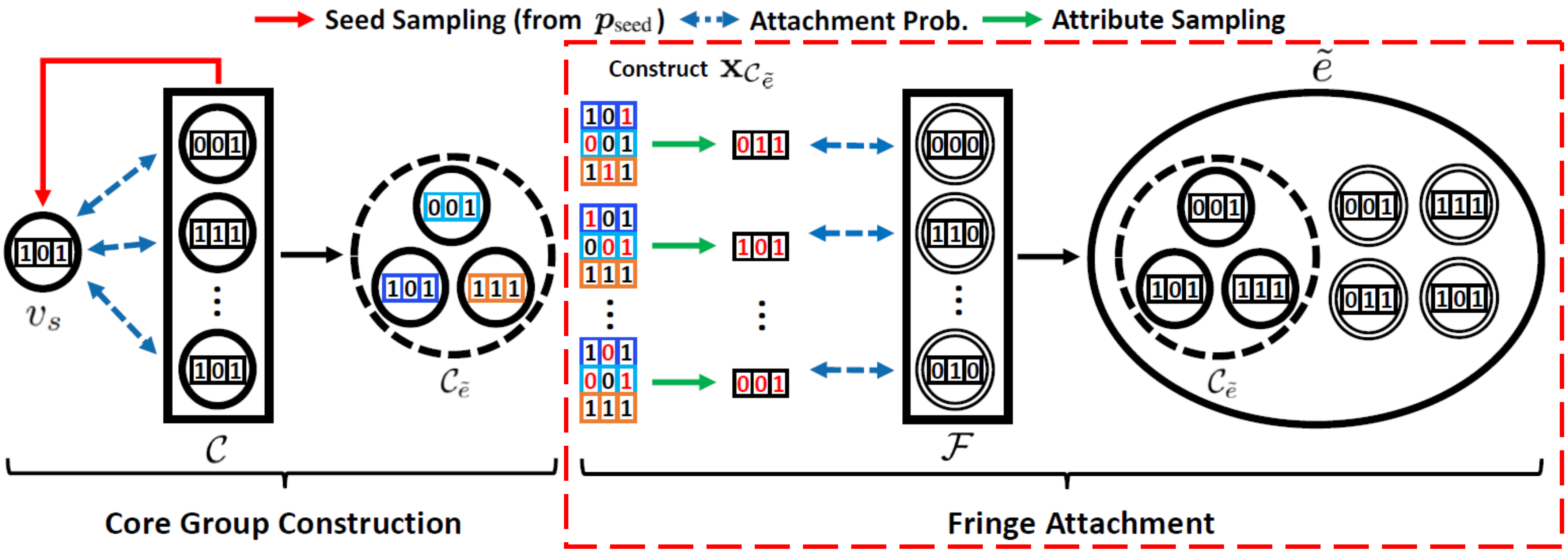
- 2) Compute attaching probability between seed core and other core nodes.



- Θ_C: core affinity matrices
- Θ_F: fringe affinity matrices

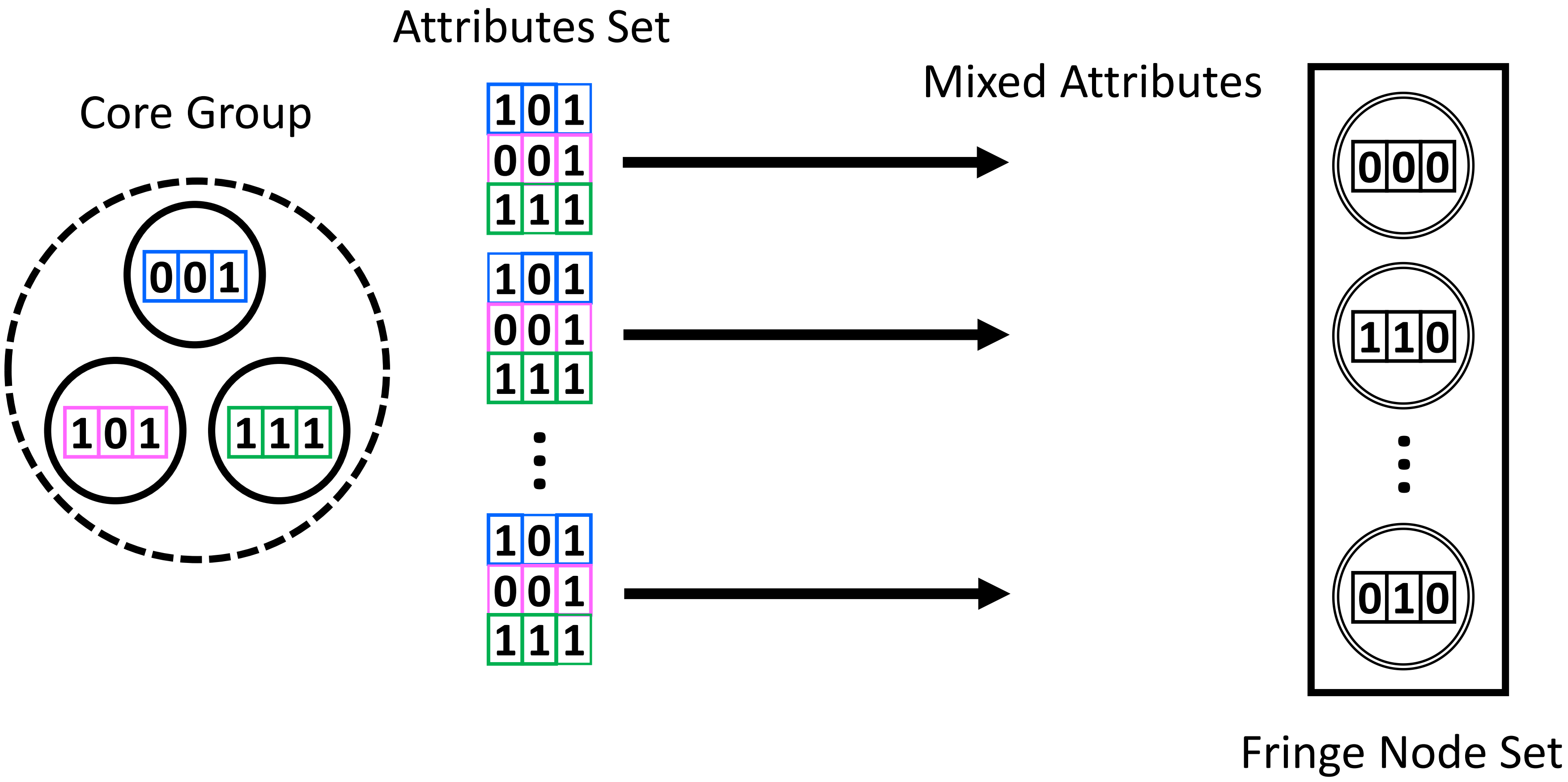
Details of NoAH: 2. Fringe Attachment

- NoAH generates each hyperedge through a two-step process:
    1. Core group construction
    2. **Fringe attachment**



