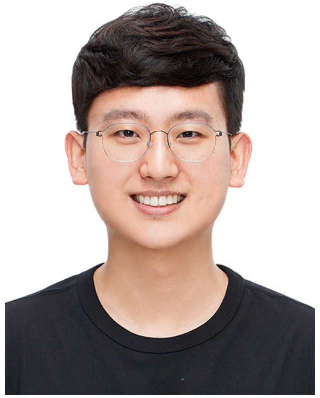
# How Transitive Are Real-World Group Interactions? Measurement and Reproduction







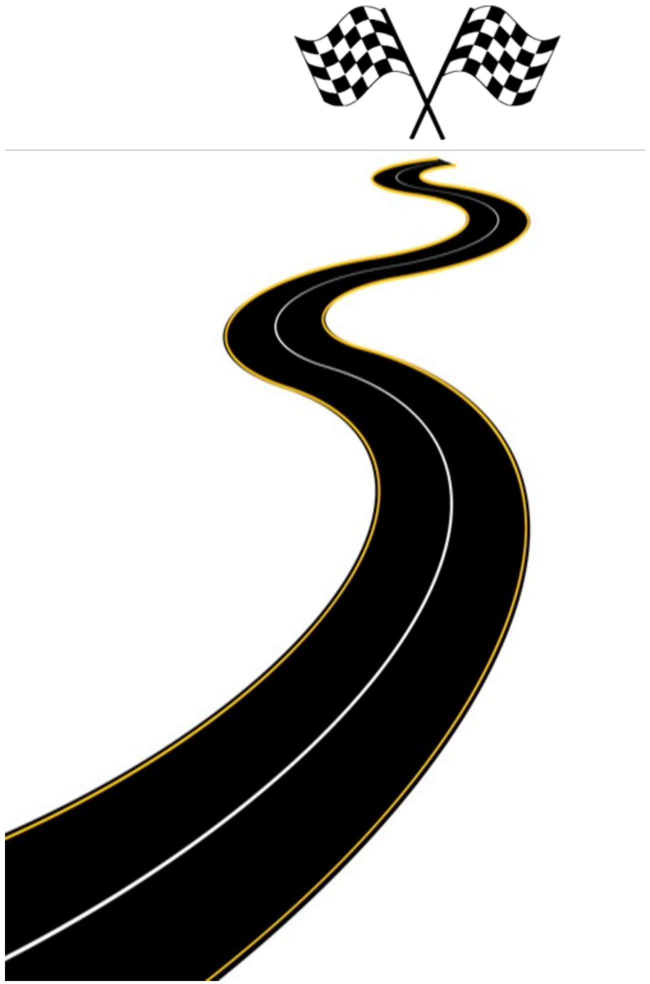

Sunwoo Kim | Fanchen Bu | Minyoung Choe | Jaemin Yoo | Kijung Shin

KDD2023 AUGUST 6 - 10 LONG BEACH, CA

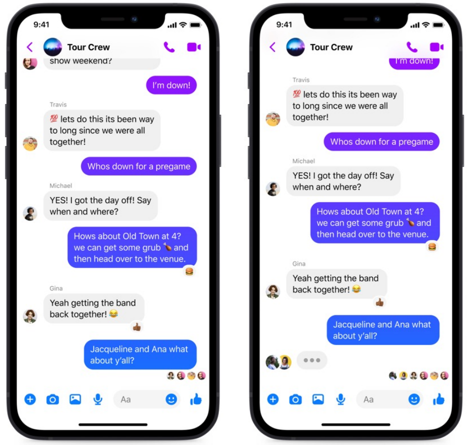
KAIST AI Kim Jaechul Graduate School

# Roadmap

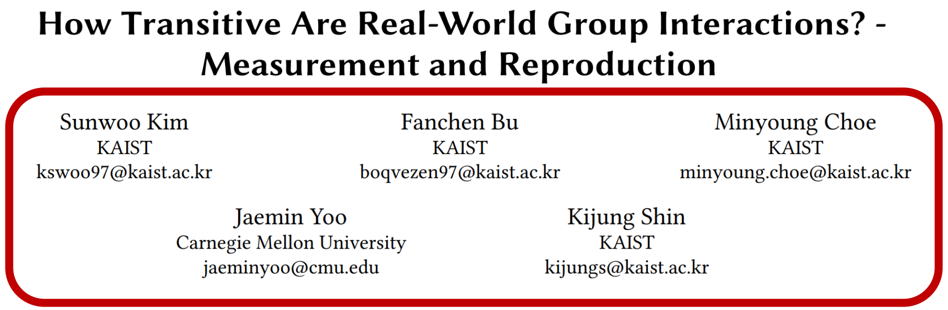
- Overview
- Transitivity Measure
- Observations
- Generator
- Conclusions



# Group Interactions are EVERYWHERE!



Group Chat


How Transitive Are Real-World Group Interactions? - Measurement and Reproduction

Sunwoo Kim
KAIST
kswoo97@kaist.ac.kr

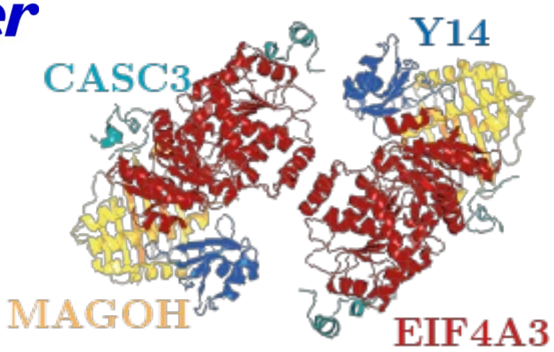
Fanchen Bu
KAIST
boqvezen97@kaist.ac.kr

Minyoung Choe
KAIST
minyoung.choe@kaist.ac.kr

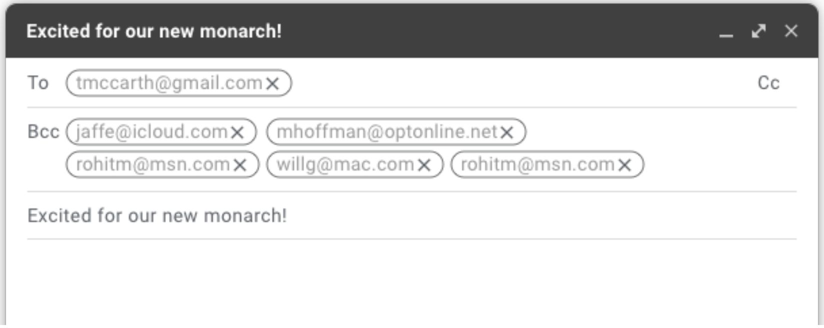
Jaemin Yoo
Carnegie Mellon University
jaeminyoo@cmu.edu

Kijung Shin
KAIST
kijungs@kaist.ac.kr

Co-Authorship




Protein Interaction




Email

# Group Interactions → HYPERGRAPH!

**How Transitive Are Real-World Group Interactions? - Measurement and Reproduction**

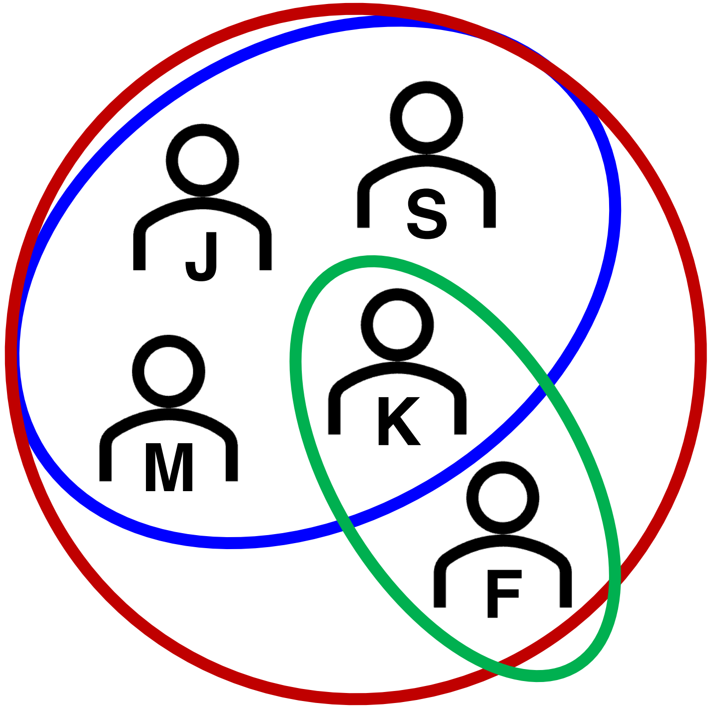
Sunwoo Kim, KAIST, kswoo97@kaist.ac.kr
Fanchen Bu, KAIST, boqvezen97@kaist.ac.kr
Minyoung Choe, KAIST, minyoung.choe@kaist.ac.kr
Jaemin Yoo, Carnegie Mellon University, jaeminyoo@cmu.edu
Kijung Shin, KAIST, kijungs@kaist.ac.kr

**Classification of Edge-dependent Labels of Nodes in Hypergraphs**

Minyoung Choe, KAIST, Seoul, Korea, minyoung.choe@kaist.ac.kr
Sunwoo Kim, KAIST, Seoul, Korea, kswoo97@kaist.ac.kr
Jaemin Yoo, Carnegie Mellon University, Pittsburgh, PA, USA, jaeminyoo@cmu.edu
Kijung Shin, KAIST, Seoul, Korea, kijungs@kaist.ac.kr

**On Improving the Cohesiveness of Graphs by Merging Nodes: Formulation, Analysis, and Algorithms**

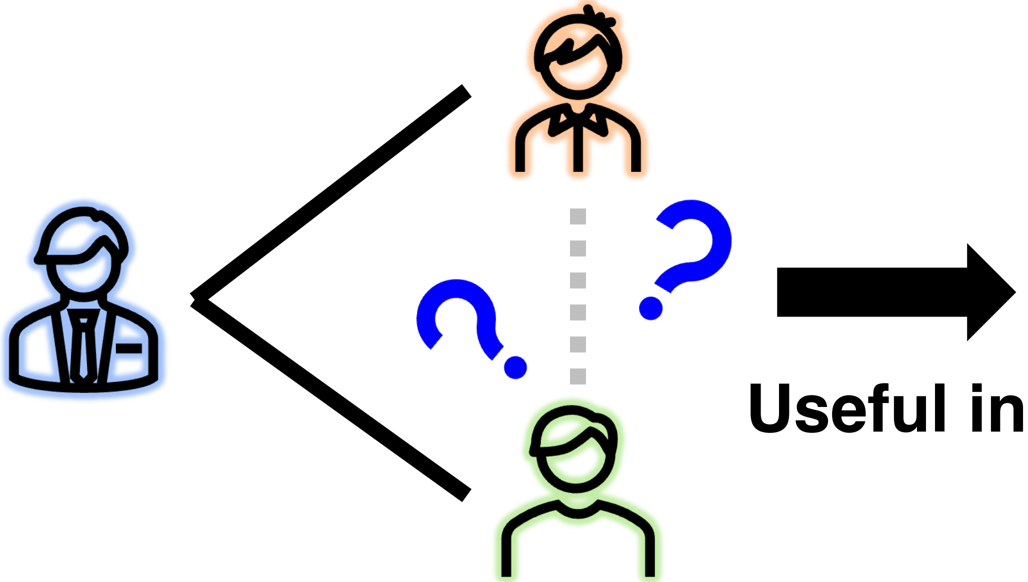
Fanchen Bu, School of Electrical Engineering, KAIST, Daejeon, South Korea, boqvezen97@kaist.ac.kr
Kijung Shin, Kim Jaechul Graduate School of AI, KAIST, Seoul, South Korea, kijungs@kaist.ac.kr



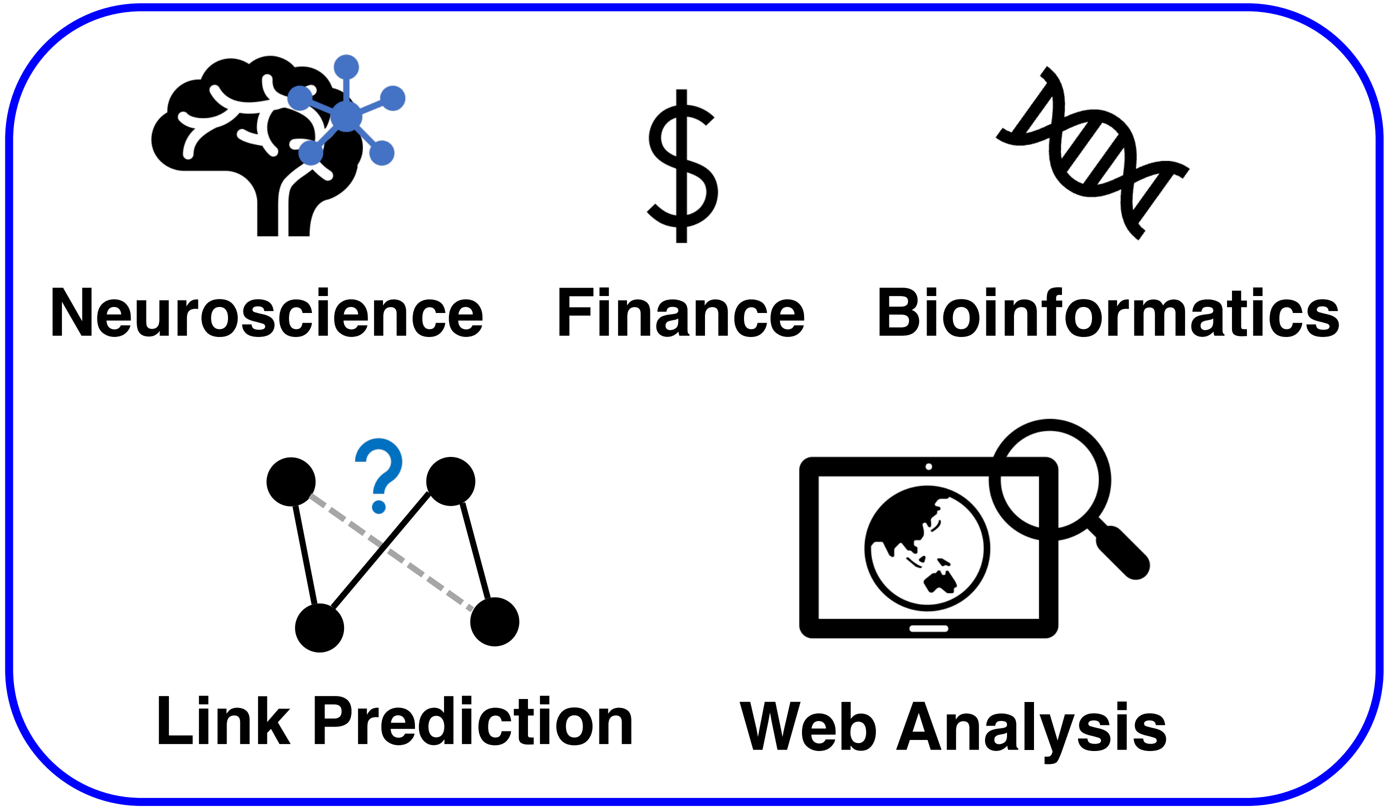

**Co-Authorship Hypergraph**

# Transitivity in a GRAPH

- Probability of two neighbors of a node **being also adjacent.**

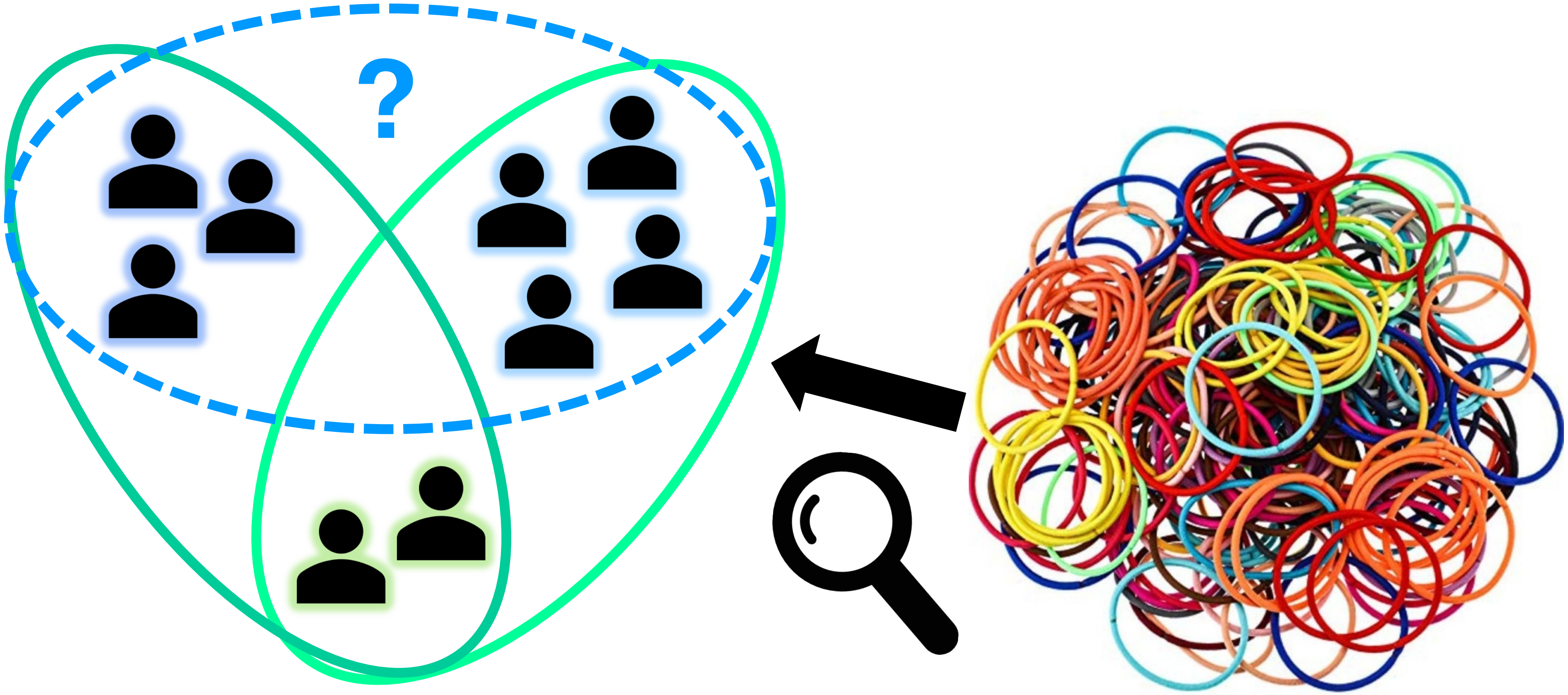


Useful in



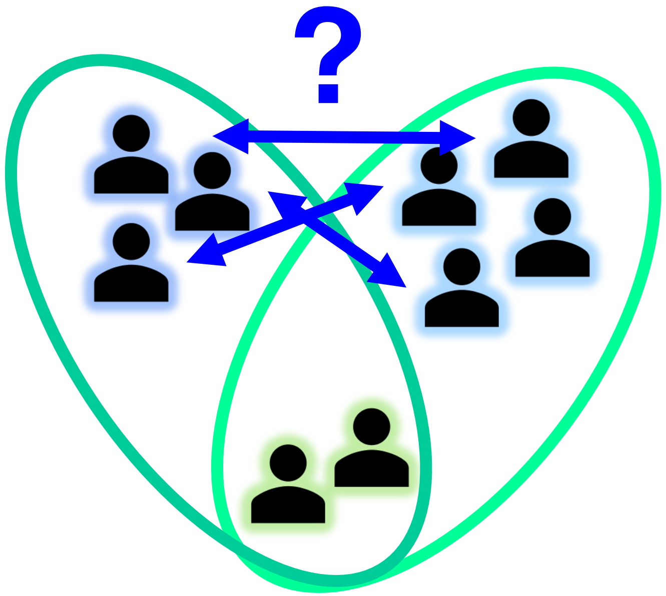
# Transitivity of GROUP INTERACTIONS?

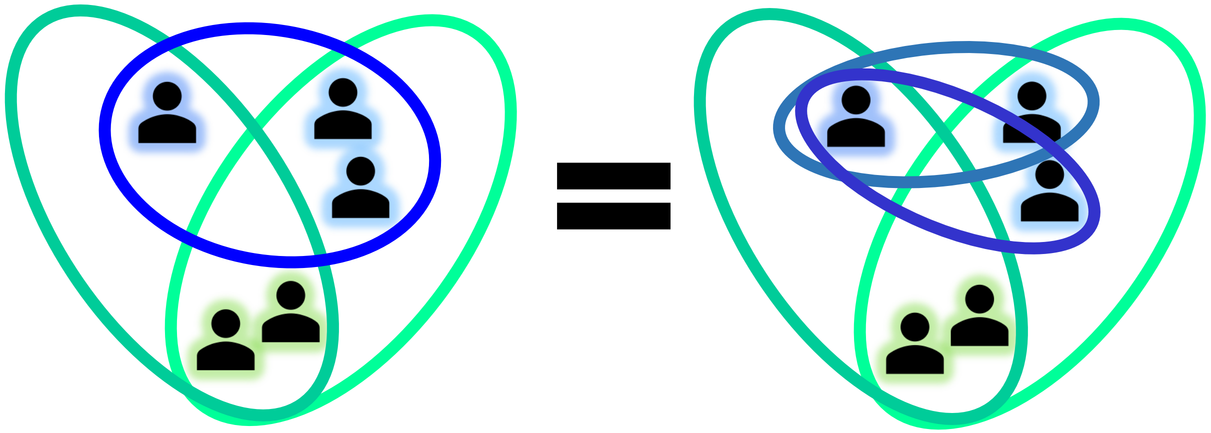
- How can we measure the **transitivity of group interactions?**



# Transitivity of GROUP INTERACTIONS?

- Existing hypergraph transitivity measures **focus on pairwise interactions** rather than group interactions.



**Inappropriate for capturing the transitivity of group interactions**

**Focus:**

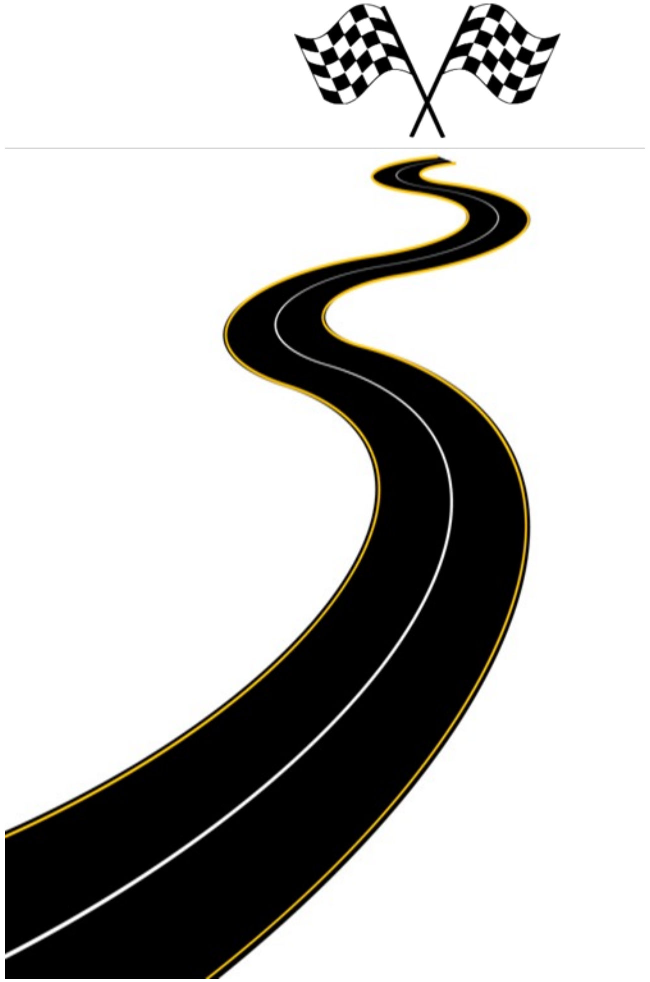
**Are they neighbors?**

**Limitation:**

**Equivalent in terms of transitivity**

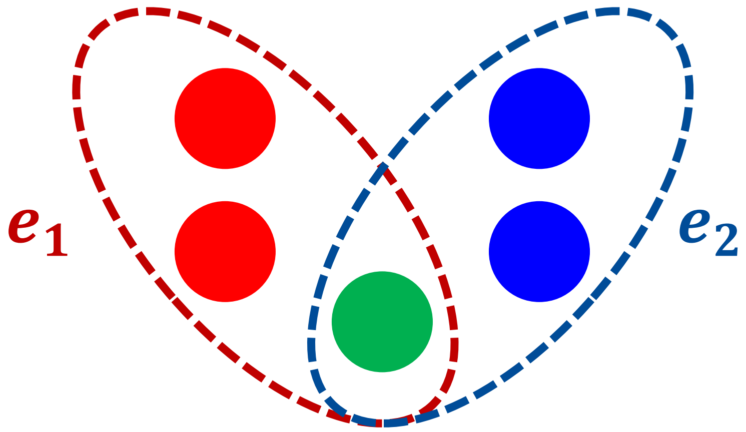
# Roadmap

- Overview
- **Transitivity Measure**
- Observations
- Generators
- Conclusions



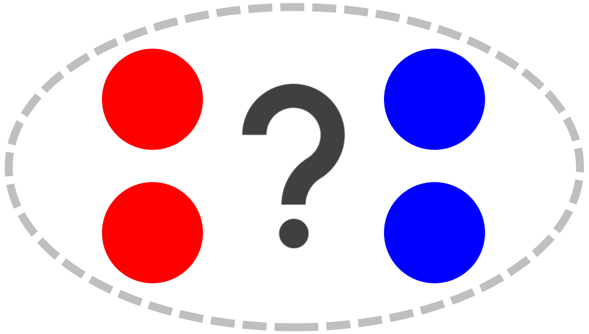
# Basic Terminology & Concept

- We describe several basic terminologies.



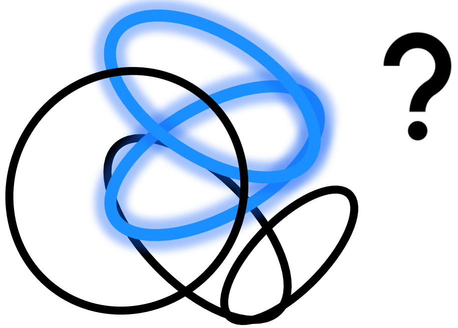

- **Hyperwedge $w$:** $\{e_1, e_2\}$
- **Left-wing $L(w)$:**
- **Right-wing $R(w)$:**
- **Body-group $B(w)$:**

- We aim to quantify the group interactions between the nodes in the two wings: $L(w)$ and $R(w)$.

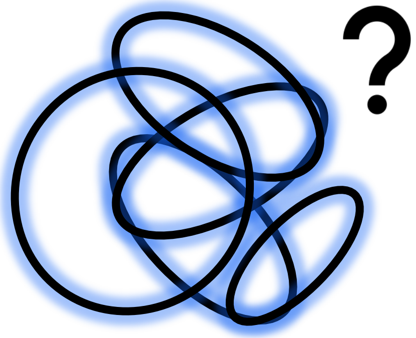


# Basic Terminology & Concept

- We define transitivity measures on two different levels.
- **Hyperwedge-level ($\mathcal{T}$):** Transitivity of a given hyperwedge $\boldsymbol{w}$.
- **Hypergraph-level ($T$):** Transitivity of a given hypergraph $\boldsymbol{G}$.




**Hyperwedge-level Transitivity**




**Hypergraph-level Transitivity**

# Criteria of a principled measure

- How can we say a specific measure is principled (captures the transitivity of group interactions well)?
- We provide **7 AXIOMs** to formally describe such desirable properties of a principled transitivity measure.

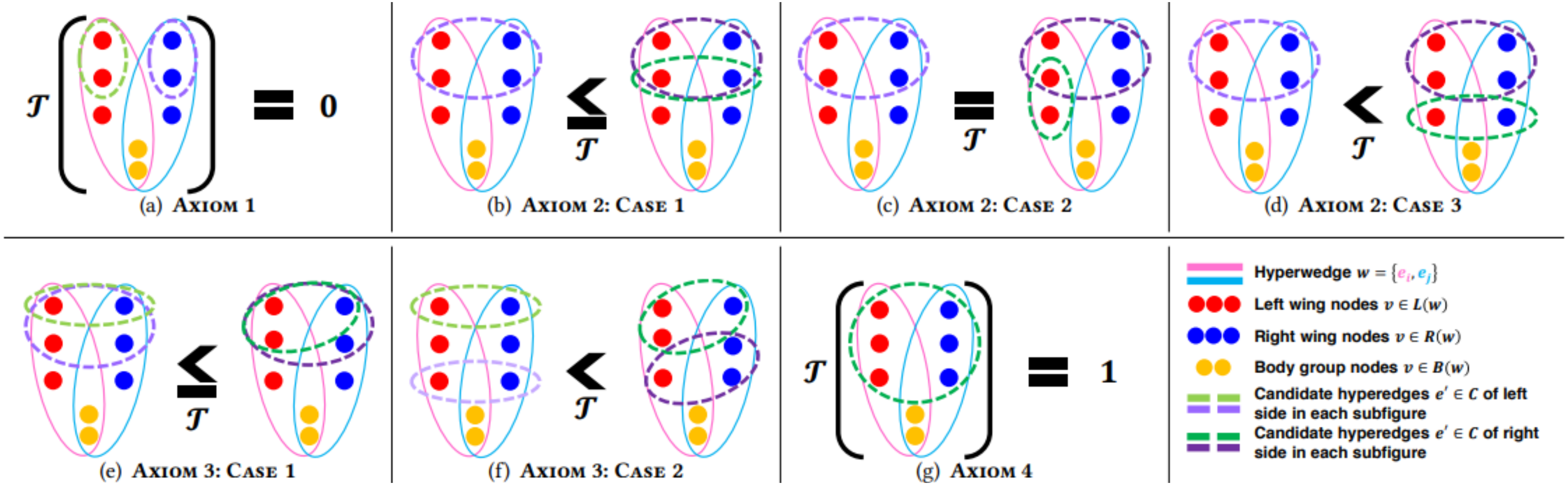
| Measures | Satisfies Axioms? |
|---|---|
| Measure A | O |
| Measure B | X |