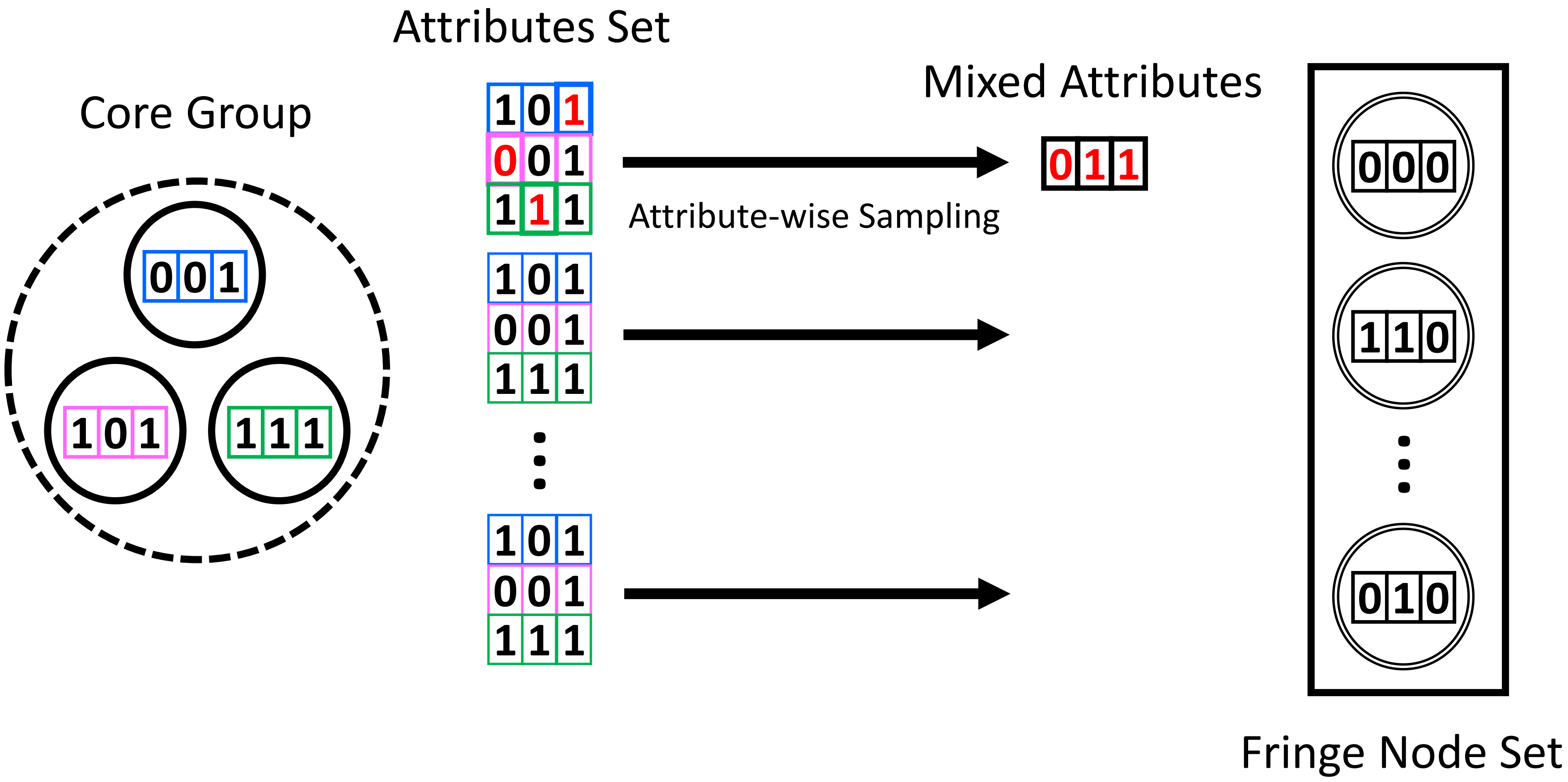
Details of NoAH: 2. Fringe Attachment

- 1) Mix the node attributes via attribute-wise sampling for each fringe node.



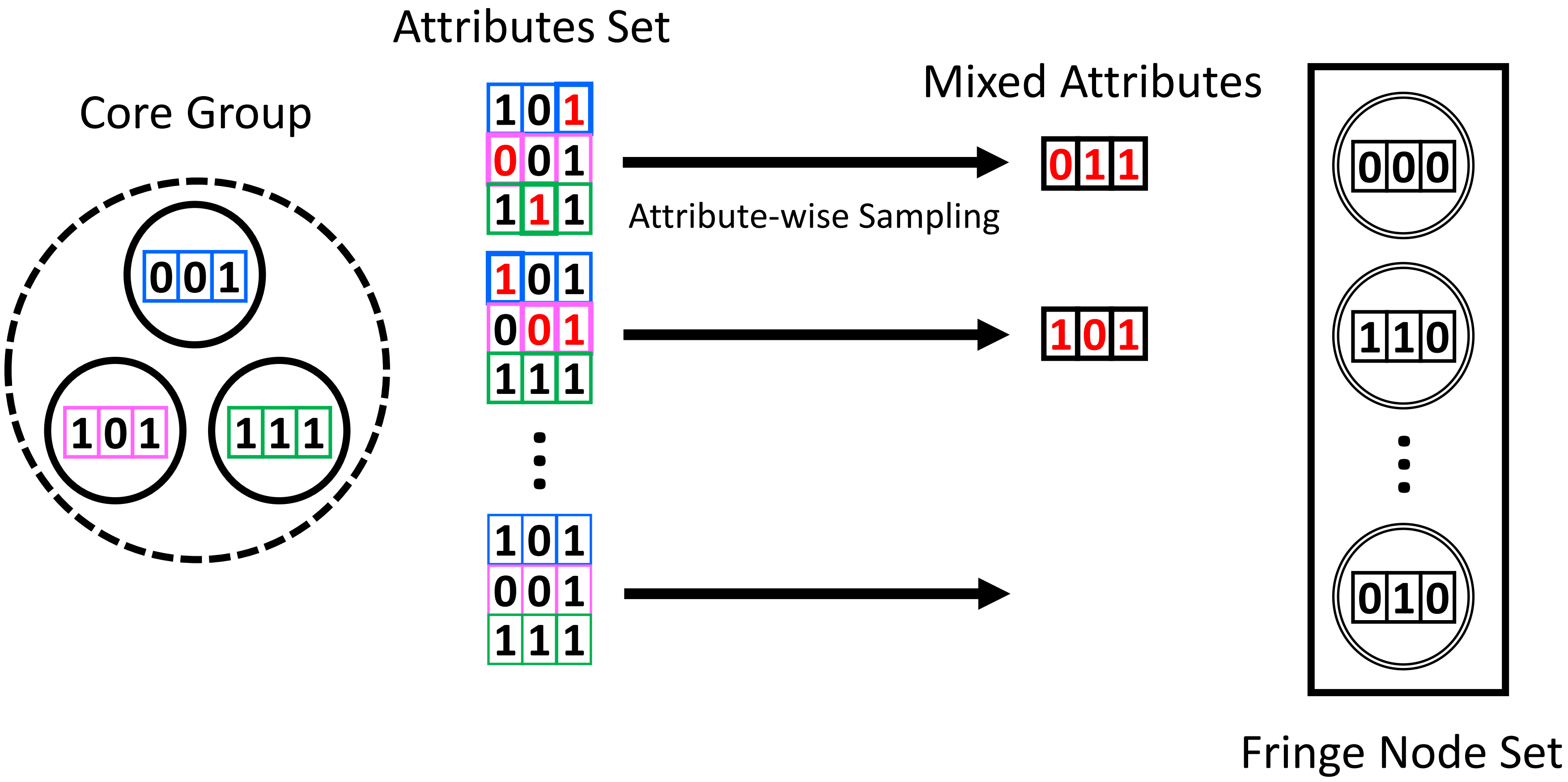
Details of NoAH: 2. Fringe Attachment

- 1) Mix the node attributes via attribute-wise sampling for each fringe node.