**Measure A: Principled** 

**Measure B: Not principled** 

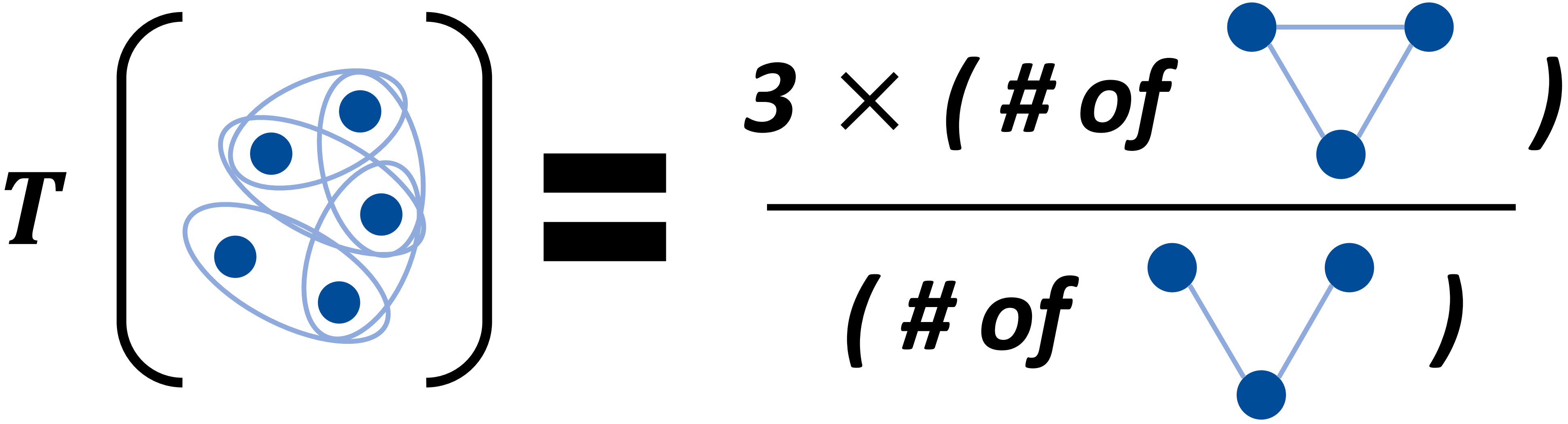
# Criteria of a principled measure

- **Axiom 1 - 5** describe conditions of a hyperwedge-level measure $\mathcal{T}$:
  - **A1** & **A4**: When the value of $\mathcal{T}$ is minimized/maximized.
  - **A2** & **A3**: How the value of $\mathcal{T}$ varies under certain situations.
  - **A5**: The bound of $\mathcal{T}$.



# Criteria of a principled measure

- **Axiom 6 & 7** describe conditions of a hypergraph-level measure $\boldsymbol{T}$:
  - **A6**: How should the value of $\boldsymbol{T}(\boldsymbol{G})$ be when $\boldsymbol{G}$ is an ordinary graph.
  - **A7**: The bound of $\boldsymbol{T}$.



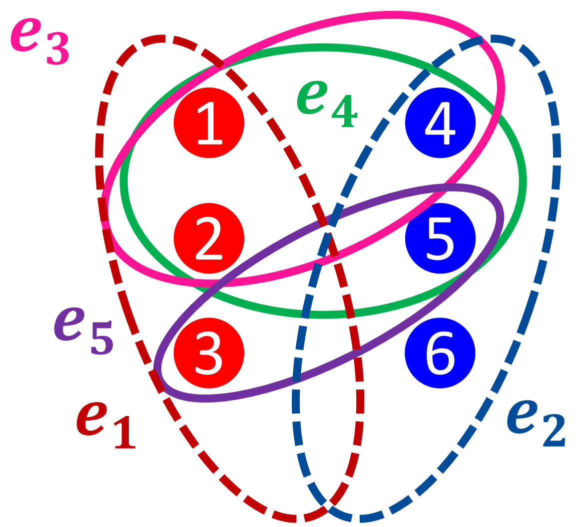
# Proposed Measure

- We describe our proposed transitivity measure: **HyperTrans**.
- Most importantly, **HyperTrans** is a principled transitivity measure.

| Measure | Axioms 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| **B1** (Jaccard index) | ✗ | ✗ | ✗ | ✗ | ✔ | ✔ | ✔ |
| **B2** (Ratio of covered interacations) | ✔ | ✔ | ✗ | ✗ | ✔ | ✔ | ✔ |
| **B3** (Klamt et al. [29]) | ✔ | ✗ | ✗ | ✗ | ✔ | ✔ | ✔ |
| **B4** (Torres et al. [47]) | ✔ | ✔ | ✗ | ✗ | ✔ | ✔ | ✔ |
| **B5** (Gallager et al. [20] A) | ✗ | ✗ | ✗ | ✗ | ✔ | ✔ | ✔ |
| **B6** (Gallager et al. [20] B) | ✗ | ✗ | ✗ | ✗ | ✔ | ✗ | ✔ |
| **B7** (HyperTrans-mean) | ✔ | ✗ | ✔ | ✔ | ✔ | ✔ | ✔ |
| **B8** (HyperTrans-non-$P(w)$) | ✔ | ✗ | ✔ | ✔ | ✔ | ✔ | ✔ |
| **B9** (HyperTrans-unnormalized) | ✔ | ✔ | ✔ | ✔ | ✗ | ✔ | ✗ |
| Proposed: HyperTrans | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

# Proposed Measure

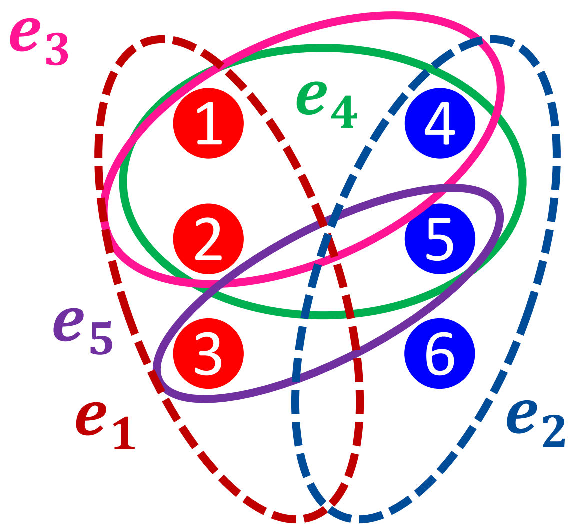
- We describe further details of **HyperTrans**.
- First, **hyperwedge-level HyperTrans** is an average of scores of pair interactions between the nodes in $L(w)$ and those in $R(w)$.




- Assume a hyperwedge $\boldsymbol{w} = \{\boldsymbol{e_1}, \boldsymbol{e_2}\}$.
- There are 3 overlapping hyperedges: $\boldsymbol{e_3}$, $\boldsymbol{e_4}$, and $\boldsymbol{e_5}$.
- 9 pair interactions between $\boldsymbol{L}(\boldsymbol{w})$ and $\boldsymbol{R}(\boldsymbol{w})$ are

1 ↔ 4 1 ↔ 5 1 ↔ 6 2 ↔ 4 2 ↔ 5
2 ↔ 6 3 ↔ 4 3 ↔ 5 3 ↔ 6

# Proposed Measure




| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | | | | | | | | | |

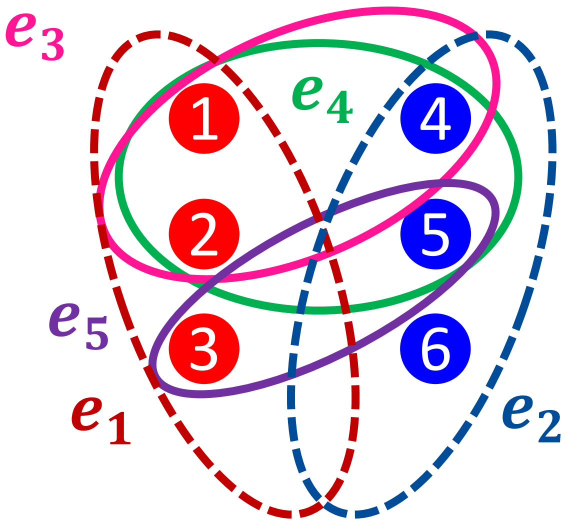
**Computing score of a pair interaction $\{v_i, v_j\}$:**

Step 1) Find hyperedges that contain $\{v_i, v_j\}$.

Step 2) Compute scores of the found hyperedges.

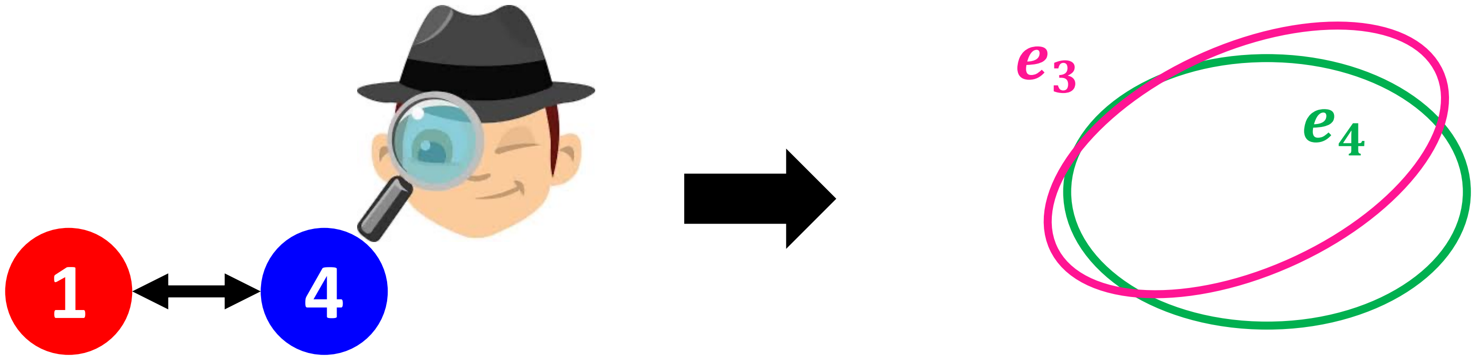
Step 3) Find the maximum score among them → score of $\{v_i, v_j\}$.

# Proposed Measure

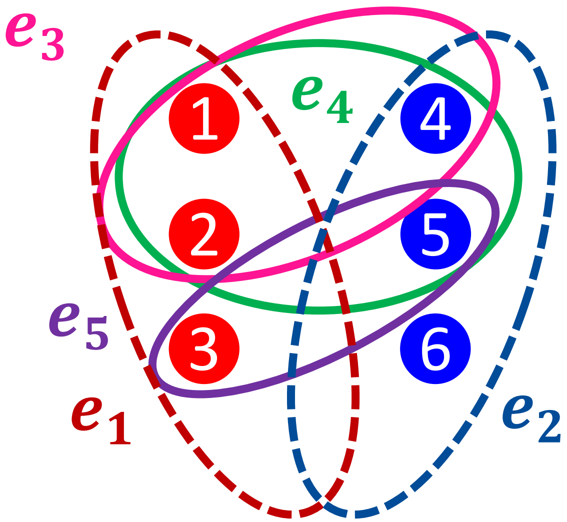



| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | | | | | | | | | |

Step 1) Find hyperedges that contain $\{v_i, v_j\}$.



# Proposed Measure



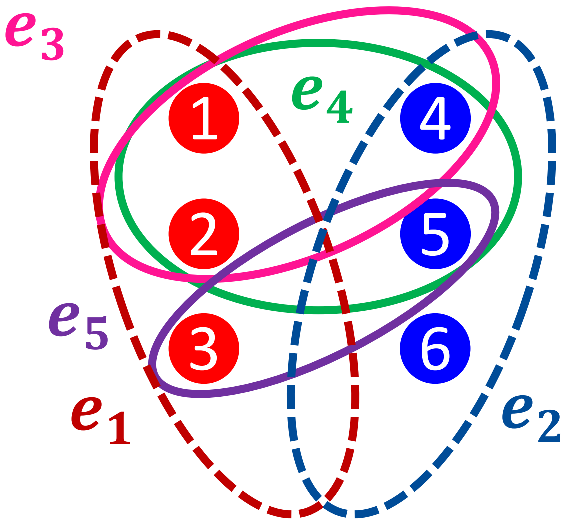

| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | | | | | | | | | |

Step 2) Compute scores of the found hyperedges.

- **Score of $e$ in $w$** is the quantified contribution of $e$ to the interaction between $L(w)$ and $R(w)$.

E.g. $$\frac{|L(w) \cap e| \times |R(w) \cap e|}{|(L(w) \cup e) \setminus R(w)| \times |(R(w) \cup e) \setminus L(w)|}$$

# Proposed Measure

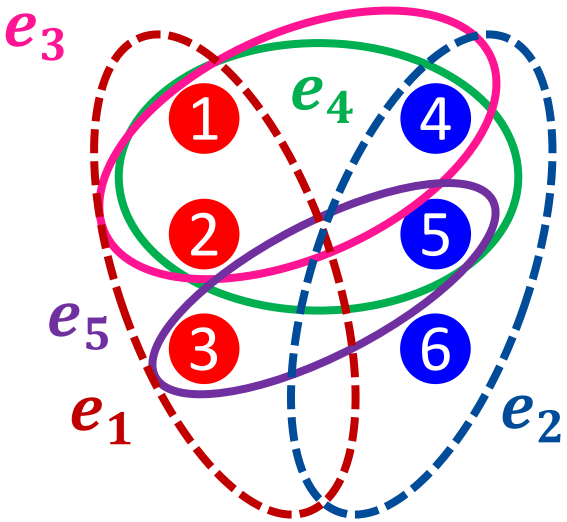



| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | | | | | | | | | |

Step 2) Compute scores of the found hyperedges.

| Hyperedges | $e_3$ | $e_4$ |
|---|---|---|
| Score | $\frac{2\times1}{3\times3}=\frac{2}{9}$ | $\frac{2\times2}{3\times3}=\frac{4}{9}$ |

# Proposed Measure

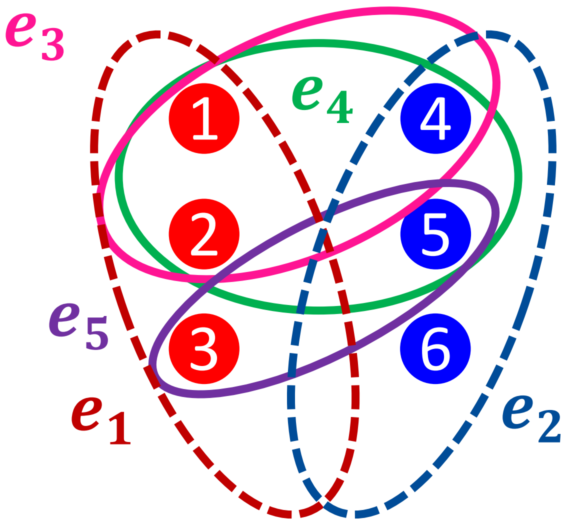



| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | 4/9 | | | | | | | | |

Step 3) Find the maximum score among them → score of $\{v_i, v_j\}$.

| Hyperedges | $e_3$ | $e_4$ |
|---|---|---|
| Score | $\frac{2\times1}{3\times3} = \frac{2}{9}$ | $\frac{2\times2}{3\times3} = \frac{4}{9}$ |

# Proposed Measure

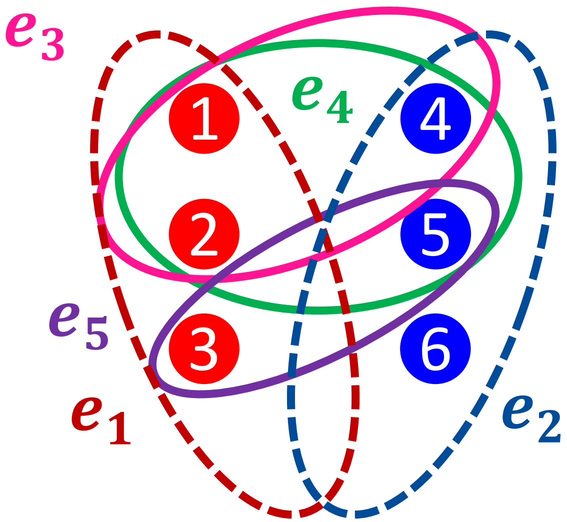



| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | 4/9 | 4/9 | | | | | | | |

**Do the same process on a pair $\{1, 5\}$.**

| Hyperedges | $e_4$ |
|---|---|
| Score | $\frac{2\times2}{3\times3} = \frac{4}{9}$ |

# Proposed Measure




| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | 4/9 | 4/9 | 0 | | | | | | |

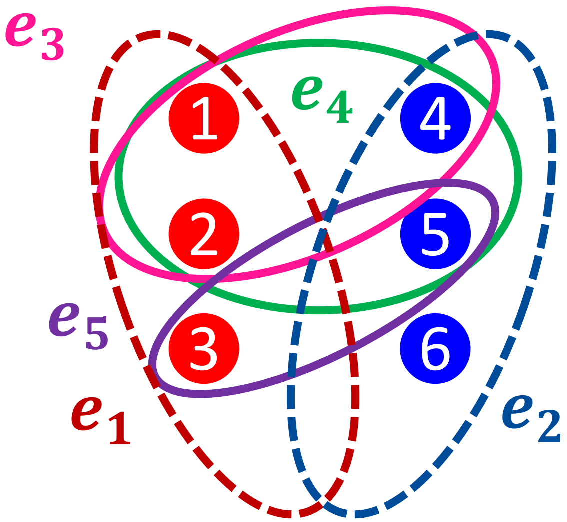
**<u>Issue: When finding hyperedges that contain $\{v_1, v_6\}$,</u>**

| Hyperedges |
|---|
| Score |

None of $\{e_3, e_4, e_5\}$ contains $\{1,6\}$ !

**Score = 0**

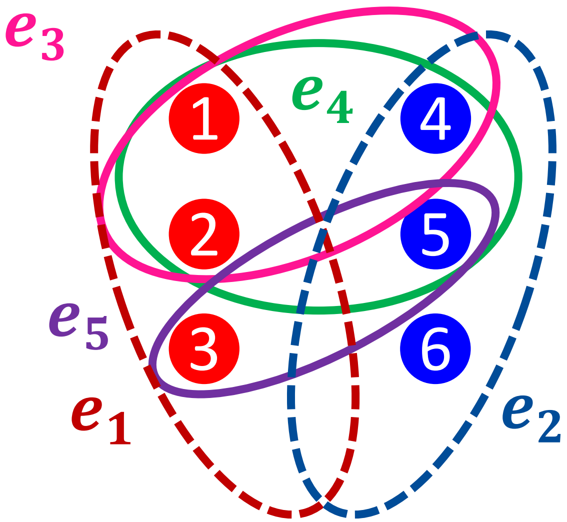
# Proposed Measure




| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | 4/9 | 4/9 | 0 | 4/9 | 4/9 | 0 | 0 | 1/9 | 0 |

**Perform this process on the entire pairs.**

# Proposed Measure



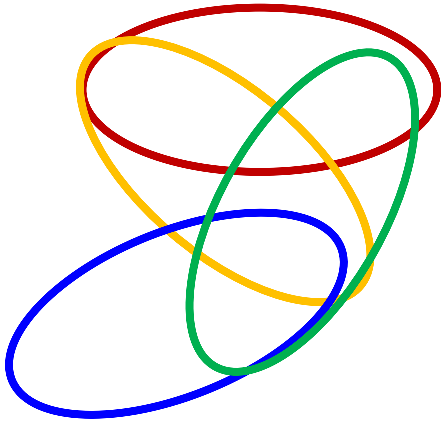

| Pairs | 1 – 4 | 1 – 5 | 1 – 6 | 2 – 4 | 2 – 5 | 2 – 6 | 3 – 4 | 3 – 5 | 3 – 6 |
|---|---|---|---|---|---|---|---|---|---|
| Score | 4/9 | 4/9 | 0 | 4/9 | 4/9 | 0 | 0 | 1/9 | 0 |

**Compute an average of all scores = Hyperwedge-level HyperTrans**

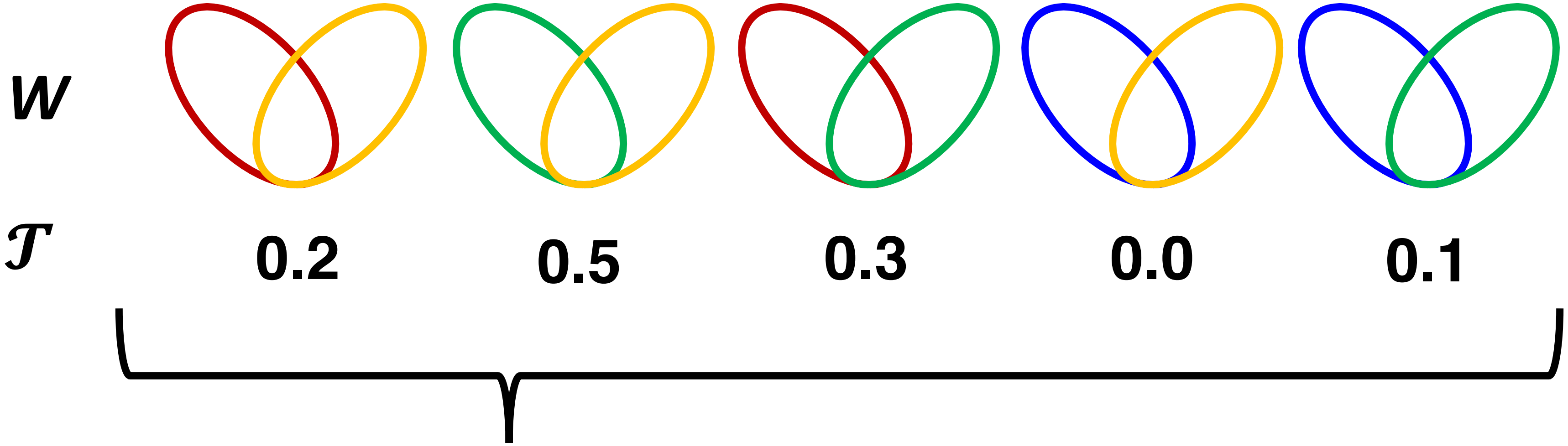
$$\frac{1}{9}\times\left(\frac{4}{9}+\frac{4}{9}+0+\frac{4}{9}+\frac{4}{9}+0+0+\frac{1}{9}+0\right)=\mathbf{\frac{17}{81}}$$

# Proposed Measure

- **Hypergraph-level HyperTrans** is an average of all hyperwedge-level HyperTrans of a given hypergraph.



a hypergraph $G$




$$\boldsymbol{T}(\boldsymbol{G}) = \frac{1}{5}(0.2 + 0.5 + 0.3 + 0.0 + 0.1) = \boldsymbol{\frac{11}{50}}$$

# Proposed Measure

- Formal expression of HyperTrans:

**Details**

$$\mathcal{T}(w) = \sum_{v \in L(w)} \sum_{v' \in R(w)} \frac{\max_{e \in E} \{ f(w,e) \times \mathbb{I}[v, v' \in e] \}}{|L(w)| \times |R(w)|}$$

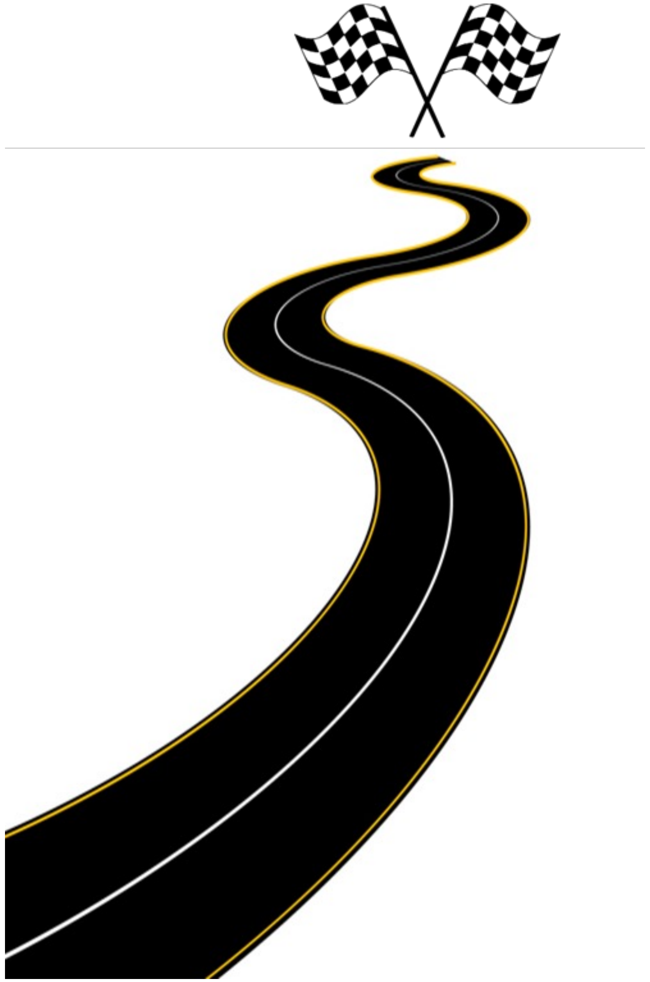
**Hyperwedge-Level**

$$\mathcal{T}(G) = \frac{1}{|W_G|} \sum_{w \in W_G} \mathcal{T}(w)$$

**Hypergraph-Level**

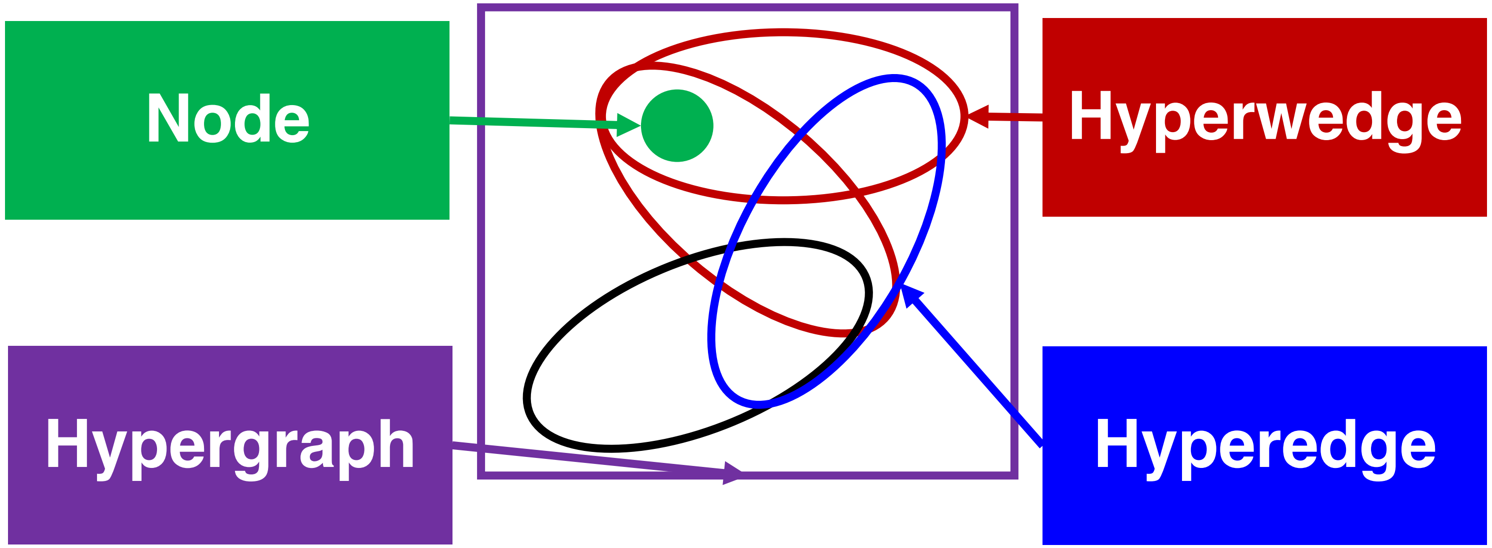
# Roadmap

- Overview
- Transitivity Measure
- **Observations**
- Generator
- Conclusions



# Level of Observations

- We analyze the **transitivity patterns** in real-world hypergraphs.
- To show that the observed patterns are not random, we use HyperCL [1] to generate random counterparts of real-world hypergraphs.




| Pattern A | Pattern B |
|---|---|
| Real HG O | Real HG O |
| Random HG X | Random HG O |


[1]: Lee, Choe, and Shin. How Do Hyperedges Overlap in Real-World Hypergraphs? – Patterns, Measures, and Generators. The Web Conference. 2021.

** HyperCL generates random hypergraphs where the expected node degree distribution is preserved.

# Observations

- Observation 1 (Hypergraph-level): Real-world hypergraphs are **more transitive** than randomized hypergraphs.

**Hypergraph Transitivity**

| Data | Real | HyperCL | Z-stat | P-value |
|---|---|---|---|---|
| email-enron | 0.195 | 0.078 | 378.3 | 0.00** |
| email-eu | 0.125 | 0.053 | 240.1 | 0.00** |
| ndc-classes | 0.052 | 0.008 | 146.7 | 0.00** |
| ndc-substances | 0.019 | 0.005 | 47.3 | 0.00** |
| contact-high | 0.345 | 0.119 | 764.7 | 0.00** |
| contact-primary | 0.336 | 0.223 | 380.7 | 0.00** |
| coauth-dblp | 0.007 | 0.000* | 23.2 | 0.00** |
| coauth-geology | 0.005 | 0.000* | 16.6 | 0.00** |
| coauth-history | 0.002 | 0.000* | 6.6 | 0.00** |
| qna-ubuntu | 0.005 | 0.014 | 32.0 | 0.00** |
| qna-server | 0.005 | 0.017 | 38.3 | 0.00** |
| qna-math | 0.025 | 0.040 | 46.6 | 0.00** |

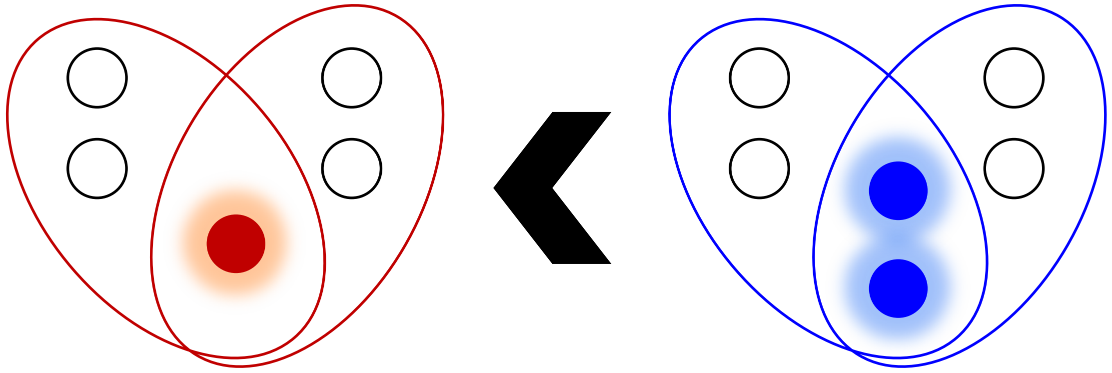
Differences are all ***statistically significant*** (at $\alpha = 0.05$).