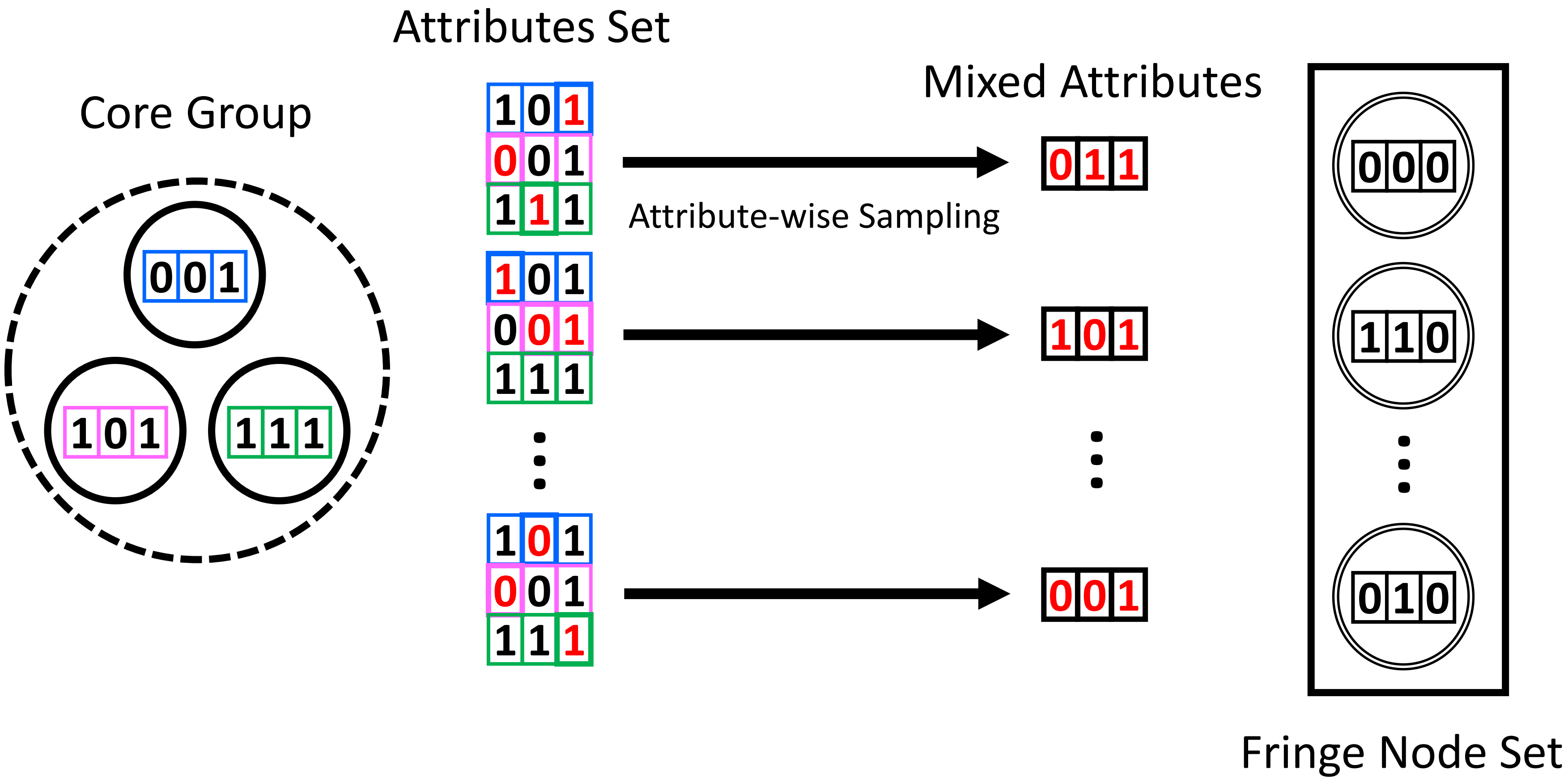
Details of NoAH: 2. Fringe Attachment

- 1) Mix the node attributes via attribute-wise sampling for each fringe node.



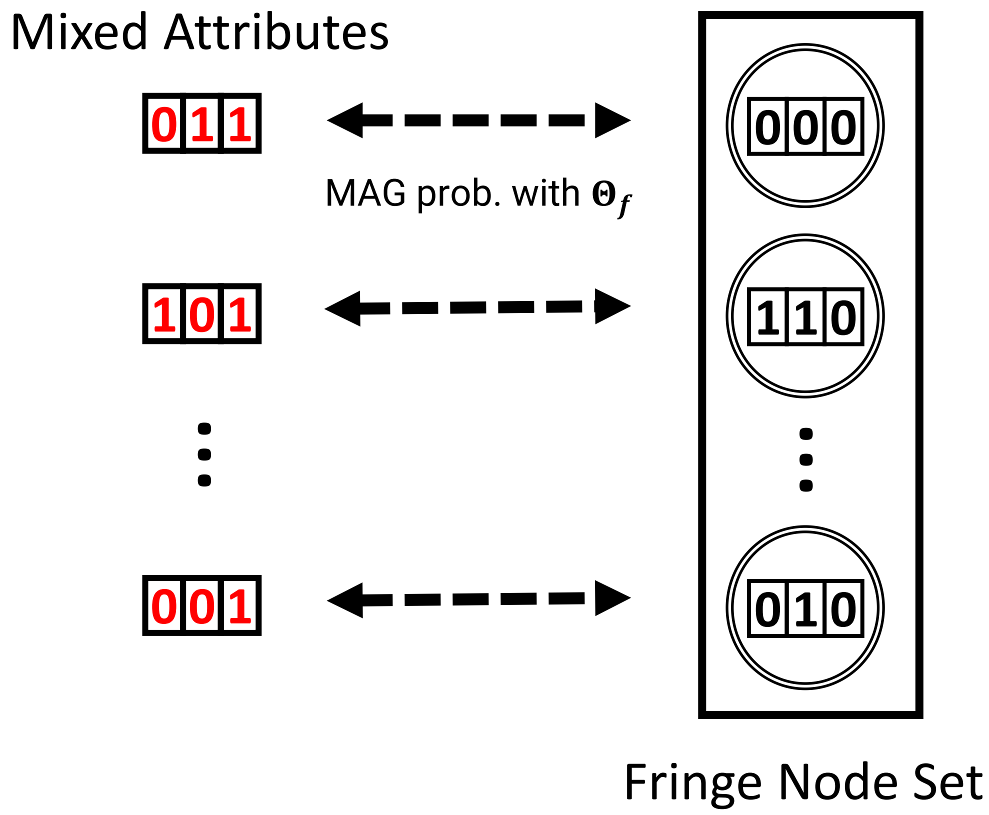
Details of NoAH: 2. Fringe Attachment

- 1) Mix the node attributes via attribute-wise sampling for each fringe node.



Details of NoAH: 2. Fringe Attachment (cont.)