# Observations

- Observation 2 (Hyperwedge-level): There exist **positive correlations** between body-group sizes and hyperwedge transitivities in real-world hypergraphs.

**Rank Correlation**

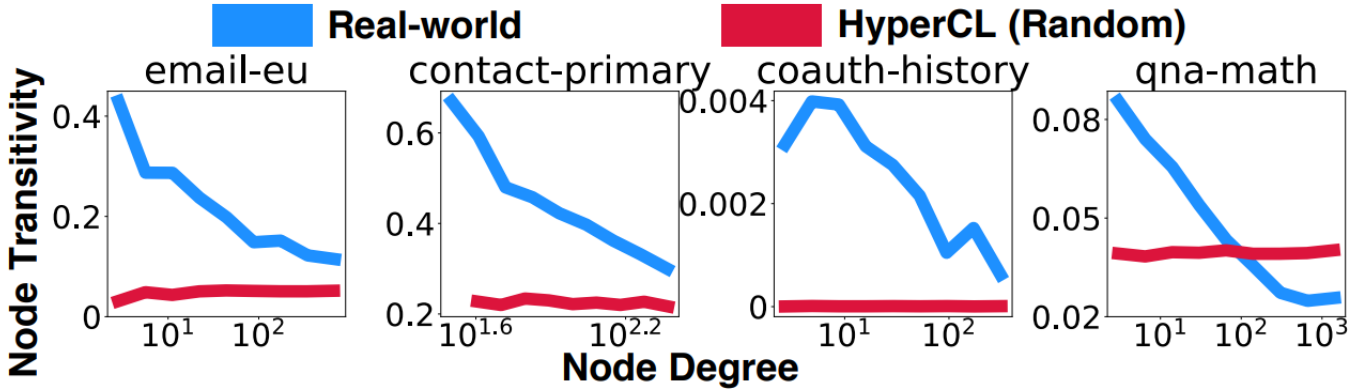
| Data | Real | HyperCL |
|---|---|---|
| email-enron | 0.09 | -0.09 |
| email-eu | 0.12 | -0.14 |
| ndc-classes | 0.32 | -0.10 |
| ndc-substances | 0.14 | -0.10 |
| contact-high | 0.13 | 0.00* |
| contact-primary | 0.13 | 0.00* |
| coauth-dblp | 0.12 | 0.00* |
| coauth-geology | 0.14 | 0.00* |
| coauth-history | 0.12 | 0.05 |
| qna-ubuntu | 0.04 | 0.00* |
| qna-server | 0.04 | 0.00* |
| qna-math | 0.04 | 0.01 |

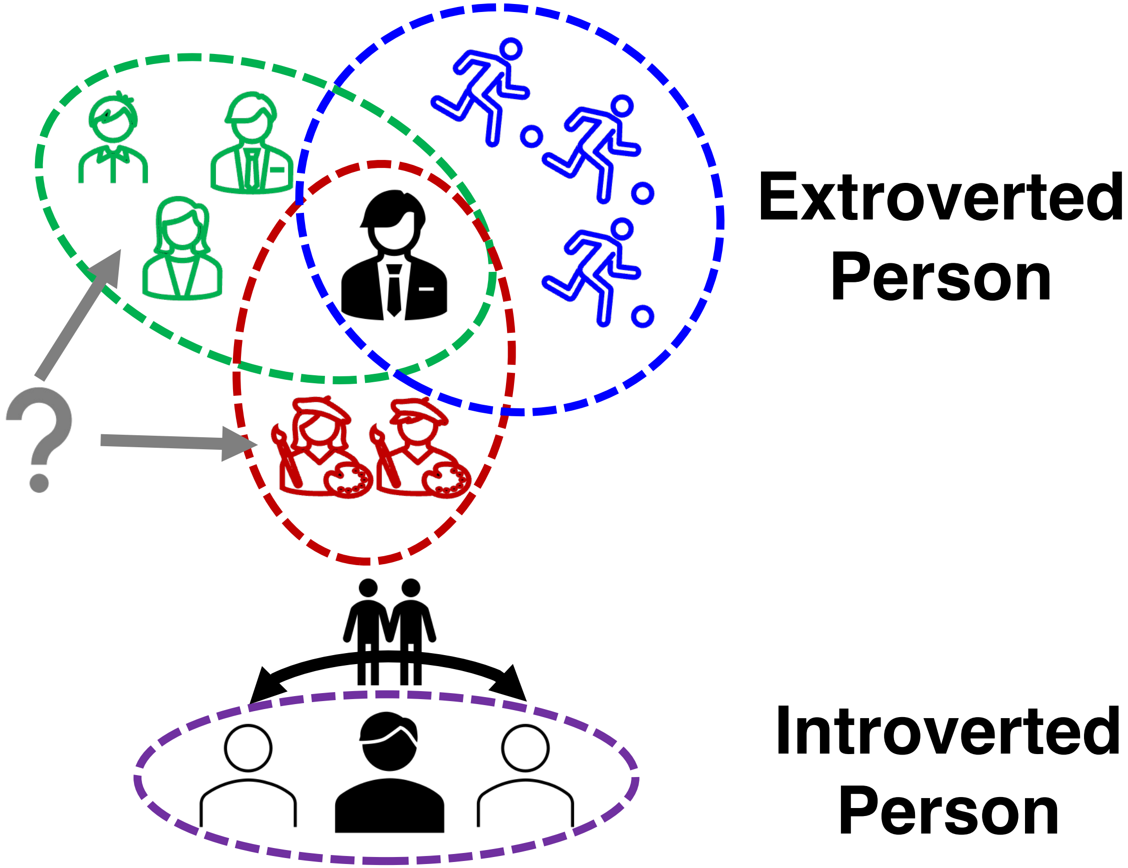


**More Transitive!**

# Observations

- Observation 3 (Node-level): In real-world hypergraphs, the transitivity of a node is **negatively correlated** to its degree.



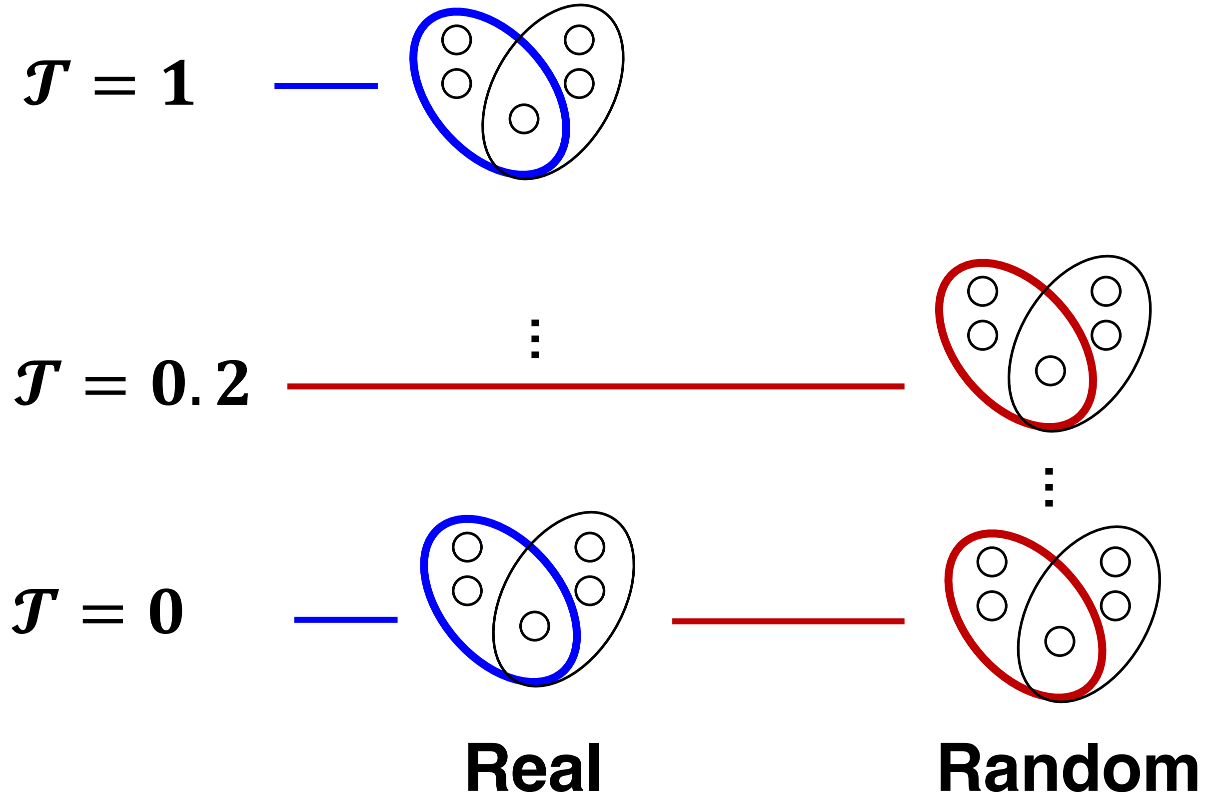



# Observations

- Observation 4 (Hyperedge-level): Real-world hypergraphs have **wider ranges** of hyperedge transitivity than their random counterparts.

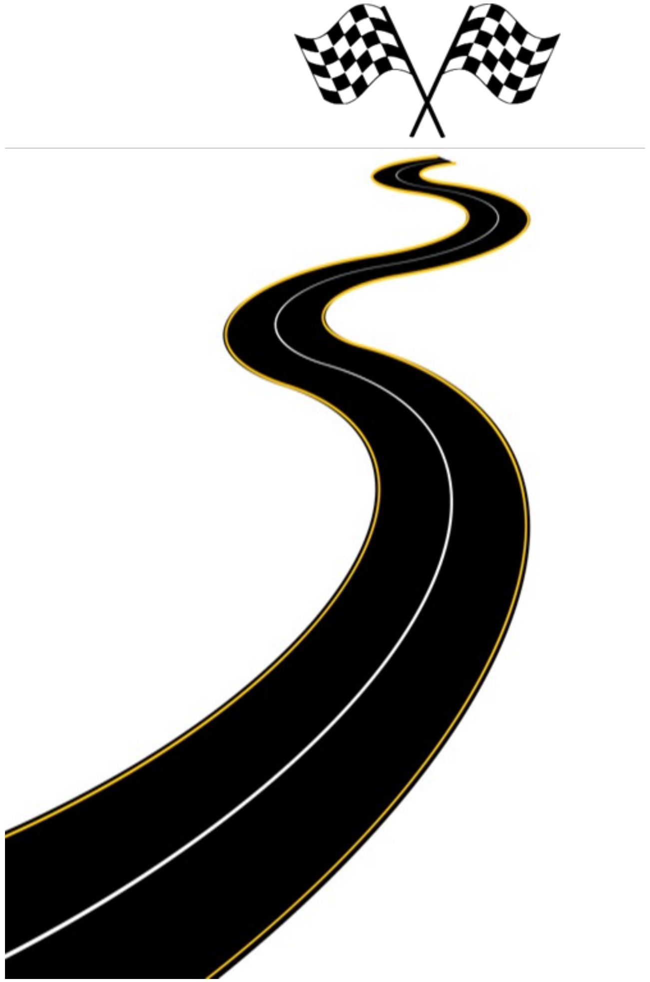
**Transitivity Range**

| Data | Real | HyperCL | THERA |
|---|---|---|---|
| email-enron | 0.725 | 0.279 | 0.732 |
| email-eu | 0.809 | 0.248 | 0.792 |
| ndc-classes | 0.600 | 0.075 | 0.410 |
| ndc-substances | 1.0 | 0.032 | 0.411 |
| contact-high | 0.794 | 0.316 | 0.768 |
| contact-primary | 0.693 | 0.395 | 0.839 |
| coauth-dblp | 1.0 | 0.105 | 1.0 |
| coauth-geology | 1.0 | 0.069 | 1.0 |
| coauth-history | 1.0 | 0.333 | 1.0 |
| qna-ubuntu | 0.667 | 0.5 | 1.0 |
| qna-server | 0.667 | 0.333 | 1.0 |
| qna-math | 0.667 | 1.00 | 1.0 |



# Roadmap

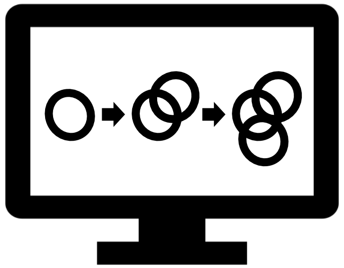
- Overview
- Transitivity Measure
- Observations
- **Generator**
- Conclusions



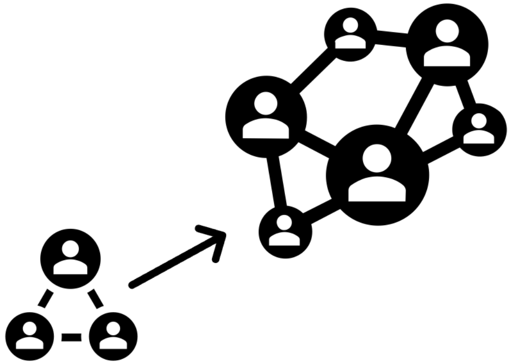
# THera: Transitivity-preserving Generator

- A **realistic synthetic (hyper)graph** is useful in…

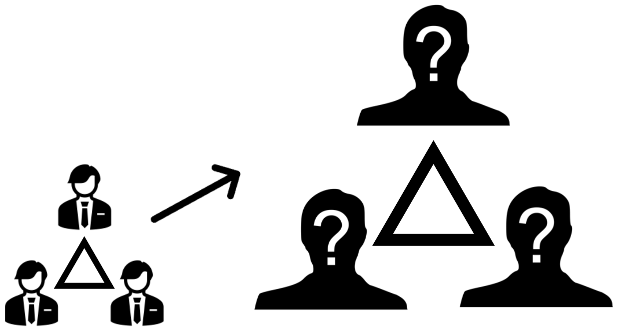
**Finding causes of patterns**



**Simulation**



**Upscaling**



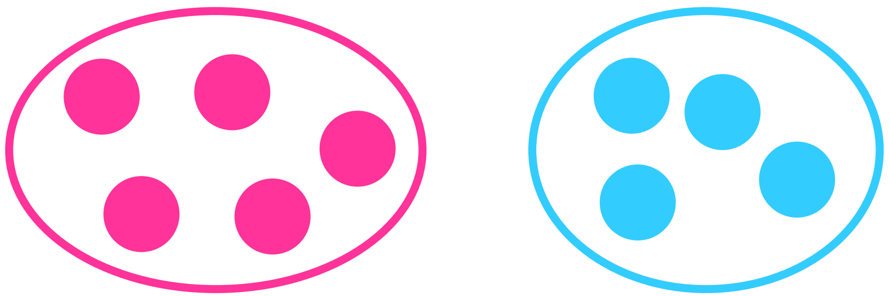
**Anonymization**

# THera: Transitivity-preserving Generator

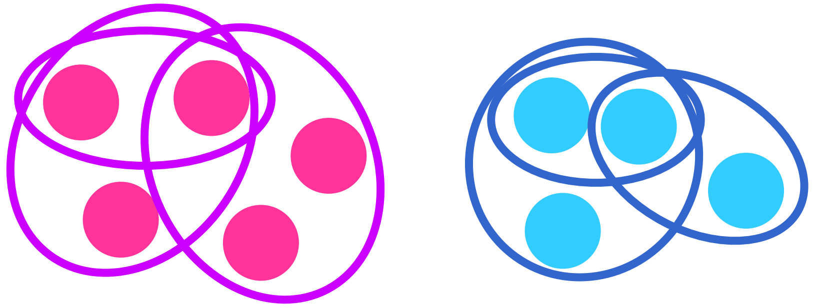
- We propose a transitivity-preserving hypergraph generator, namely

**THera: T**ransitive **H**ypergraph gen**ERA**tor.

- **THera** is based on a community structure of nodes:



**Community 1** **Community 2**

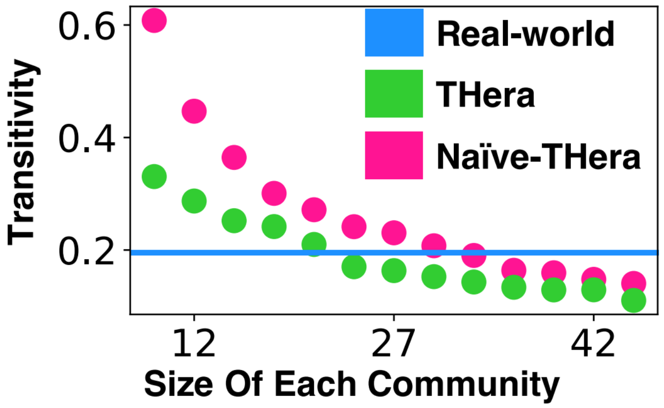


**Creating hyperedges within communities**

# THera: Transitivity-preserving Generator

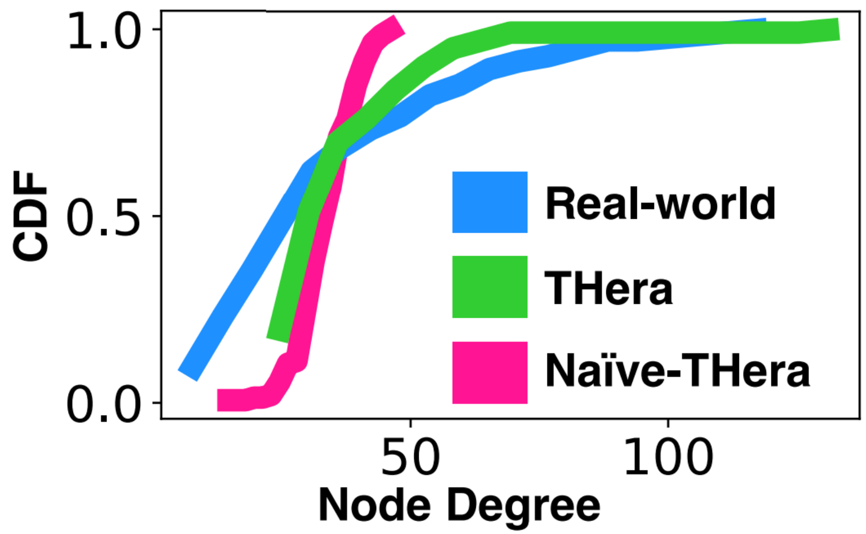
- We first introduce our preliminary approach: **Naïve-THera**.
- Naïve-THera assigns each node to a community and creates intra-community hyperedges only.
- By adjusting the community sizes, we can control the transitivity of generated hypergraphs.

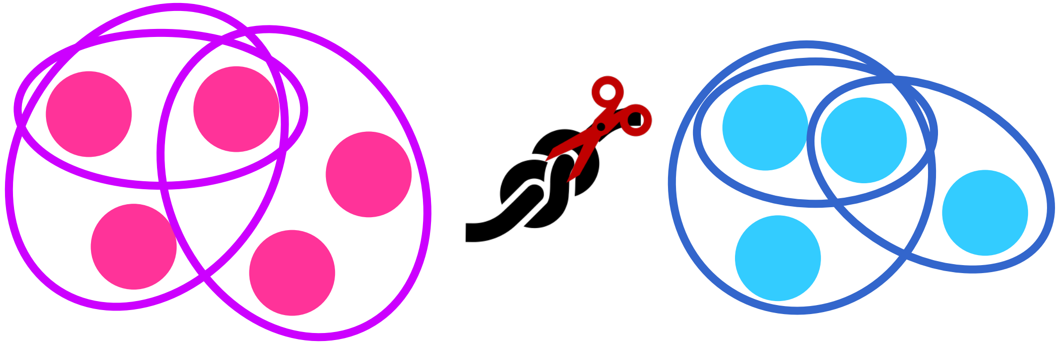
**See the pink dotted line!**



# THera: Transitivity-preserving Generator

- However, Naïve-THera makes **unrealistic hypergraphs**:
  - It creates near-uniform degree distribution.
  - It creates disconnected hypergraphs.

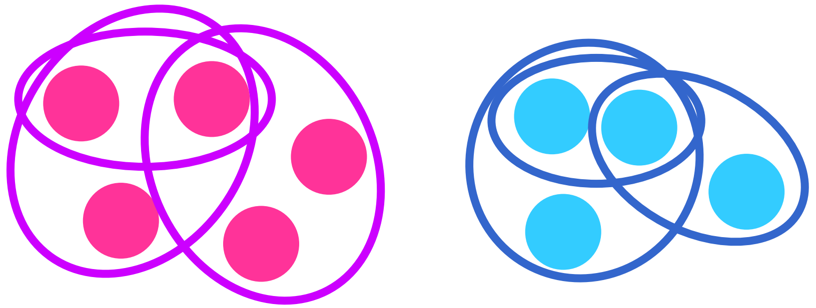


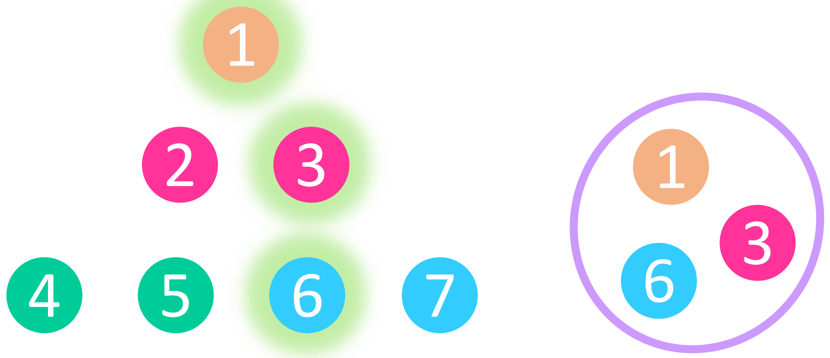
# THera: Transitivity-preserving Generator

- **THera (Our proposal)** has two types of hyperedge generation, and the second type of hyperedge mitigates such issues:

**`#1) IntraCommunity:`** Assigns each node to a community and creates hyperedges within each community.



**`#2) Hierarchical:`** Assigns each node a level in the tree and creates hyperedges by sampling nodes proportional to the level of each node.

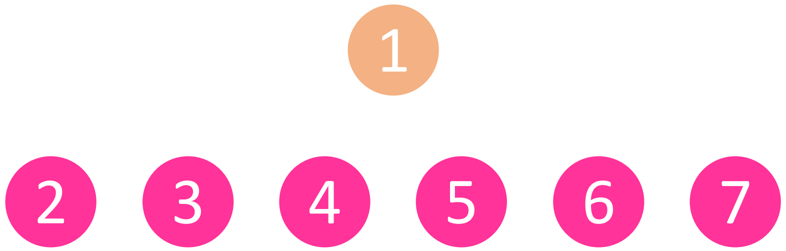


# THera: Transitivity-preserving Generator

1. Start from node 1.
2. Introduce $C = 6$ nodes.
3. Decide hyperedge type ($\propto \boldsymbol{p}$) & size.
4. If `IntraCommunity:` create a hyperedge within the community.
5. If `Hierarchical:` sample nodes depending on the level and fill a hyperedge with them.

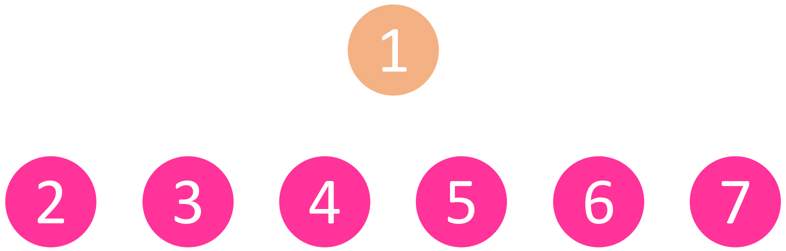
# THera: Transitivity-preserving Generator

1. Start from node 1.
2. Introduce $C = 6$ nodes.
3. Decide hyperedge type ($\propto p$) & size.
4. If `IntraCommunity:` create a hyperedge within the community.
5. If `Hierarchical:` sample nodes depending on the level and fill a hyperedge with them.



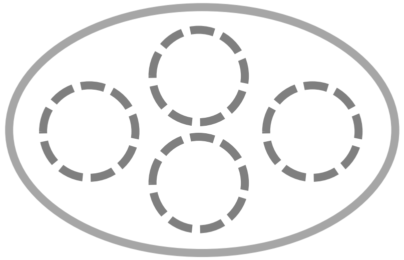
# THera: Transitivity-preserving Generator

1. Start from node 1.
2. Introduce $C = 6$ nodes.
3. Decide hyperedge type ($\propto \boldsymbol{p}$) & size.
4. If `IntraCommunity:` create a hyperedge within the community.
5. If `Hierarchical:` sample nodes depending on the level and fill a hyperedge with them.




$\propto \boldsymbol{p}$) `IntraCommunity`

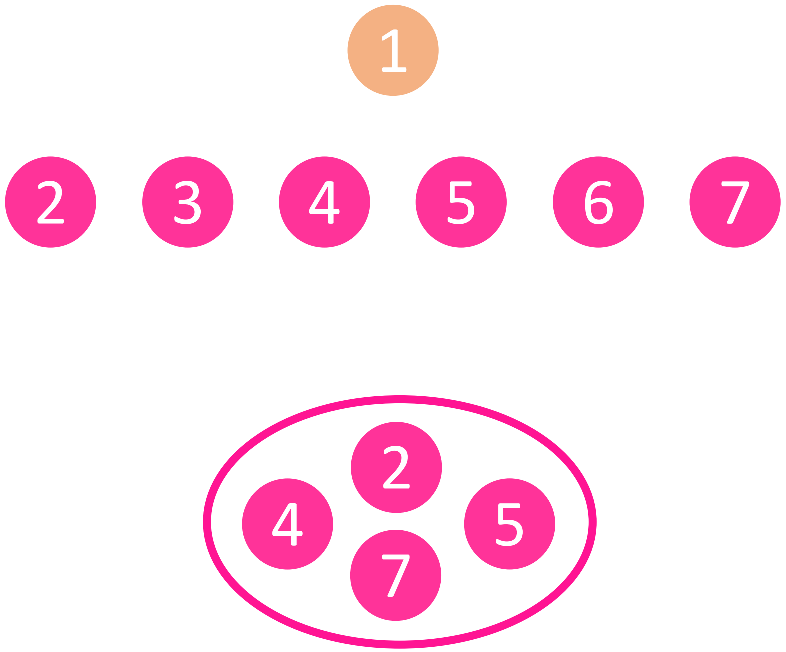
$\propto 1 - \boldsymbol{p}$) `Hierarchical`



~ **From real-word hyperedge size distribution**

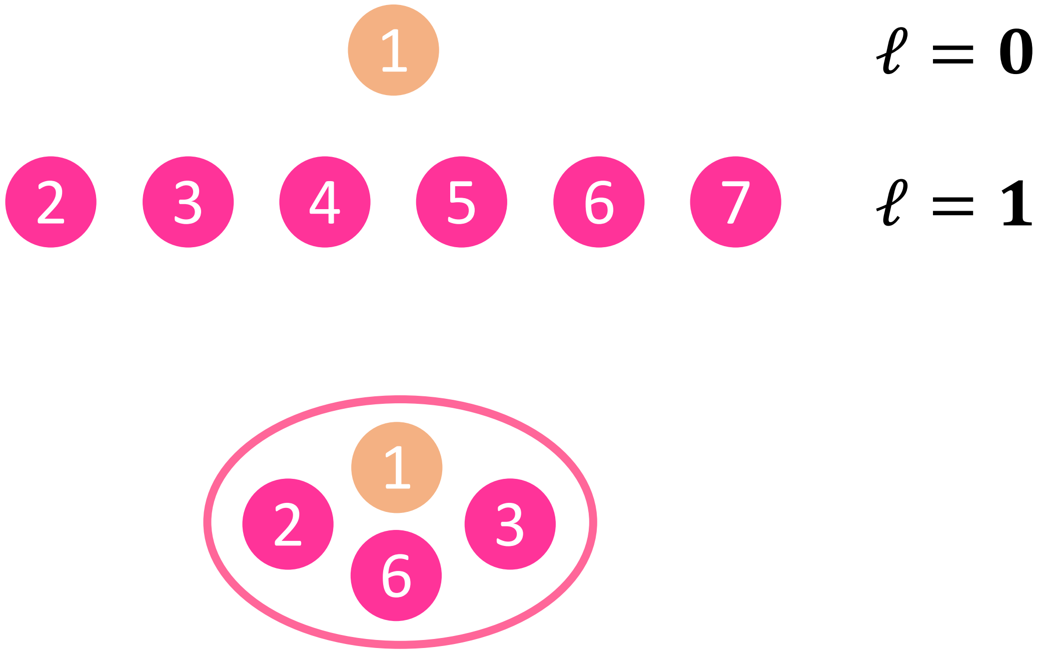
# THera: Transitivity-preserving Generator

1. Start from node 1.
2. Introduce $C = 6$ nodes.
3. Decide hyperedge type ($\propto p$) & size.
4. If `IntraCommunity:` create a hyperedge within the community.
5. If `Hierarchical:` sample nodes depending on the level and fill a hyperedge with them.



# THera: Transitivity-preserving Generator

1. Start from node 1.
2. Introduce $C = 6$ nodes.
3. Decide hyperedge type ($\propto p$) & size.
4. If **`IntraCommunity:`** create a hyperedge within the community.
5. If **`Hierarchical:`** sample nodes depending on the level and fill a hyperedge with them.