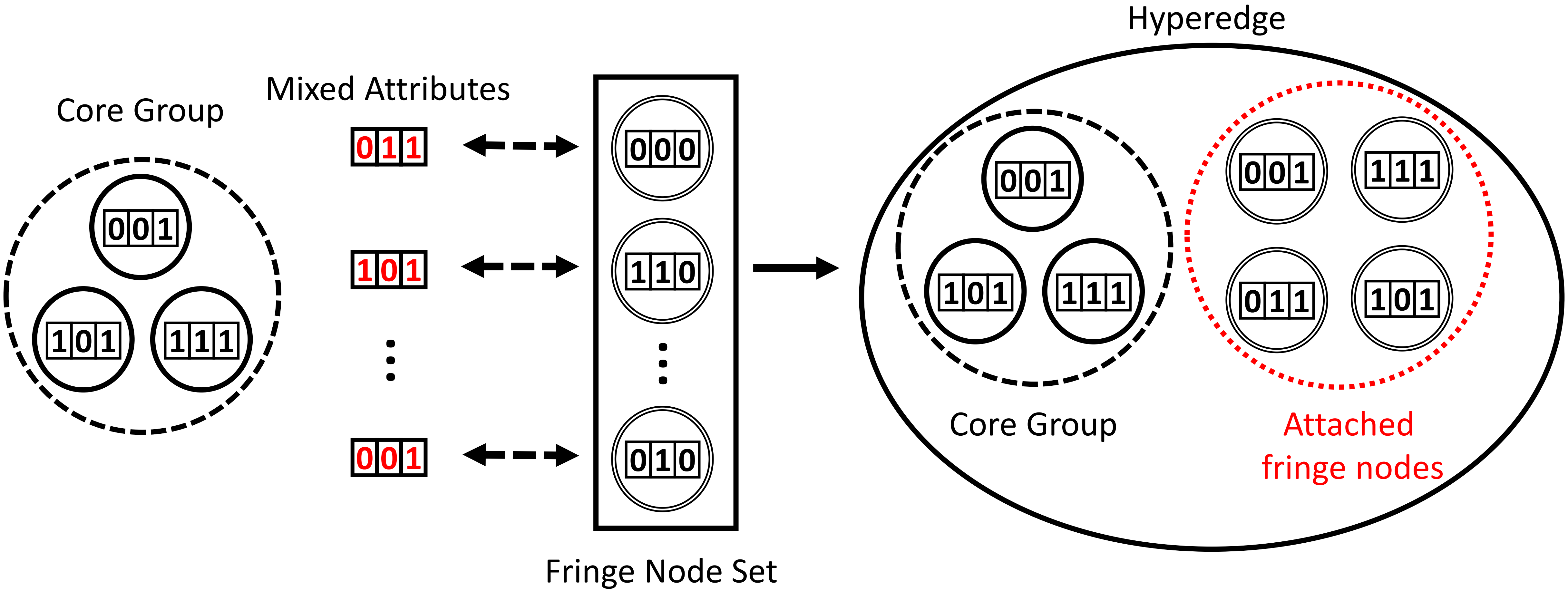
- 2) Compute attaching probability between core group and fringe nodes.



- Θ_C: core affinity matrices
- Θ_F: fringe affinity matrices

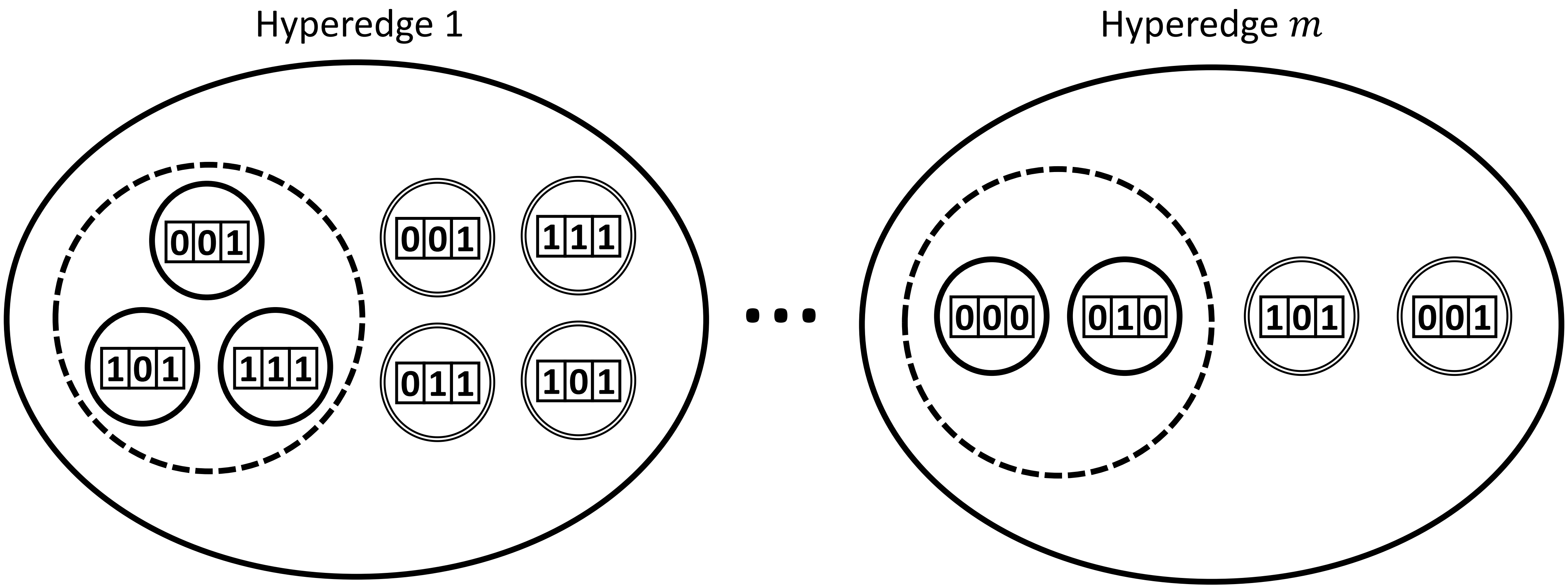
Details of NoAH: 2. Fringe Attachment (cont.)

- 3) Construct a hyperedge.



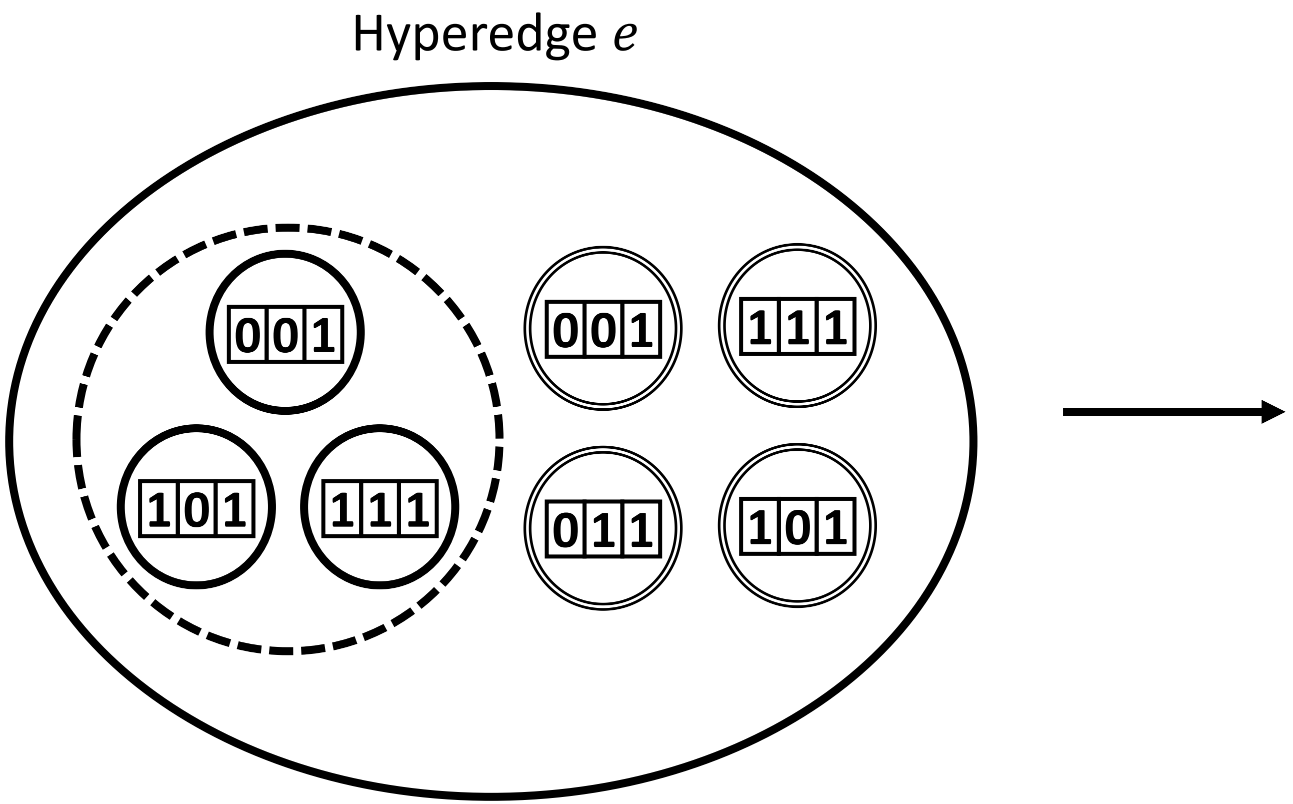
Details of NoAH (cont.)