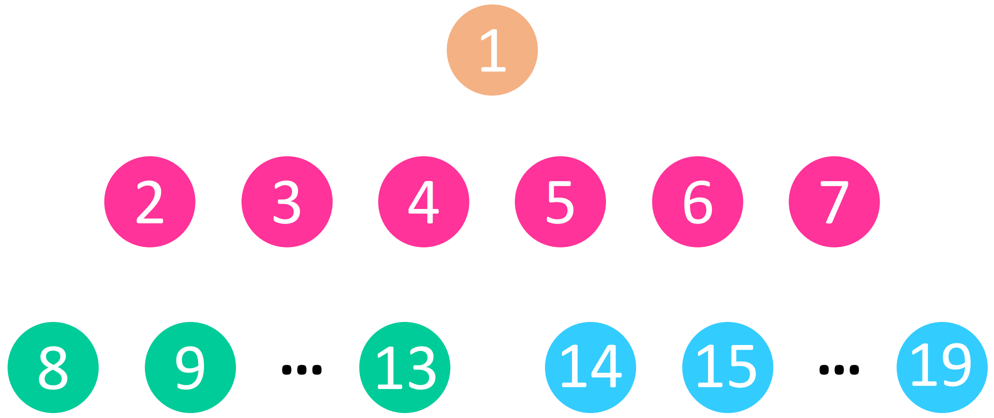



$$P(v_1) \propto \alpha^{-0}$$

$$P(v_2) = \cdots = P(v_7) \propto \alpha^{-1}$$

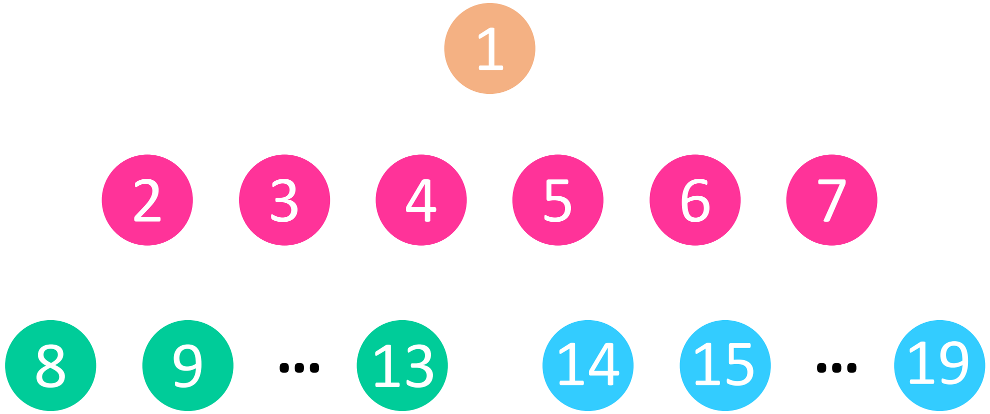
# THera: Transitivity-preserving Generator

7. Now, introduce $2 \times 6$ nodes.
8. Decide hyperedge type ($\propto p$) & size.
9. If `IntraCommunity` that contains $v_8$:
   create a hyperedge with green nodes.
10. If `IntraCommunity` that contains $v_{14}$:
    create a hyperedge with cyan nodes.
11. If `Hierarchical:` sample nodes dep. level and fill a hyperedge with them.



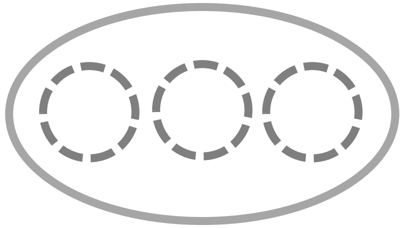
# THera: Transitivity-preserving Generator

7. Now, introduce 2×6 nodes.
8. Decide hyperedge type ($\propto p$) & size.
9. If `IntraCommunity` that contains $v_8$: create a hyperedge with green nodes.
10. If `IntraCommunity` that contains $v_{14}$: create a hyperedge with cyan nodes.
11. If `Hierarchical`: sample nodes dep. level and fill a hyperedge with them.




$\propto p$) `IntraCommunity`

$\propto 1-p$) `Hierarchical`



~ **From real-word hyperedge size distribution**

# THera: Transitivity-preserving Generator

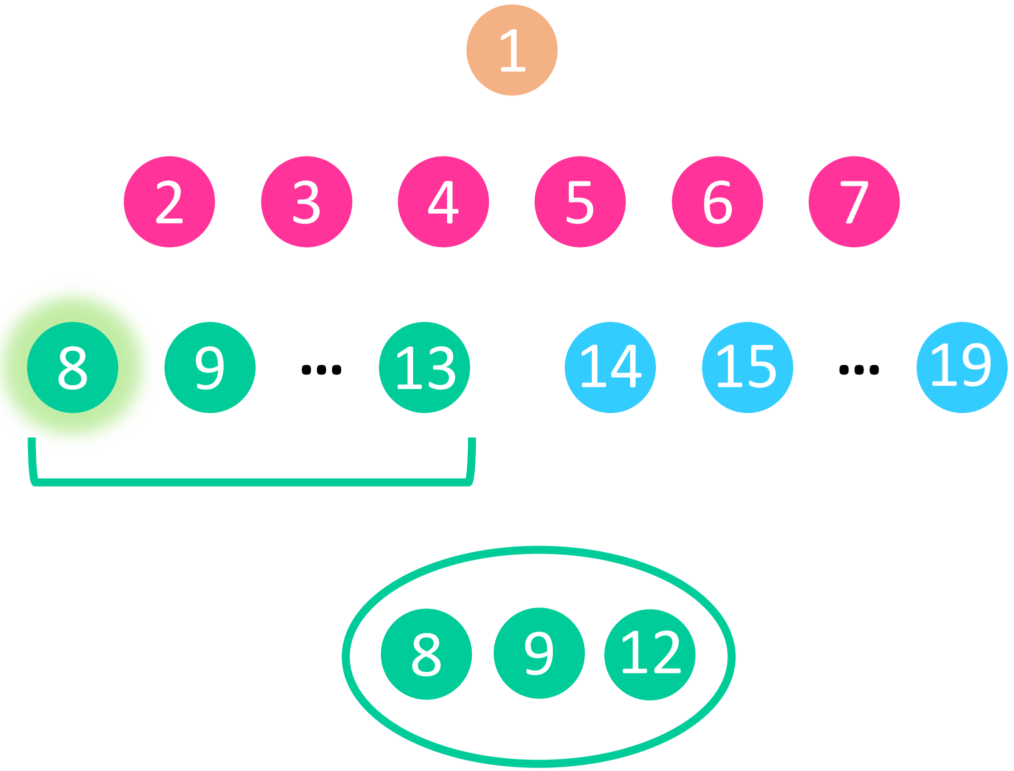
7. Now, introduce 2×6 nodes.

8. Decide hyperedge type ($\propto p$) & size.

9. If `IntraCommunity` that contains $v_8$:
   create a hyperedge with green nodes.

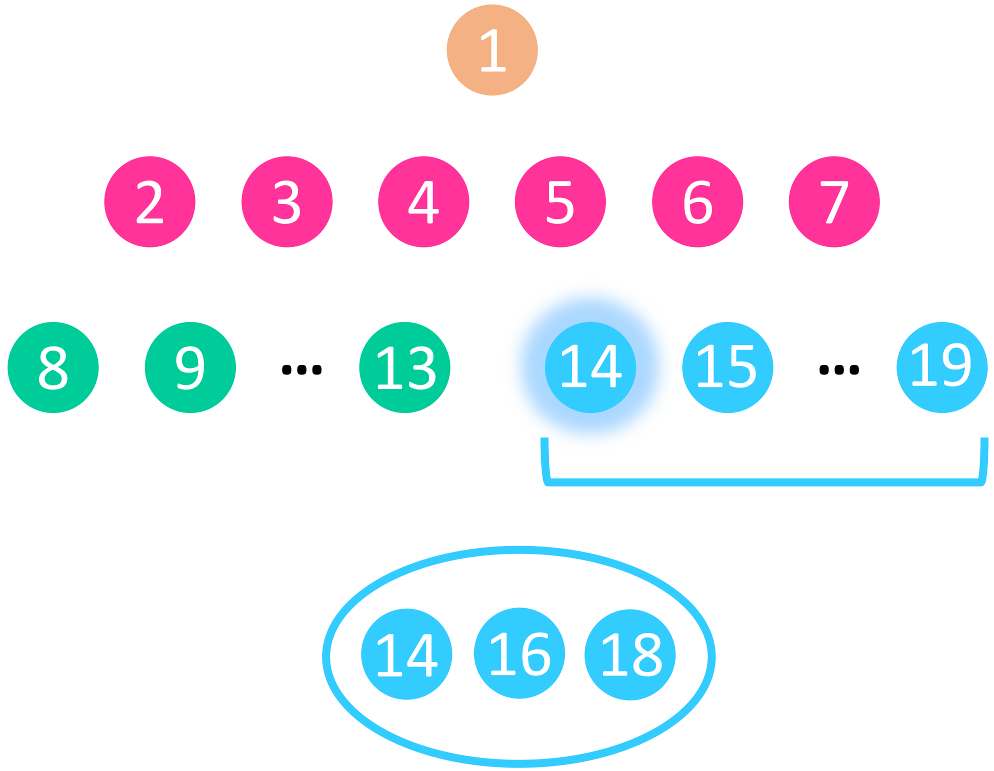
10. If `IntraCommunity` that contains $v_{14}$:
    create a hyperedge with cyan nodes.

11. If `Hierarchical`: sample nodes dep.
    level and fill a hyperedge with them.



# THera: Transitivity-preserving Generator

7. Now, introduce 2×6 nodes.
8. Decide hyperedge type ($\propto p$) & size.
9. If `IntraCommunity` that contains $v_8$: create a hyperedge with green nodes.
10. If `IntraCommunity` that contains $v_{14}$: create a hyperedge with cyan nodes.
11. If `Hierarchical`: sample nodes dep. level and fill a hyperedge with them.



# THera: Transitivity-preserving Generator

**dep: depending on the*

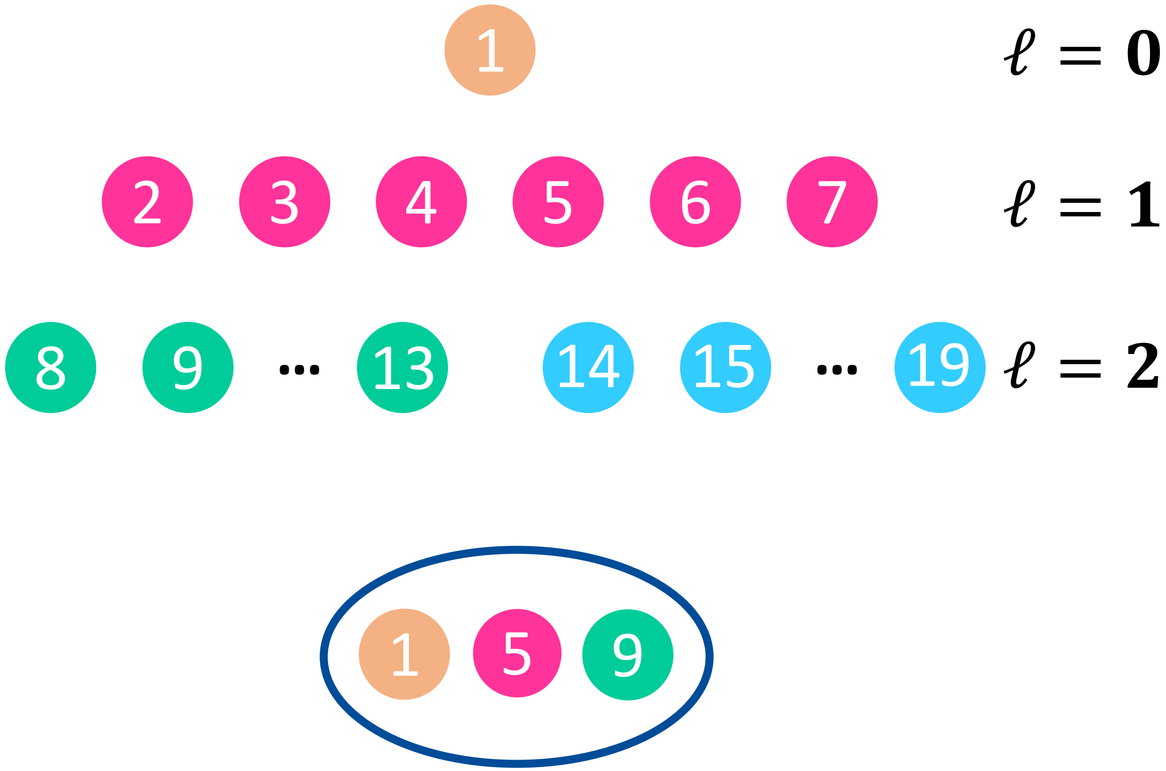
7. Now, introduce 2×6 nodes.

8. Decide hyperedge type ($\propto p$) & size.

9. If **`IntraCommunity`** that contains $v_8$:
   create a hyperedge with green nodes.

10. If **`IntraCommunity`** that contains $v_{14}$:
    create a hyperedge with cyan nodes.

11. If **`Hierarchical:`** sample nodes dep.
    level and fill a hyperedge with them.



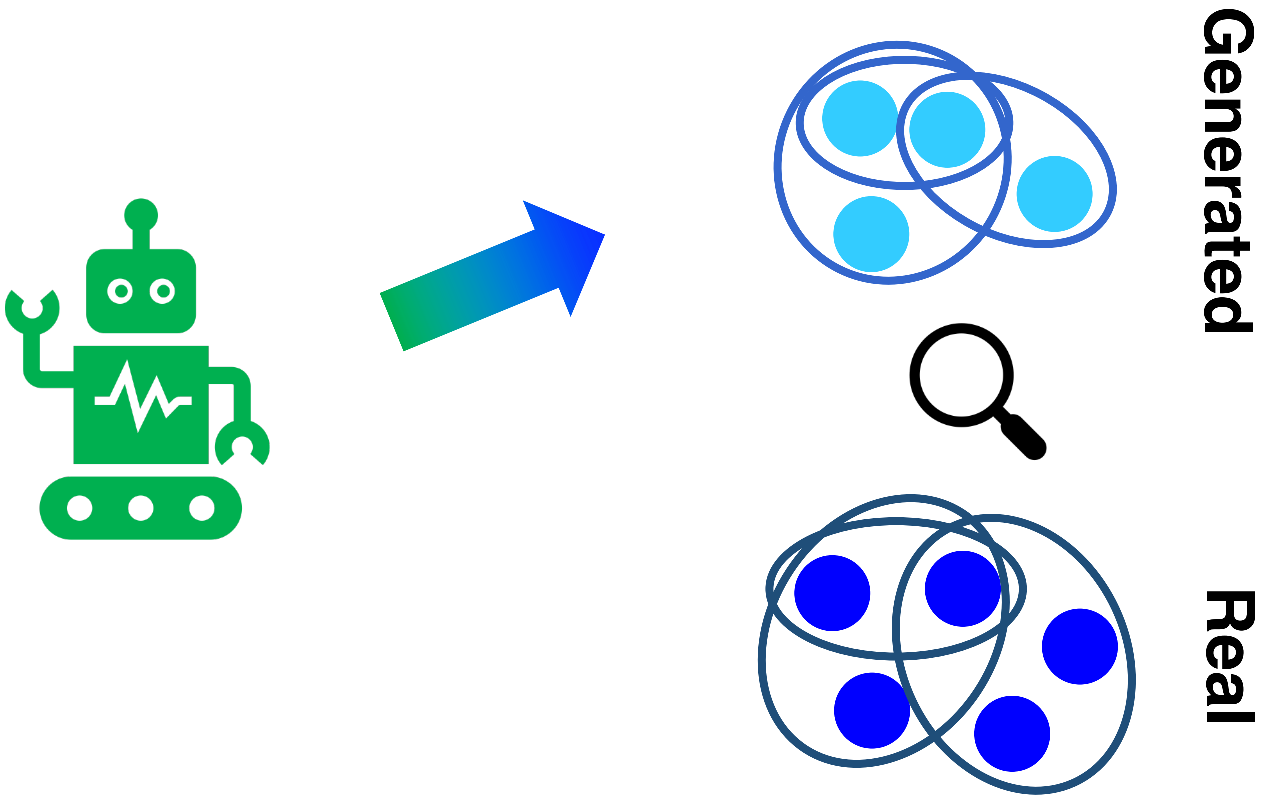

$$P(v_1) \propto \alpha^{-0}$$

$$P(v_2) = \cdots = P(v_7) \propto \alpha^{-1}$$

$$P(v_8) = \cdots = P(v_{19}) \propto \alpha^{-2}$$

# Reproducibility of THera

- Verify how well **THera** reproduces the transitivity patterns of real-world hypergraphs.
- We investigate whether **THera** reproduces Observation 1 – 4 well.