- Iterate m times to generate a hypergraph.



NoAHFit

- The probability of each hyperedge being formed is parameterized through NoAH.



$$P(e) = f(\boldsymbol{p}_{\mathrm{seed}}, \Theta_C, \Theta_F)$$

- p_{seed}: seed core probability
- Θ_C: core affinity matrices
- Θ_F: fringe affinity matrices

NoAHFit (cont.)

- The probability of each hyperedge being formed is parameterized through NoAH.

  → We propose **NoAHFit**, a method for tuning the parameters of NoAH.

- NoAHFit enables NoAH to generate hypergraphs that closely resemble the input hypergraph in both 1) structure and 2) attribute patterns.
  - Useful for downstream applications including community detection [11] and hyperedge prediction [19].

NoAHFit (cont.)

- NoAHFit updates each parameter to maximize the likelihood of a given hypergraph,
  - i.e., the product of the likelihood of each hyperedge, directly modeled by NoAH.
  - Each parameter is updated using gradient descent.

$$L = -\log P(\mathcal{H} | p_{seed}, \Theta_C, \Theta_F)$$
$$= \sum_e -\log P(e | p_{seed}, \Theta_C, \Theta_F)$$

$$\rightarrow \nabla_{\boldsymbol{p}_{\text{seed}}} L$$
$$\rightarrow \nabla_{\Theta_C} L$$
$$\rightarrow \nabla_{\Theta_F} L$$

Theoretical Analysis of NoAH and NoAHFit

- Additionally, we show theoretical properties of NoAH and NoAHFit, including:
  - Capability of generating hypergraph with power-law degree distribution.
  - Time and space complexity analysis.

- For details, please refer to the paper.

Roadmap

- Overview
- Preliminaries
- Proposed Method: NoAH, NoAHFit
- Experiments <<
- Conclusion

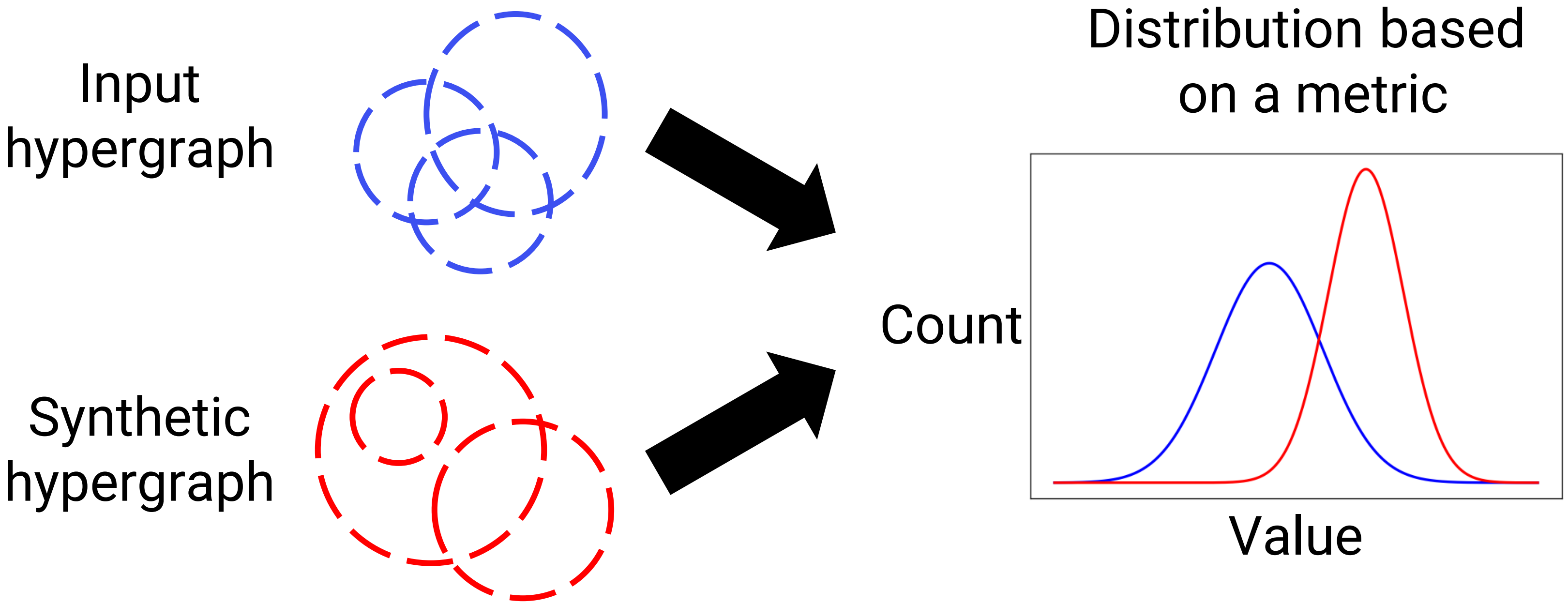
Experimental Settings

- **8 baselines**
  - 1 attribute-based model: HyCoSBM [11]
  - 3 node-identity preserved models: HyperCL, HyperLAP [1], hyper dK-series [10]
  - 4 randomized models: HyperPA [3], HyperFF [9], THera [2], HyRec [12]
- **1 variant:** NoAH-CF: no core-fringe node hierarchy

- **9 datasets:**
  - Citeseer, Cora [13] (academic paper domain)
  - High School [14], Workspace [15] (contact domain)
  - Amazon Music [16], Yelp Restaurant, Yelp Bar [17] (review domain)
  - Devops, Patents [18] (online Q&A domain)

Experimental Settings (cont.)

- **6 metrics:**
  - Type-s affinity ratio scores **(T2, T3, T4)**: fine-grained patterns in hyperedge attribute dist.
  - Hyperedge entropy **(HE, HOHE)**: coarse-grained patterns in hyperedge attribute dist.
  - Node homophily score **(NHS)**: node-level patterns of attribute dist.



*For each **metric distribution**, calculate the difference between the **input hypergraph** and the **synthetic hypergraph**.*

Performance Comparison

- NoAH reproduces most realistic structure-attribute interplay.
  - Column T2, T3, T4, HE, HOHE, NHS: difference with the original hypergraph
  - Column A.R.: average ranking over measures

	T2	T3	T4	HE	HOHE	NHS	A.R.
HyperCL	27.3	53.0	63.6	1.016	0.382	1.053	6.8
HyperPA	27.3	55.4	71.2	1.154	0.527	1.096	8.8
HyperFF	24.1	54.3	60.5	0.449	0.299	1.055	5.2
HyperLAP	27.3	52.4	68.3	1.026	0.361	1.042	6.3
hyper dK-series	31.4	52.4	61.6	1.249	0.450	1.026	7.3
THera	26.0	50.6	67.0	0.976	0.394	1.003	5.0
HyCoSBM	11.8	57.9	72.1	0.306	0.371	0.900	4.7
HyRec	25.3	50.6	61.8	1.138	0.402	0.982	5.3
NoAH	21.0	47.8	55.1	0.275	0.394	0.229	1.8
NoAH-CF	21.8	49.7	58.0	0.363	1.188	0.402	3.7

Results on Amazon Music dataset

Rank 1
Rank 2
Rank 3

∗NoAH-CF: NoAH without core-fringe hierarchy

Performance Comparison (cont.)

- NoAH reproduces most realistic structure-attribute interplay.
  - Each column indicates **average ranking** over 9 datasets

	T2	T3	T4	HE	HOHE	NHS	A.R.
HyperCL	6.9	5.7	5.0	6.6	5.3	6.6	6.2
HyperPA	7.8	8.6	7.4	7.7	8.0	7.2	9.7
HyperFF	5.2	6.3	6.2	5.1	5.4	6.1	5.7
HyperLAP	5.3	5.3	5.3	6.2	2.8	5.6	4.7
hyper dK-series	6.1	3.9	4.7	4.3	5.4	4.6	3.8
THera	4.9	4.0	5.8	5.2	4.0	5.0	3.8
HyCoSBM	1.2	4.8	6.4	4.1	6.0	6.7	5.3
HyRec	7.7	7.1	3.9	5.6	5.1	4.0	5.0
NoAH	2.9	2.1	3.9	3.4	4.4	1.4	1.5
NoAH-CF	7.0	7.2	6.3	6.8	8.4	7.9	9.0

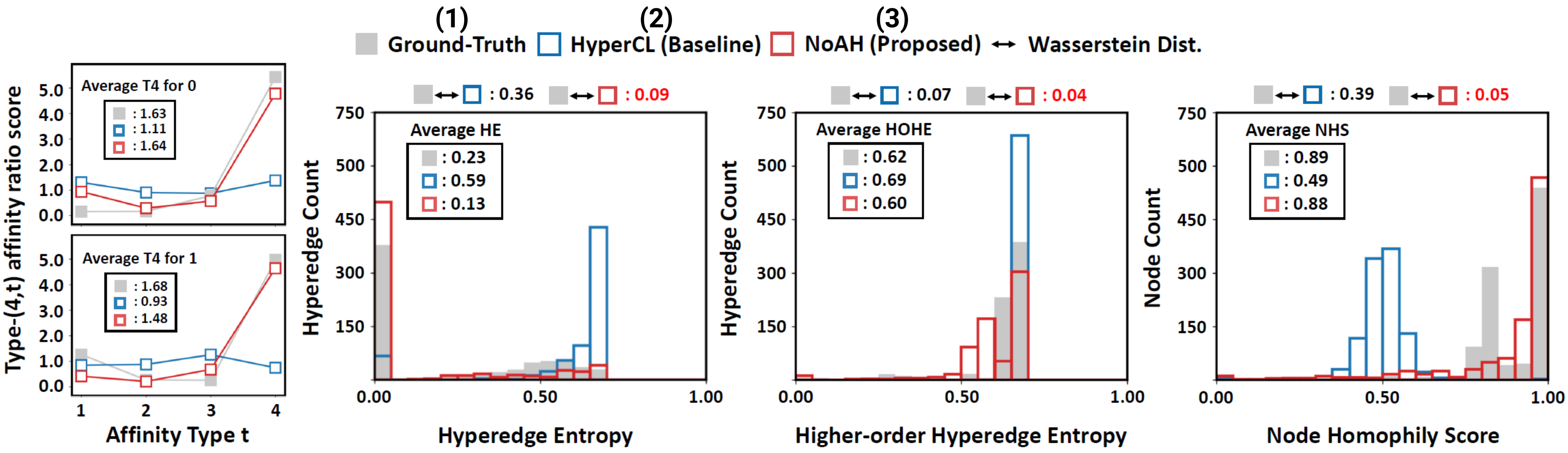
Overall results

Rank 1
Rank 2
Rank 3

∗NoAH-CF: NoAH without core-fringe hierarchy

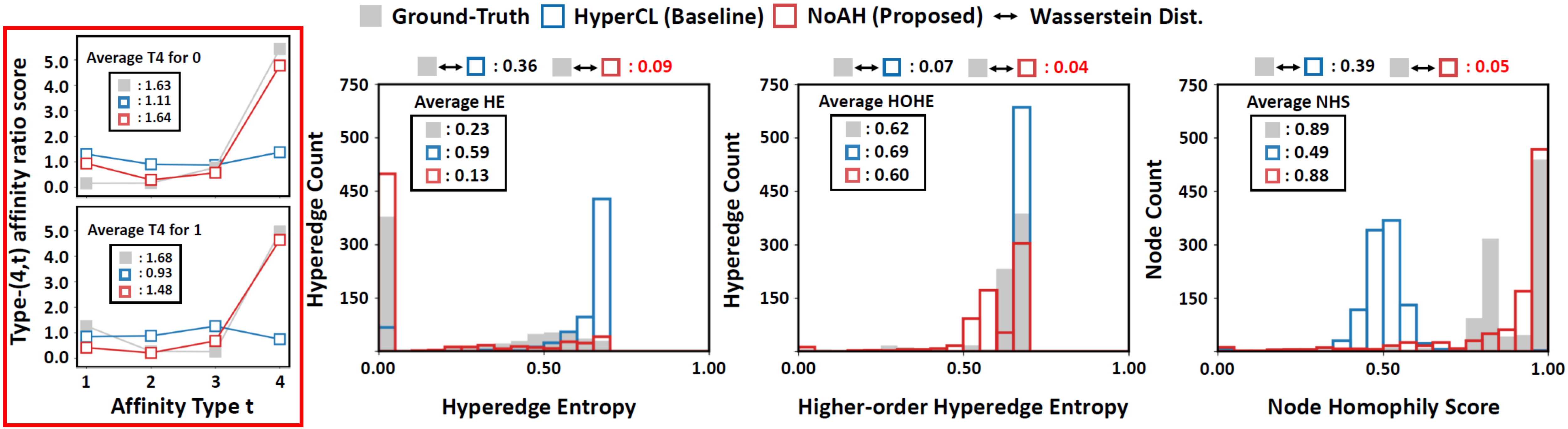
Case Study

- We compare **(1) original hypergraph**, hypergraphs generated by **(2) HyperCL**, and **(3) NoAH**.
- An attribute in the Amazon Music: whether a reviewer has reviewed a particular genre.



Case Study (cont.)