# Preserving Observation 1

- **THera** preserves the hypergraph transitivity value of real-world hypergraphs most well among the 6 generators.

| Statistic | Generator | email enron | email eu | NDC classes | NDC substances | contact high | contact primary | coauthorship dblp | coauthorship geology | coauthorship history | q&a ubuntu | q&a server | q&a math | Average ranking |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hypergraph transitivity $T(G)$ | **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 | Real |
| | **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 | 1.08 |
| | **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 | 4.08 |
| | **HyperPA** [16] | 0.090 | 0.110 | 0.070 | - | 0.121 | 0.153 | - | - | - | 0.003 | - | - | 4.75 |
| | **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 | 4.83 |
| | **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 | 3.25 |
| | **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 | 3.54 |

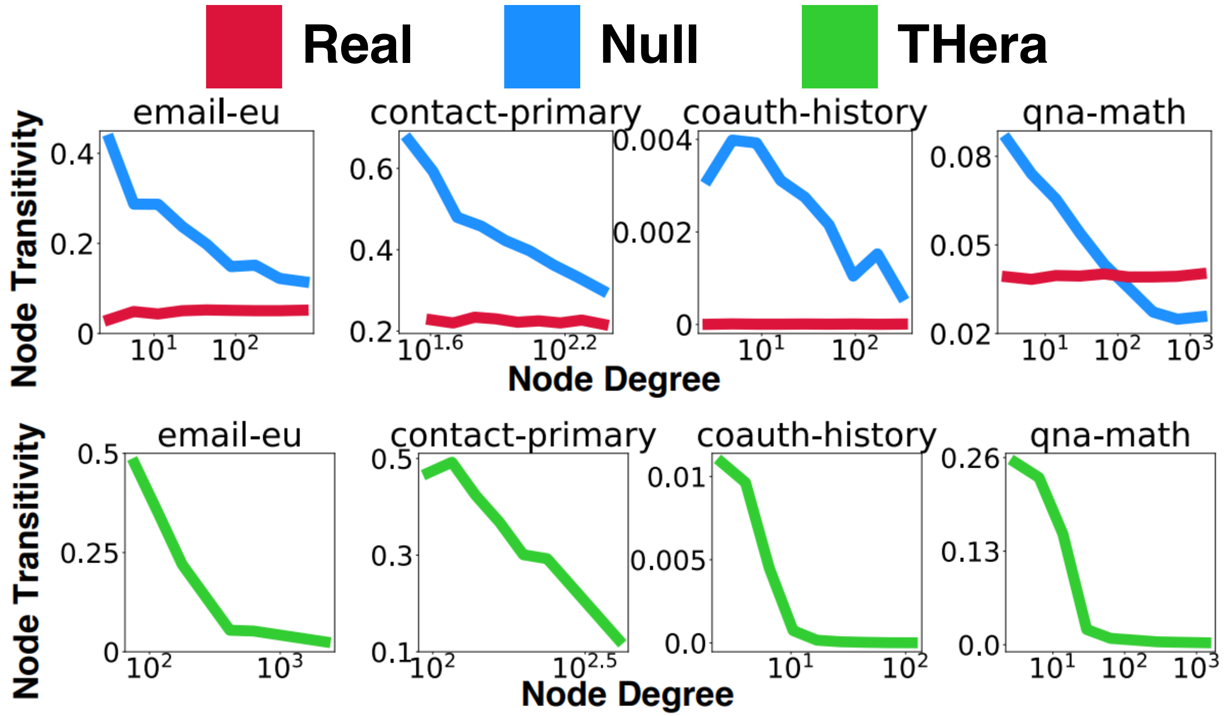
**** Colors: Most well**

# Preserving Observation 2 & 3

- **THera** preserves the correlation between,
  - (Obs 2) Hyperwedge transitivities and body-group sizes.
  - (Obs 3) Node transitivities and node degrees.

| Data | Real | HyperCL | THera |
|---|---|---|---|
| email-enron | 0.09 | -0.09 | 0.23 |
| email-eu | 0.12 | -0.14 | 0.22 |
| ndc-classes | 0.32 | -0.10 | 0.40 |
| ndc-substances | 0.14 | -0.10 | 0.24 |
| contact-high | 0.13 | 0.00* | 0.29 |
| contact-primary | 0.13 | 0.00* | 0.30 |
| coauth-dblp | 0.12 | 0.00* | 0.20 |
| coauth-geology | 0.14 | 0.00* | 0.26 |
| coauth-history | 0.12 | 0.05 | 0.19 |
| qna-ubuntu | 0.04 | 0.00* | 0.03 |
| qna-server | 0.04 | 0.00* | 0.04 |
| qna-math | 0.04 | 0.01 | 0.13 |

**Observation 2**




**Observation 3**

# Preserving Observation 4

- **THera** preserves the range of hyperedge transitivity well.

| Data | Real | HyperCL | THera |
|---|---|---|---|
| email-enron | 0.725 | 0.279 | 0.732 |
| email-eu | 0.809 | 0.248 | 0.792 |
| ndc-classes | 0.600 | 0.075 | 0.410 |
| ndc-substances | 1.0 | 0.032 | 0.411 |
| contact-high | 0.794 | 0.316 | 0.768 |
| contact-primary | 0.693 | 0.395 | 0.839 |
| coauth-dblp | 1.0 | 0.105 | 1.0 |
| coauth-geology | 1.0 | 0.069 | 1.0 |
| coauth-history | 1.0 | 0.333 | 1.0 |
| qna-ubuntu | 0.667 | 0.5 | 1.0 |
| qna-server | 0.667 | 0.333 | 1.0 |
| qna-math | 0.667 | 1.00 | 1.0 |

# Scalability of THera

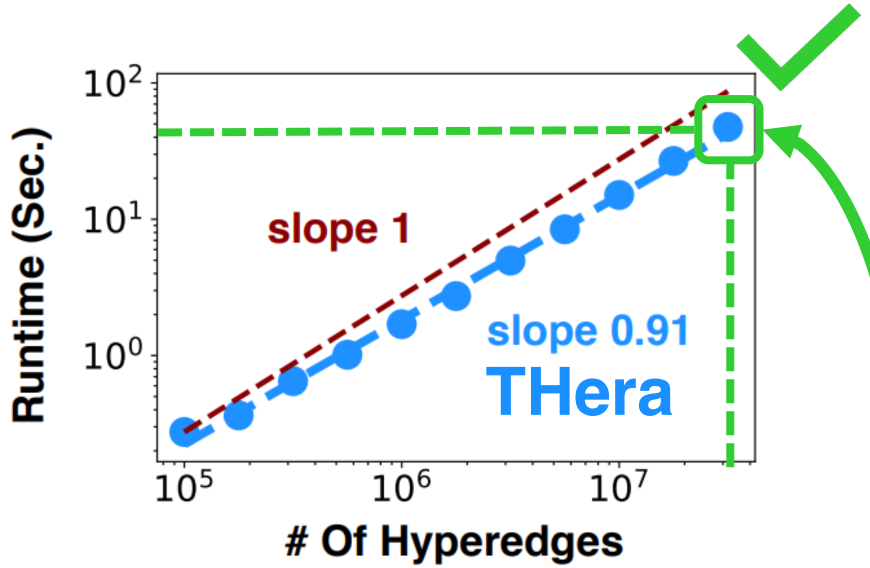
- **THera** achieves the fastest generation time and the least memory consumption among the 5 generators.

| Statistic | Generator | coauthorship dblp | coauthorship geology | coauthorship history | q&a ubuntu | q&a math | q&a server | email enron | email eu | ndc classes | ndc substances | contact high | contact primary | Average ranking |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Runtime (sec.) | THera | 4.07 | 2.39 | 1.86 | 0.39 | 0.33 | 0.63 | 0.04 | 0.18 | 0.04 | 0.09 | 0.10 | 0.18 | 1.7 |
| | HyperPA [16] | - | - | - | 374.51 | - | - | 5.02 | 5011.84 | 1155.00 | - | 2.29 | 4.56 | 5.0 |
| | HyperFF [30] | 226.79 | 114.24 | 53.48 | 12.78 | 17.23 | 3.15 | 0.05 | 0.09 | 0.32 | 0.10 | 0.02 | 0.02 | 2.7 |
| | HyperLap [33] | 19.07 | 9.57 | 2.24 | 0.57 | 0.45 | 1.14 | 0.01 | 0.14 | 0.01 | 0.10 | 0.06 | 0.04 | 1.8 |
| | HyperLap+ [33] | 1322.6 | 611.91 | 160.75 | 12.08 | 1.00 | 23.46 | 0.06 | 2.74 | 0.29 | 3.92 | 0.26 | 0.51 | 3.7 |
| Memory consumption (MB) | THera | 1535 | 761 | 325 | 90 | 36 | 129 | 2 | 3 | 2 | 11 | 11 | 23 | 2.0 |
| | HyperPA [16] | - | - | - | 21 | - | - | 76 | 60169 | 21351 | - | 2 | 2 | 4.2 |
| | HyperFF [30] | 3307 | 1655 | 739 | 107 | 147 | 26 | 1 | 2 | 4 | 1 | 1 | 1 | 2.0 |
| | HyperLap [33] | 3197 | 1498 | 529 | 134 | 86 | 204 | 11 | 32 | 15 | 41 | 12 | 11 | 3.1 |
| | HyperLap+ [33] | 5042 | 3005 | 1901 | 591 | 110 | 321 | 9 | 39 | 14 | 412 | 10 | 18 | 3.8 |

# Scalability of THera

- **THera** has the least time & memory complexity among the 3 incremental generators.

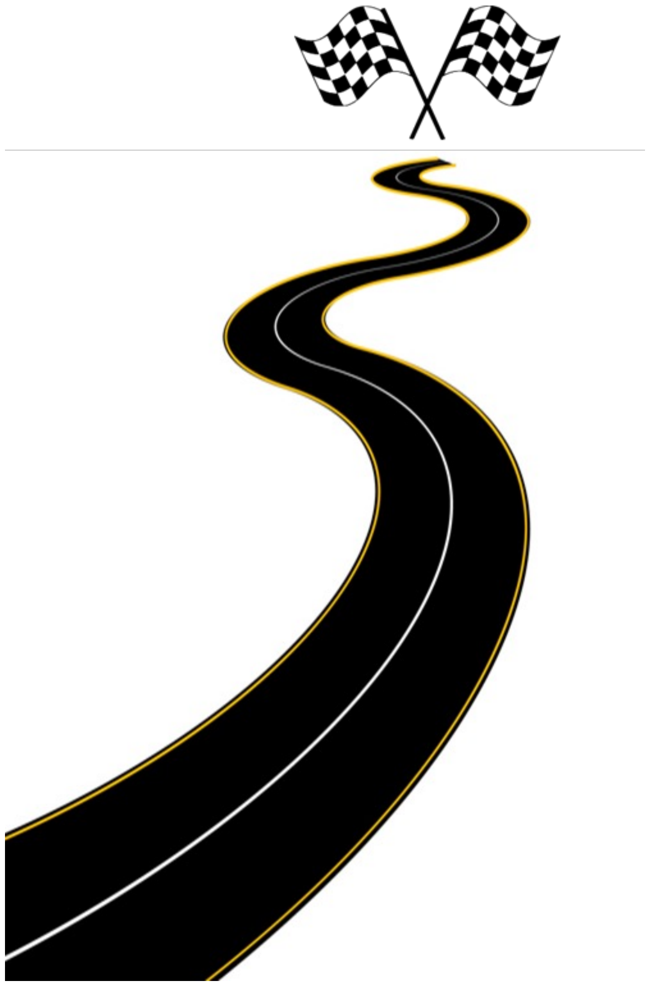
| Generator | Time complexity | Memory complexity | Incremental |
|---|---|---|---|
| THERA | $O(\log_2 \lvert V\rvert \times \sum_{e\in E}\lvert e\rvert)$ | $O(\lvert V\rvert + \sum_{e\in E}\lvert e\rvert)$ | ✔ |
| HyperPA [16] | $O(\sum_{e\in E} \log_2 \binom{\lvert V\rvert}{\lvert e\rvert})$ | $O(\sum_{e\in E} 2^{\lvert e\rvert})$ | ✔ |
| HyperFF [30] | $O(\lvert V\rvert \times \sum_{e\in E}\lvert e\rvert)$ | $O(\lvert V\rvert + \sum_{e\in E}\lvert e\rvert)$ | ✔ |
| HyperLap [33] | $O(\sum_{e\in E}\lvert e\rvert)$ | $O(\lvert V\rvert + \sum_{e\in E}\lvert e\rvert)$ | ✗ |
| HyperLap+ [33] | $O(\log_2 \lvert V\rvert \times \sum_{e\in E}\lvert e\rvert)$ | $O(\lvert V\rvert + \sum_{e\in E}\lvert e\rvert)$ | ✗ |




*Can generate $10^{7.5}$ ≈ **3.5M** hyperedges within a minute!*

# Roadmap

- Overview
- Transitivity Measure
- Observations
- Generator
- **Conclusions**



# Conclusions

## In this work, we present…

- **Criteria: Axioms** that a principled transitivity measure should satisfy.
- **Measure: HyperTrans**, a principled transitivity measure.
- **Observations: Analysis** of the real-world hypergraphs.
- **Generator: THera**, a scalable and realistic hypergraph generator.

**Paper:** https://doi.org/10.1145/3580305.3599382

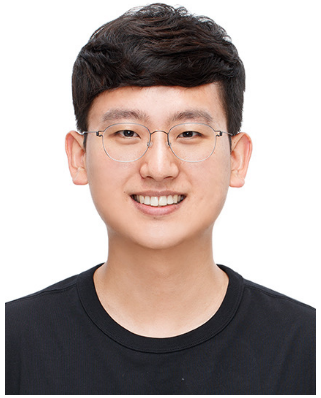
**Github:** https://github.com/kswoo97/hypertrans

# Thank You For Listening!

**Sunwoo Kim**

**Fanchen Bu**

**Minyoung Choe**



**Jaemin Yoo**

**Kijung Shin**