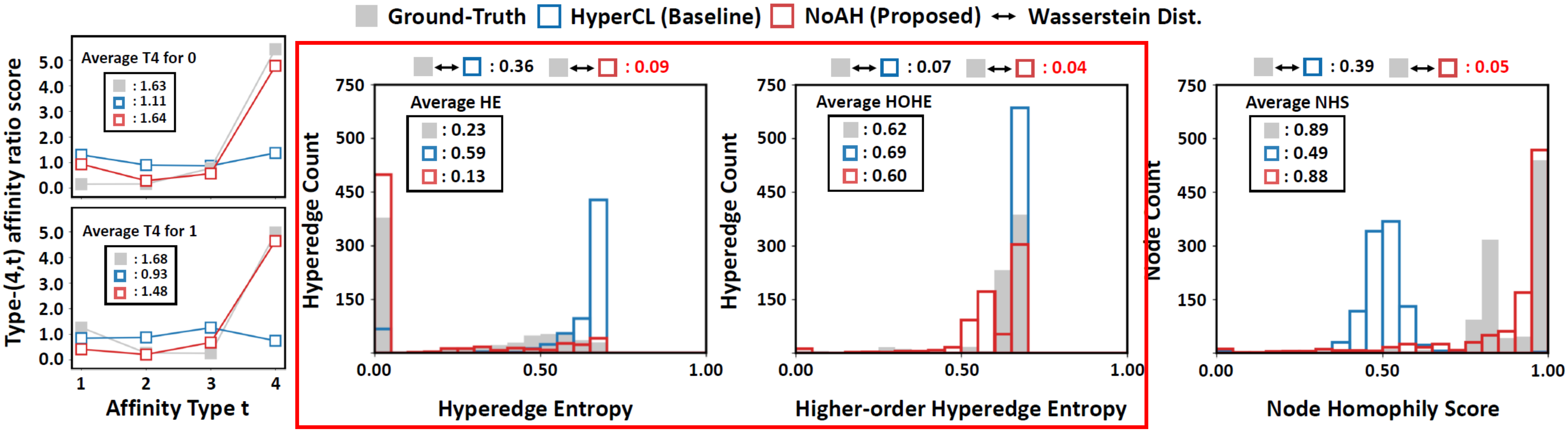
- We compare **(1) original hypergraph**, hypergraphs generated by **(2) HyperCL**, and **(3) NoAH**.
- An attribute in the Amazon Music: whether a reviewer has reviewed a particular genre.



NoAH produces similar trends for *type-4 affinity ratio scores* w.r.t. original hypergraph.
"How attribute is distributed among size-4 hyperedges"

Case Study (cont.)

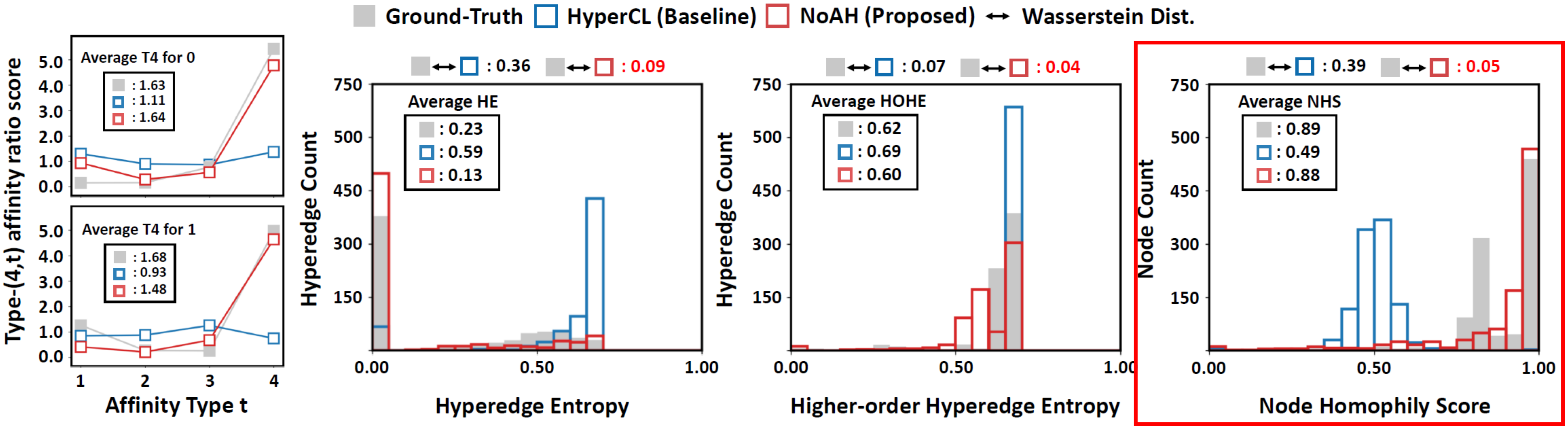
- We compare **(1) original hypergraph**, hypergraphs generated by **(2) HyperCL**, and **(3) NoAH**.
- An attribute in the Amazon Music: whether a reviewer has reviewed a particular genre.



NoAH produces similar distribution for *hyperedge entropy* w.r.t. original hypergraph.
"Hyperedge-level attribute homogeneity"

Case Study (cont.)

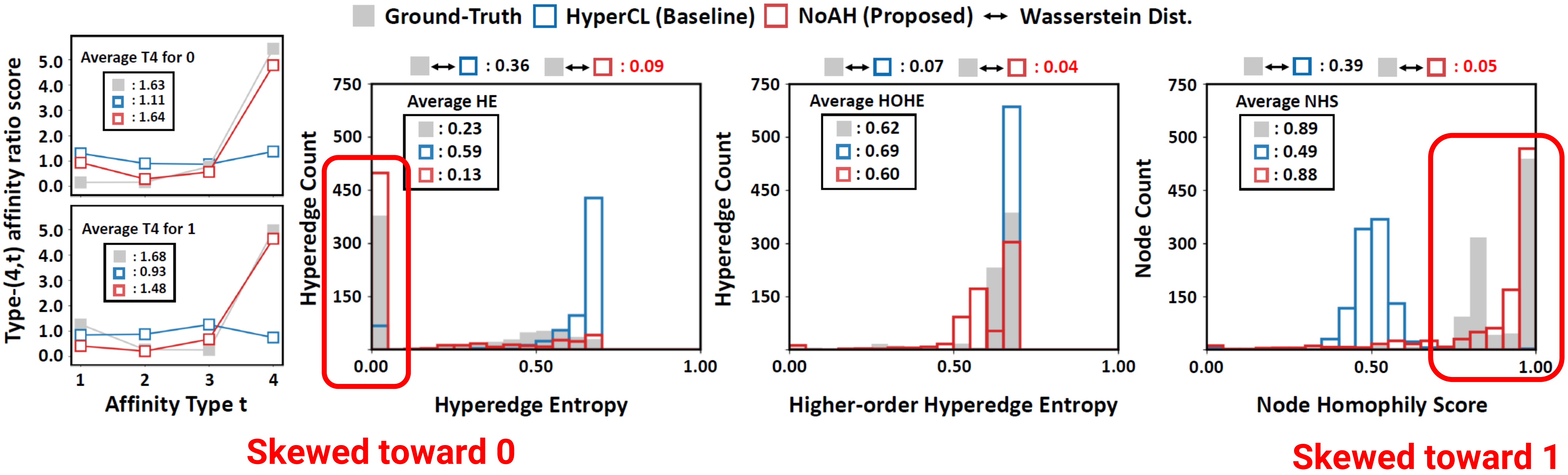
- We compare **(1) original hypergraph**, hypergraphs generated by **(2) HyperCL**, and **(3) NoAH**.
- An attribute in the Amazon Music: whether a reviewer has reviewed a particular genre.



NoAH produces similar distribution with original hypergraph for node homophily score.
"Node-level attribute homogeneity"

Case Study (cont.)

- First attribute of Amazon Music dataset exhibits strong **homophily**.
- NoAH reproduces this homophily accurately which HyperCL fails to capture.



Case Study (cont.)

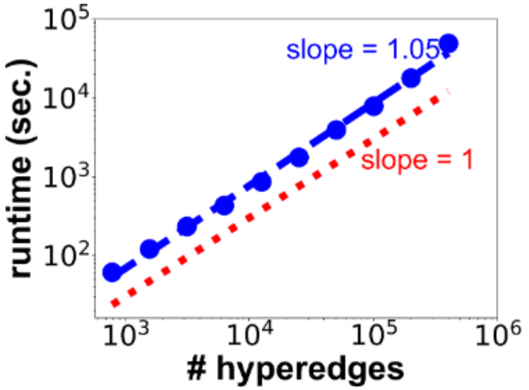
- This homophilic characteristic is reflected in the affinity matrices fitted by NoAHFit.
  - Diagonal entries are edge formation weights between nodes with the same attribute values.
  - Diagonal entries are larger than the entry for 0-1 pair.

Affinity matrices estimated by NoAHFit

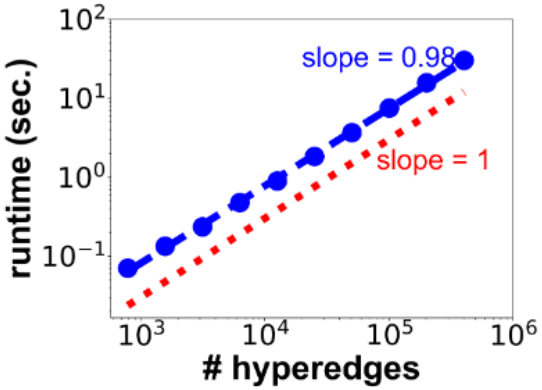
$$\theta_{\mathcal{C}}^{(1)}: \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.373 & 0.324 \\ 1 & 0.324 & 0.406 \end{array} \qquad \theta_{\mathcal{F}}^{(1)}: \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0.804 & 0.070 \\ 1 & 0.070 & 0.861 \end{array}$$

Scalability Analysis

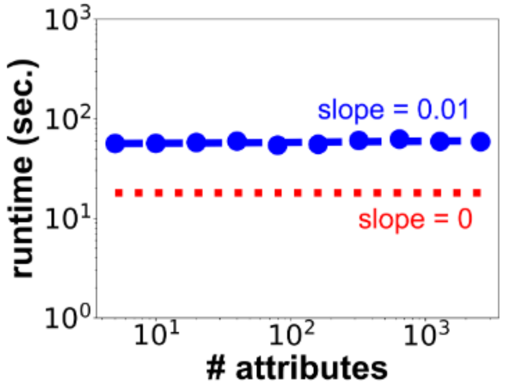
- Both NoAH and NoAHFit are scalable with respect to the number of hyperedges and attributes.



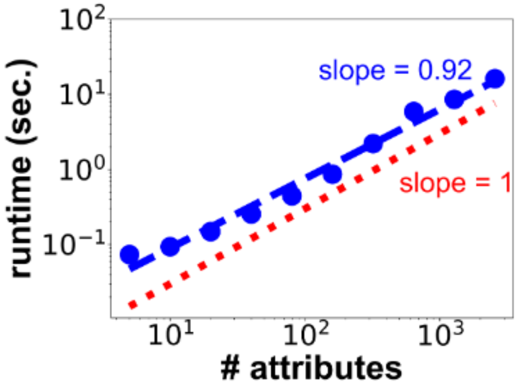
(a) NoAHFit (fitting) w.r.t. number of hyperedge



(b) NoAH (generation) w.r.t. number of hyperedge



(c) NoAHFit (fitting) w.r.t. number of attributes



(d) NoAH (generation) w.r.t. number of attributes

W.r.t. number of hyperedges

W.r.t. number of attributes

Roadmap

- Overview
- Preliminaries
- Proposed Method: NoAH, NoAHFit
- Experiments
- Conclusion <<

Conclusions

- We propose NoAH, a stochastic model for attributed hypergraphs that produces a realistic interplay between structure and attributes.
- We develop NoAHFit, a parameter fitting algorithm for NoAH that captures the relationship between structure and node attributes in a given hypergraph.
- We empirically demonstrate that NoAH outperforms eight baseline hypergraph generative models.

Github: https://github.com/jaewan01/NoAH

ICDM 2025
November 12–15, 2025
Washington DC, USA

IEEE International Conference on Data Mining

KAIST

Attributed Hypergraph Generation with Realistic Interplay Between Structure and Attributes

Best Paper Award


Jaewan Chun*


Seokbum Yoon*


Minyoung Choe


Geon Lee


Kijung Shin

References

[1]: G. Lee, M. Choe, and K. Shin, “How do hyperedges overlap in real-world hypergraphs?-patterns, measures, and generators,” in WWW, 2021.

[2]: S. Kim, F. Bu, M. Choe, J. Yoo, and K. Shin, “How transitive are real-world group interactions?-measurement and reproduction,” in KDD, 2023.

[3]: M. T. Do, S. Yoon, B. Hooi, and K. Shin, “Structural patterns and generative models of real-world hypergraphs,” in KDD, 2020.

[4]: F. Bu, G. Lee, and K. Shin, “Hypercore decomposition for non-fragile hyperedges: concepts, algorithms, observations, and applications,” Data Mining and Knowledge Discovery, 2023.

[5]: F. Tudisco and D. J. Higham, “Core-periphery detection in hypergraphs,” SIAM Journal on Mathematics of Data Science, 2023.

[6]: M. Kim and J. Leskovec, “Multiplicative attribute graph model of real-world networks,” Internet Mathematics, 2012.

[7]: N. Veldt, A. R. Benson, and J. Kleinberg, “Combinatorial characterizations and impossibilities for higher-order homophily,” Science Advances, 2023.

[8]: G. Lee, S. Y. Lee, and K. Shin, “Villain: Self-supervised learning on homogeneous hypergraphs without features via virtual label propagation,” in WWW, 2024.

[9]: Y. Kook, J. Ko, and K. Shin, “Evolution of real-world hypergraphs: Patterns and models without oracles,” in ICDM, 2020.

[10]: K. Nakajima, K. Shudo, and N. Masuda, “Randomizing hypergraphs preserving degree correlation and local clustering,” TNSE, 2021.

[11]: A. Badalyan, N. Ruggeri, and C. De Bacco, “Structure and inference in hypergraphs with node attributes,” Nature Communications, 2024.

References

[12]: M. Choe, J. Ko, T. Kwon, K. Shin, and C. Faloutsos, “Kronecker generative models for power-law patterns in real-world hypergraphs,” in WWW, 2025.

[13]: N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar,“Hypergcn: A new method for training graph convolutional networks on hypergraphs,” in NeurIPS, 2019.

[14]: P. S. Chodrow, N. Veldt, and A. R. Benson, “Hypergraph clustering: from block models to modularity,” Science Advances, 2021.

[15]: M. G´enois and A. Barrat, “Can co-location be used as a proxy for face-to-face contacts?,” EPJ Data Science, 2018.

[16]: J. Ni, J. Li, and J. McAuley, “Justifying recommendations using distantly-labeled reviews and fine-grained aspects,” in EMNLP, 2019.

[17]: I. Amburg, N. Veldt, and A. R. Benson, “Fair clustering for diverse and experienced groups,” arXiv:2006.05645, 2020.

[18]: https://archive.org/download/stackexchange

[19]: M. Contisciani, F. Battiston, and C. De Bacco, “Inference of hyperedges and overlapping communities in hypergraphs,” Nature Communications, 2022.

[20]: G. Lee, J. Ko, and K. Shin, Hypergraph motifs: Concepts, algorithms, and discoveries. Proceedings of the VLDB Endowment, 2020